

**T.C.
MERSİN ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME BİLGİ YÖNETİMİ
ANABİLİM DALI**



MAKİNE ÖĞRENME PERSPEKTİFİNDEN TELE- PAZARLAMA BAŞARI STRATEJİSİ

Yüksek Lisans Tezi

**Hazırlayan
Mehmet SALTİ**

**Danışman
Doç. Dr. Evrim Ersin KANGAL**

AĞUSTOS-2023, MERSİN

**T.C.
MERSİN ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME BİLGİ YÖNETİMİ
ANABİLİM DALI**

**MAKİNE ÖĞRENME PERSPEKTİFİNDEN TELE-PAZARLAMA
BAŞARI STRATEJİSİ**

Yüksek Lisans Tezi

**Hazırlayan
Mehmet SALTİ
ORCID No: 0000-0002-1122-2464**

**Danışman
Doç. Dr. Evrim Ersin KANGAL
ORCID No: 0000- 0001-5906-3143**

**2. Danışman
Dr. Öğr. Üyesi Bilgin ZENGİN
ORCID No: 0000- 0002-9355-8294**

AĞUSTOS-2023, MERSİN

ÖZET

MAKİNE ÖĞRENME PERSPEKTİFİNDEN TELE-PAZARLAMA BAŞARI STRATEJİSİ

Makine öğrenme insanoğlu olarak inceleyemeyeceğimiz kadar büyük miktarda verilerin olduğu birçok alanda yeni şeyler öğrenme yeteneği vermektedir. Dolayısıyla, makine öğrenme bilgisayarlar verilerdeki örneklere bakarak kalıpları öğrenmeyi öğrettiğimiz bir süreçtir, böylece bu kalıpları tanırlar ve onları daha önce görmedikleri yeni şeylere uygularlar. Bu tez çalışması kapsamında ele alınan temel hipotez şöyledir: denetimli sınıflandırma işlerinde öznitelik seçimi özellikler arasındaki korelasyon temelinde gerçekleştirilebilir ve böyle bir seçim süreci çeşitli makine öğrenimi algoritmaları için faydalı olabilir. Bu bağlamda, seçilen bir telefonla pazarlama veri setinin bir makine öğrenme algoritmasında analiz edilmesiyle herhangi bir şirketin karar destek mekanizması için başarılı tahminlerin yapılıp yapılamayacağı araştırılmıştır. Sonuç olarak, bu tez denetimli bir makine öğrenimi analizinde özellik seçiminin karşılık gelen öznitelikler arasındaki korelasyon temelinde yapılabileceğini ortaya koymaktadır.

Anahtar Kelimeler: Yapay Zekâ, Makine Öğrenme, Derin Öğrenme, Tele-pazarlama.

Danışman: Doç. Dr. Evrim Ersin KANGAL, İşletme Bilgi Yönetimi Anabilim Dalı, Mersin Üniversitesi, Mersin.

İkinci Danışman: Dr. Öğr. Üyesi Bilgin ZENGİN, Hesaplamalı Bilimler ve Mühendislik Anabilim Dalı, Munzur Üniversitesi, Tunceli.

ABSTRACT

TELEMARKETING SUCCESS STRATEGY FROM MACHINE LEARNING PERSPECTIVE

Machine learning (ML) provides us the flair to achieve knowledge in different areas from large data pools that we as human beings may not study. So, the ML is a process in which we train computers to recognize patterns in a dataset, such that they learn that significant information and use them to achieve new things that they haven't encountered before. The main hypothesis covered in this thesis is as follows: feature selection in supervised classification tasks can be performed on the basis of correlation between features and such a selection process can be beneficial for various machine learning algorithms. On this purpose, we turned our attention on investigating whether successful predictions can be achieved for a company's decision-support mechanism by analyzing a selected telemarketing dataset in a machine learning algorithm. In conclusion, this thesis reveals that the feature selection in a supervised machine learning analysis can be performed on the basis of correlation among the corresponding attributes.

Keywords: Artificial Intelligence, Machine Learning, Deep Learning, Tele-marketing.

Advisor: Assoc. Prof. Evrim Ersin KANGAL, Department of Business Information Management, Mersin University, Mersin.

Co-Advisor: Assist. Prof. Bilgin ZENGİN, Department of Computational Science and Engineering, Munzur University, Tunceli.

TEŞEKKÜR

Tez çalışmam boyunca desteklerini esirgemeyen danışman hocalarım Sayın Doç. Dr. Evrim Ersin KANGAL ve Sayın Dr. Öğr. Üyesi Bilgin ZENGİN'e, hayatımın her aşamasında bana ışık tutan ve her zaman yanımda olan Doç. Dr. Mustafa SALTİ başta olmak üzere tüm aile fertlerime ve manevi desteklerinden dolayı yol arkadaşım Hazal DAĞ'a teşekkürlerimle...



İÇİNDEKİLER

	Sayfa
İÇ KAPAK	i
ONAY	ii
ETİK BEYAN	iii
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
TABLolar DİZİNİ	ix
ŞEKİLLER DİZİNİ	xi
KISALTMALAR VE SİMGELER	xv
GİRİŞ	1
1. KURAMSAL TEMELLER	2
1.1. Tele-pazarlama Nedir?	2
1.2. Makine Öğrenimi Nedir? Ne İşe Yarar?	2
1.3. Öznitelik Seçimi ve Örnekleme	5
1.4. Veri Dengeleme ve Sınıflandırma	11
1.5. Veri Tipleri, Önemli Veri Kaynakları ve Verilerin Sunumu	13
1.6. Karar Destek Sistemi	15
1.7. Öğrenme Yaklaşımları	16
1.7.1. NBC	17
1.7.2. LSTM ve “Bırakma” Aracı	19
1.7.3. Genetik Algoritma	25
1.7.4. FIM	27
1.7.5. DTs	28
1.7.6. Örnek-tabanlı Öğrenim	30
1.7.7. Lojistik Regresyon	32
1.7.8. Destek Vektör Makineleri	35
1.7.9. kNN	39
1.8. Mesafe Ölçümü Modelleri	43
1.9. Performans Metrikleri	54
2. KAYNAK ARAŞTIRMALARI	60
3. MATERYAL VE METOT	64
3.1. Kodlama Materyalleri	64
3.2. Öğrenme Metodolojisi	64
3.3. Veri Kümesi	66
3.4. Analiz Yöntemi	69
3.4.1. KMA Işınım Sıcaklığına Derin Öğrenme Yaklaşımı	70
3.4.2. Hubble Parametresinin Veri-yönlendirmeli Tahmin Temelli Modellemesi	78
3.4.3. Derin Öğrenme Destekli Hubble Parametresi Analizi	84
4. BULGULAR VE TARTIŞMALAR	87
4.1. Veri Önışleme	87
4.1.1. Etkisiz Özniteliklerin Dışlanması	87
4.1.2. Veri Tipi Dönüştürme	88
4.1.3. Bilgi Görselleştirme	89
4.1.4. Verilerin Dengeli Biçime Genişletilmesi	91
4.2. Algoritmayı Ayarla, Eğit ve Test Et!	100
4.3. Nihai Algoritmayı Tasarla ve Çalıştır!	112
4.4. Ne Öğrendik?	125
SONUÇ VE ÖNERİLER	127
KAYNAKLAR	129
BENZERLİK RAPORU ÖZET SAYFASI	145
ÖZGEÇMİŞ	146

TABLOLAR DİZİNİ

	Sayfa
Tablo 1.1. Popüler veri havuzu kaynakları.	14
Tablo 1.2. Örnek veri kümesi: golf oynamaya yönelik bilgiler.	15
Tablo 1.3. Hava durumuna göre golf verileri.	17
Tablo 1.4. Sıcaklığa göre golf verileri.	17
Tablo 1.5. Nem oranına göre golf verileri.	17
Tablo 1.6. Rüzgâr durumuna göre golf verileri.	18
Tablo 1.7. Golf verileri için hesaplanan mesafe değerleri.	32
Tablo 1.8. Sınava giren öğrencilerin hazırlık ve sonuç bilgileri.	33
Tablo 3.1. Tez kapsamında kullanılan mesafe ölçüm yöntemleri.	66
Tablo 3.2. Seçilen telefonla pazarlama veri havuzundan örnekler (1. Kısım).	67
Tablo 3.3. Seçilen telefonla pazarlama veri havuzundan örnekler (2. Kısım).	68
Tablo 3.4. Seçilen telefonla pazarlama veri havuzundan örnekler (3. Kısım).	69
Tablo 3.5. KMA ışıınım sıcaklığı için astrofizik gözlem verileri (1. Kısım).	70
Tablo 3.6. KMA ışıınım sıcaklığı için astrofizik gözlem verileri (2. Kısım).	71
Tablo 3.7. Genetik algoritmada kullanılan hiper-parametre değerleri.	72
Tablo 3.8. KMA modeli için keyfi parametrelerin en uygun değerleri.	73
Tablo 3.9. LSTM-Bırakma analizimizde en uygun hiper-parametre değerleri.	77
Tablo 3.10. Güncel Hubble parametresi ölçümleri (1. Kısım).	79
Tablo 3.11. Güncel Hubble parametresi ölçümleri (2. Kısım).	80
Tablo 3.12. Hubble parametresi modelimizdeki keyfi niceliklerin en uygun değerleri.	80
Tablo 3.13. Güncel Hubble sabiti ölçümleri.	82
Tablo 3.14. Hubble parametresi modelimiz için uygun hiper-parametre değerleri.	85
Tablo 4.1. Veri kümesindeki öznitelik türleri.	87
Tablo 4.2. Kategorik “Sales (Satışlar)” bilgilerinin sayısal dönüşümü.	88
Tablo 4.3. Kategorik “Gender (Cinsiyet)” bilgilerinin sayısal dönüşümü.	88
Tablo 4.4. CityBlock modeli analizinde elde edilen tutarlılık oranları.	102
Tablo 4.5. Chebyshev yaklaşımı analizinde elde edilen tutarlılık oranları.	104
Tablo 4.6. Euclidean ölçüm modeli analizinde elde edilen tutarlılık oranları.	105
Tablo 4.7. Correlation modeli analizinde elde edilen tutarlılık oranları.	107
Tablo 4.8. Canberra ölçümünün analizinde elde edilen tutarlılık oranları.	108
Tablo 4.9. Dice ölçümü için elde edilen tutarlılık oranları.	110
Tablo 4.10. Cosine modelinde elde edilen tutarlılık oranları.	111
Tablo 4.11. kNN algoritmasında ayarlama evresinin önemli sonuçları.	111
Tablo 4.12. CityBlock-SFFS-kNN_vk2 algoritmasının performans raporu.	113
Tablo 4.13. CityBlock-SFFS-kNN_vk2 mimarisinin detaylı sınıflandırma raporu.	114
Tablo 4.14. Chebyshev-SFFS-kNN_vk2 ve Euclidean-SFFS-kNN_vk2 raporları	115
Tablo 4.15. Chebyshev-SFFS-kNN_vk2 mimarisinin detaylı sınıflandırma raporu.	115
Tablo 4.16. Euclidean-SFFS-kNN_vk2 mimarisinin detaylı sınıflandırma raporu.	116
Tablo 4.17. Correlation-SFFS-kNN_vk1 yaklaşımı performans raporu.	117
Tablo 4.18. Correlation-SFFS-kNN_vk1 mimarisinin detaylı sınıflandırma raporu.	118
Tablo 4.19. Canberra-SFFS-kNN_vk2 yaklaşımı performans raporu.	119
Tablo 4.20. Canberra-SFFS-kNN_vk2 mimarisinin detaylı sınıflandırma raporu.	120
Tablo 4.21. Dice-SFFS-kNN_vk1 ve Dice-SFFS-kNN_vk5 için performans raporu.	121
Tablo 4.22. Dice-SFFS-kNN_vk1 mimarisinin detaylı sınıflandırma raporu.	122
Tablo 4.23. Dice-SFFS-kNN_vk5 mimarisinin detaylı sınıflandırma raporu.	123
Tablo 4.24. Cosine-SFFS-kNN_vk2 için performans raporu.	124
Tablo 4.25. Cosine-SFFS-kNN_vk2 mimarisinin detaylı sınıflandırma çıktıları.	125
Tablo 4.26. Tasarlanan kNN algoritmalarının karşılaştırılması.	125

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1. Sıradan bir ML algoritmasının genel akış şeması.	2
Şekil 1.2. Öznitelik seçim yöntemleri.	6
Şekil 1.3. Denetimli öznitelik seçiminde önemli modeller.	7
Şekil 1.4. Örneklemeye yönetimine temsili bir örnek.	9
Şekil 1.5. Sık kullanılan örneklemeye yönetimleri.	9
Şekil 1.6. Örneklemeye çerçevesinin ana basamakları.	10
Şekil 1.7. Veri azaltma (aşağı örneklemeye) yöntemiyle veri dengeleme.	12
Şekil 1.8. Veri çoğaltma (yukarı örneklemeye) yöntemiyle veri dengeleme.	12
Şekil 1.9. Temel veri tipleri.	13
Şekil 1.10. Makine öğrenme yaklaşımlarına genel bir bakış.	16
Şekil 1.11. Tipik bir LSTM modülü diyagramı.	20
Şekil 1.12. Genetik algoritma uygulama alanlarına örnekler.	25
Şekil 1.13. Genetik algortmada genel akış şeması.	26
Şekil 1.14. Golf veri kümesi için bir karar ağacı.	30
Şekil 1.15. IBL yaklaşımında yakın komşuluk testinin bir temsili.	31
Şekil 1.16. LR yaklaşımın önemli avantajları.	32
Şekil 1.17. LR yaklaşımına uygun alanlar.	35
Şekil 1.18. Destek Vektör Makinelerinin yaygın kullanım alanları.	36
Şekil 1.19. Sınıflandırma analizinin temsili akışı.	36
Şekil 1.20. SVMs yaklaşımının temsili bir şeması	37
Şekil 1.21. kNN sınıflandırması için temsili bir görsel.	40
Şekil 1.22. Sınıflandırma durumları.	41
Şekil 1.23. Karışıklık matrisi ve bazı performans metrikleri.	42
Şekil 1.24. Popüler mesafe ölçüm fonksiyonu tipleri.	44
Şekil 1.25. Bazı mesafe ölçüm yöntemleri.	45
Şekil 1.26. MSE hesabı için örnek bir grafik.	55
Şekil 1.27. MAE hesabı için örnek bir grafik.	55
Şekil 1.28. Karışıklık matrisinin sade bir temsili.	57
Şekil 1.29. Örnek bir AUROC eğrisi.	59
Şekil 3.1. Sarıcı kNN mekanizmasının genel akış şeması.	65
Şekil 3.2. Uyarılma yaklaşımımızın temel işlem akışı şeması.	72
Şekil 3.3. KMA sıcaklık modeli parametrelerinin kombinasyonu için olasılık dağılımları.	73
Şekil 3.4. KMA ısıtım sıcaklığının davranışı.	74
Şekil 3.5. LSTM-Bırakma algoritması test sürecinin R -kare analizi.	75
Şekil 3.6. Belirli aralıklar için R -kare analizimizin sayısal sonuçları.	76
Şekil 3.7. FIM yaklaşımına göre KMA ısıtım sıcaklığı tahmin analizi.	77
Şekil 3.8. LSTM-Bırakma yaklaşımına göre KMA ısıtım sıcaklığı tahmin analizi.	78
Şekil 3.9. Hubble parametresi modeli keyfi niceliklerinin olasılık dağılımları.	81
Şekil 3.10. $H(z)$ fonksiyonunu davranışı ve güncel gözlemsel Hubble parametresi verileri.	83
Şekil 3.11. FIM yaklaşımı ile elde edilen tahmin $H(z)$ verileri.	83
Şekil 3.12. Hiper-parametre ayarlama evresi nümerik R -kare sonuçları.	84
Şekil 3.13. LSTM-Bırakma algoritmasının Hubble parametresi için verdiği tahmin sonuçları.	86
Şekil 4.1. Veri yolu tanımlama ve işlevsiz özniteliklerin temizlenmesi.	88
Şekil 4.2. Kategorik verilerin dönüştürülmesi.	89
Şekil 4.3. Nümerik verilerin algortmaya tanıtılması.	89
Şekil 4.4. Ele alınan özniteliklerin korelasyon katsayıları.	90
Şekil 4.5. Korelasyon tablosu komut satırları.	90
Şekil 4.6. “Sales (Satışlar)” özniteliğinin seçenek dağılımları.	91
Şekil 4.7. SMOTE ile “True (ya da 1)” etiketli yeni veri sentezleme.	91
Şekil 4.8. “Sales (Satışlar)” özniteliğinin SMOTE sonrası seçenek dağılımları.	91
Şekil 4.9. SMOTE öncesi (üst) ve sonrası (alt) müşteri cinsiyetleri dağılımı kod satırları.	92

	Sayfa
Şekil 4.10. SMOTE öncesi (sol) ve sonrası (sağ) müşteri cinsiyetleri dağılımı.	92
Şekil 4.11. SMOTE öncesi (üst) ve sonrası (alt) müşterilerin yaş-cinsiyet keman dağılımı kodları	93
Şekil 4.12. MOTE öncesi (sol) ve sonrası (sağ) müşterilerin yaş ve cinsiyet keman dağılımları.	93
Şekil 4.13. SMOTE öncesi (üst) ve sonrası (alt) müşteri sayısının yaşa göre dağılımı kod satırları.	94
Şekil 4.14. SMOTE öncesi (soldaki) ve sonrası (sağdaki) müşteri sayısının yaşa göre dağılımı.	94
Şekil 4.15. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-acente dağılımı kod satırları.	95
Şekil 4.16. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-acente dağılımı.	95
Şekil 4.17. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-acente-cinsiyet dağılımı kod satırları.	95
Şekil 4.18. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-acente-cinsiyet dağılımı.	96
Şekil 4.19. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-acente-sonuç dağılımı kod satırları.	96
Şekil 4.20. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-acente-sonuç dağılımı.	96
Şekil 4.21. SMOTE öncesi (üst) ve sonrası (alt) arama-cinsiyet keman dağılımı komut satırları.	97
Şekil 4.22. SMOTE öncesi (sol) ve sonrası (sağ) arama-cinsiyet keman dağılımı.	97
Şekil 4.23. SMOTE öncesi (üst) ve sonrası (alt) arama-sonuç keman dağılımı komut satırları.	97
Şekil 4.24. SMOTE öncesi (sol) ve sonrası (sağ) arama-sonuç keman dağılımı.	98
Şekil 4.25. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-ürün dağılımı kod satırları.	98
Şekil 4.26. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-ürün dağılımı.	98
Şekil 4.27. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-ürün-cinsiyet dağılımı için kodlar.	99
Şekil 4.28. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-ürün-cinsiyet dağılımı.	99
Şekil 4.29. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-ürün-sonuç dağılımı için kodlar.	100
Şekil 4.30. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-ürün-sonuç dağılımı.	100
Şekil 4.31. CityBlock ölçüm modelinin performans analizi kod satırları.	101
Şekil 4.32. CityBlock modeli performans analizi grafik kodları.	101
Şekil 4.33. CityBlock modeli performans analizi nümerik çıktı kodları.	102
Şekil 4.34. CityBlock ölçüm modelinin performans analizi.	102
Şekil 4.35. Chebyshev ölçüm modelinin performans analizi kod satırları.	103
Şekil 4.36. Chebyshev ölçüm modelinin performans analizi.	104
Şekil 4.37. Euclidean ölçüm modelinin performans analizi.	104
Şekil 4.38. Euclidean ölçüm modelinin performans analizi kod satırları.	105
Şekil 4.39. Correlation ölçüm modelinin performans analizi kod satırları.	106
Şekil 4.40. Correlation ölçüm modelinin performans analizi.	106
Şekil 4.41. Canberra ölçümünün performans analizi için Python kodları.	107
Şekil 4.42. Canberra ölçüm modelinin performans analizi.	108
Şekil 4.43. Dice ölçüm modelinin performans analizi Python kodları.	109
Şekil 4.44. Dice ölçümünün performans analizi.	109
Şekil 4.45. Cosine yaklaşımın performans analizi komut satırları.	110
Şekil 4.46. Cosine ölçüm modelinin performans analizi.	111
Şekil 4.47. CityBlock-SFFS-kNN_vk2 mimarisinin simülasyon kodları.	112
Şekil 4.48. CityBlock-SFFS-kNN_vk2 mimarisinin performans grafiği kodları.	113
Şekil 4.49. CityBlock-SFFS-kNN_vk2 mimarisinin performans grafiği.	113
Şekil 4.50. CityBlock-SFFS-kNN_vk2 mimarisinin sınıflandırma raporu.	114
Şekil 4.51. Euclidean-SFFS-kNN_vk2 mimarisinin simülasyon komutları.	114
Şekil 4.52. Chebyshev-SFFS-kNN_vk2 mimarisinin performans grafiği.	116
Şekil 4.53. Euclidean-SFFS-kNN_vk2 mimarisinin performans grafiği.	116
Şekil 4.54. Correlation-SFFS-kNN_vk1 mimarisinin simülasyon komutları.	117
Şekil 4.55. Correlation-SFFS-kNN_vk1 mimarisinin performans grafiği.	118
Şekil 4.56. Canberra-SFFS-kNN_vk2 algoritmasının simülasyon komutları.	118
Şekil 4.57. Canberra-SFFS-kNN_vk2 mimarisinin performans grafiği.	119
Şekil 4.58. Dice-SFFS-kNN_vk1 algoritmasının simülasyon komutları.	120
Şekil 4.59. Dice-SFFS-kNN_vk5 algoritmasının simülasyon komutları.	121
Şekil 4.60. Dice-SFFS-kNN_vk1 mimarisinin performans grafiği.	122
Şekil 4.61. Dice-SFFS-kNN_vk5 mimarisinin performans grafiği.	122
Şekil 4.62. Cosine-SFFS-kNN_vk2 mimarisinin simülasyon komutları.	123

	Sayfa
Şekil 4.63. Cosine-SFFS-kNN_vk5 mimarisinin performans grafiği.	124
Şekil 4.64. İncelenen veri setinin analizi için en uygun/faydalı kNN mimarisi.	125
Şekil 4.65. Tez analizlerinin genel akış şeması.	127



KISALTMALAR ve SİMGELER

Kısaltma/Simgesi	Tanım
ML	Machine Learning
AI	Artificial Intelligence
NN	Neural Network
NNs	Neural Networks
DSS	Decision Support System
NBC	Naive Bayes Classifier
LSTM	Long Short-Term Memory
DTs	Decision Trees
kNN	k-Nearest Neighbor
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
BNN	Bayesian Neural Network
ADAM	Adaptive Moment
NADAM	Nesterov Accelerated Adaptive Moment Estimation
ReLU	Rectifier Linear Unit
LReLU	Leaky Rectifier Linear Unit
ELU	Exponential Linear Unit
SELU	Scaled Exponential Linear Unit
FIM	Fisher Information Matrix
CART	Classification & Regression Tree
QUEST	Quick Unbiased Efficient Statistical Tree
IBL	Instance-based Learning
LR	Logistic Regression
SVMs	Support Vector Machines
MMC	Maximum Margin Classifier
OKM	Ortalama Karakter Mesafesi
KaE	Kare Euclidian
NeKiK	Neyman Ki-Kare
PeKiK	Pearson Ki-Kare
KaKiK	Kare Ki-Kare
OSKiK	Olasılıklı Simetrik Ki-Kare
KaSKiK	Katkılı Simetrik Ki-Kare
OSE	Ortalama Sansürlü Euclidean
VWH	Vicis-Wave Hedges
VS	Vicis Symmetric
MaSKiK	Maximum Symmetric Ki-Kare
MiSKiK	Minimum Symmetric Ki-Kare
OLL	Ortalama-(L_1 , L_∞)
KiKP	Ki-Kare Pearson
KiKi	Ki-Kare İstatistiği
WÇE	Whittaker'ın Çağırışım Endeksi
MSE	Mean Squared Error
MAE	Mean Absolute Error
AUROC	Area Under Receiver Operating Characteristics Curve
TPR	True Positive Rate
FPR	False Positive Rate
DPO	Doğru Pozitif Oran
YPO	Yanlış Pozitif Oran
CMB	Class Membership-based Classifier
SMOTE	Synthetic Minority Over-sampling Technique
ELM	Extreme Learning Machine

Kısaltma/Simgesi	Tanım
DoTeS	Dolandırıcılık Tespit Sistemleri
FDS	Fraud Detection Systems
CRISP-DM	Cross Industry Standard Process for Data Mining
SBS	Sequential Backward Selection
SBFS	Sequential Backward Floating Selection
SFS	Sequential Forward Selection
SFFS	Sequential Forward Floating Selection
CMB	Cosmic Microwave Background
KMA	Kozmik Mikrodalga Artalan
p_b	LSTM mimarisinde “Bırakma” olasılığı
k	kNN algoritması için ayarlanması gereken bir hiper-parametre
χ^2	Chi-squared
R^2	R-squared
F	Aktivasyon Fonksiyonu
Ψ	Ağırlık Matrisi
h	LSTM’de Gizli Katman
$\langle \rangle$	İlgili Niceliğin Vektör Olduğuna İşaret Eder
$\odot$	Dolaylı Matris Çarpımı
$\otimes$	Doğrudan Matris Çarpımı
b	Yanılıgı (Bias) Miktarı
$i^{(t)}$	Giriş Kapısı
$f^{(t)}$	Unutma Kapısı
$l^{(t)}$	Olası Yeni Durum
$c^{(t)}$	Değerlendirme Katmanı
$o^{(t)}$	Hücre Çıkış Kapısı
Σ	Sigmoid Fonksiyonu
$F_{\mu\nu}$	Fisher Bilgi Matrisi
θ_μ	Parametre Vektörü
$\Pi(\theta)$	FIM Yaklaşımında Nihai Olasılık
$L(\theta)$	Muhtemellik Dağılımı
$\psi(\theta)$	FIM’de Eski Olasılık
$\Upsilon(\theta)$	Bayes Kanıtı
$d(x, y)$	Düzlemde İki Nokta Arasındaki Mesafe
$\pi(x)$	Lojistik Fonksiyonu
β_0	Kesişme Noktası
β_1	Ters Ölçek Parametresi
$Odds(x)$	Tuhaflik Fonksiyonu
W	Hiper-düzlem Vektörü
R	Gerçel (Reel) Sayılar
$\in$	Kümenin Elemanıdır
$\cup$	Kümelerin Birleşimi
$\cap$	Kümelerin Kesişimi
$F1$	F puanı
z	Kızıla-kayma Parametresi
$T(z)$	KMA Işınım Sıcaklığı Fonksiyonu
$H(z)$	Hubble Parametresi Fonksiyonu
σ	Hata Payı
T_0	KMA Işınım Sıcaklığının Günümüzde Ölçülen Değeri
H_0	Hubble Sabiti
ϑ	Hız
d	Mesafe
$C_p(z_i \theta_\mu)$	Kuramsal Hubble Parametresi Değerleri

Kısaltma/Simge	Tanım
$M_D(z_i)$	Gözlemsel (Ölçülmüş) Hubble Parametresi Değerleri
Ω_m^0	Günümüzdeki Karanlık Madde Oranı
Ω_e^0	Günümüzdeki Karanlık Enerji Oranı
DP	Doğru Pozitif
YP	Yanlış Pozitif
DN	Doğru Negatif
YN	Yanlış Negatif
%	Yüzdelik Oran
$\approx$	Yaklaşık Değer
C	Karışıklık Matrisi

“Gerçek değer; gelişiyile boşluk dolduran değil, gidişiyile boşluk yaratandır...”
Bu tez çalışması aramızdan ayrılışının 5. yılında canım babam **Hacı Ömer SALTİ**’ya adanmıştır.

GİRİŞ

Pazarlama kampanyası temel olarak marka farkındalığının ve/veya satış hedeflerinin iyileştirilmesiyle ilgili olabilecek aynı amaca hizmet eden bir dizi ticari eylem topluluğudur. Geçmişte ticari operasyonların çoğunluğu kitlesel pazarlamaya dayanıyordu, günümüzde doğrudan ve hatta hedefli pazarlama giderek daha fazla arzu ediliyor (Moro ve ark., 2014). Şirketlerin belirli bir satış kampanyası için müşteri gruplarıyla iletişime geçerek doğrudan pazarlama stratejilerini kullanmaları başarı oranını arttırmıştır (Kotler ve Keller, 2015), böylece tele-pazarlama yöntemleri büyük önem kazanarak farklı sektörlerdeki çeşitli müşteri tiplerine göre özelleştirilmiştir. Diğer yandan, bilgisayar sistemlerinin veri işleme yetenekleri ve veri depolama birimlerinin hacimleri olmak üzere günümüz teknolojik gelişmeleri sayesinde müşterilerden toplanan bilgiler artık geçmişe göre daha zengin ve çeşitli hale gelmektedir. Ayrıca, makine öğrenmesi algoritmalarının ve müşteri bilgilerini içeren verilerin hammaddesini oluşturduğu veri madenciliği araçlarının ortaya çıkması pazarlama stratejilerine önemli katkılar sunma potansiyeline sahiptir (Buhl ve ark., 2013).

Bu tez çalışmasında temel amaç seçilen bir anonim tele-pazarlama veri setini en uygun makine öğrenme algoritması ile analiz ederek etkin bir başarı stratejisi oluşturmaktır. Tez araştırması bütün olarak ele alındığında esasen bir veri analiz yöntemi önerilerek başarılı tele-pazarlama sonuçlarına nasıl ulaşılabileceği ortaya konmaktadır. Takip edilen yöntem esasen farklı bakış açılarının sınanmasıyla en uygun yaklaşımın belirlenmesi fikrine dayanmaktadır. Tez çalışması bazı ana basamaklara ayrılmıştır ve bunları şöyle sıralamak mümkündür. Sıradaki bölümde kuramsal kavramlara dikkat çekmek amacıyla bazı temel bilgiler sunulmaktadır. Daha sonraki bölümde ise araştırmamıza kaynak teşkil eden önemli çalışmalar hakkında kısa bilgilere yer verilmektedir. Sonrasına tez çalışmasının kuramsal materyalleri ve takip edilen yöntemin ana hatları paylaşılmaktadır. Takip eden bölümde ise araştırma bulguları sunularak bu sonuçların kapsamlı analizleri yapılmıştır. Son bölüm ise nihai tartışmalara ve yeni araştırma konuları önerilerine ayrılmıştır.

1. KURAMSAL TEMELLER

1.1. Tele-pazarlama nedir?

Pazarlama satış kampanyaları bir şirketin belirli bir süre içerisinde üstesinden gelebileceği iş miktarının bir ölçüsü olan iş kapasitesini geliştirmeye yönelik tipik bir strateji aracılığıyla oluşturulmaktadır (Moro ve ark., 2014). Belirli hedeflere ulaşmak adına farklı müşteri tipleriyle iletişim içerisinde olan şirketler doğrudan pazarlama stratejileri kullanırken planladıkları kampanyaların operasyonel yönetimini kolaylaştırmak adına müşterilerle uzaktan etkileşimlerini bir iletişim biriminde merkezileştirmeyi tercih etmektedir. Bu tür çağrı merkezleri potansiyel müşterilerle çeşitli kanallar aracılığıyla iletişim kurmaktadır: telefon (sabit veya mobil hat) bu bağlamda en yaygın kullanılan araçtır. İletişim merkezi aracılığıyla gerçekleştirilen pazarlama yöntemi uzaklık özelliğinden dolayı literatürde *tele-pazarlama* olarak adlandırılmaktadır (Kotler ve Keller, 2015) ve hangi faktörün (müşteri veya iletişim merkezi) tetiklediğine bağlı olarak “gelen” ve “giden” olarak iki gruba ayrılmaktadır. Bu noktada her pazarlama durumun farklı zorluklar içerdiğini vurgulamakta fayda vardır. Örneğin, “giden” aramalar genellikle can sıkıcı görülmektedir. Günümüz teknolojisi bilgi ve talepleri farklı açılardan ele alarak olası müşterinin değerini artırmayı amaçlayan pazarlama yöntemlerini yeniden düşünmemizi gerektirmektedir, böylece iş talepleri doğrultusunda daha uzun ve sıkı ilişkilerin kurulması sağlanabilmektedir.

1.2. Makine Öğrenimi Nedir? Ne İşe Yarar?

Bilgisayarın açık programlama olmadan bir veri kümesini kullanarak edindiği öğrenme yeteneğine *Makine Öğrenimi*¹ denmektedir. Bilgisayar tabanlı bu analiz yaklaşımı Şekil 1.1’de kabaca tasvir edilmektedir.



Şekil 1.1. Sıradan bir ML algoritmasının genel akış şeması.

Gelişen teknoloji olanakları global veri hacminin günden güne katlanarak artmasına neden olmaktadır. Diğer yandan, bilgiyi biriktirmenin görece kolay onu depolamanın ise kayda değer ölçüde külfetli olmadığı bir çağda yaşadığımız aşikardır. Depolanan küresel veri miktarı üzerine yapılan bir araştırmada bilgi hacminin her yirmi ayda bir ikiye katlandığı öne sürülmüştür (Piatetsky-Shapiro ve Frawley, 1991). Makine tarafından okunabilen veri miktarı arttıkça onu anlama ve kullanma

¹ Bu kavram literatürde “Machine Learning” olarak bilinmekte olup kısaca “ML” şeklinde gösterilmektedir. Literatür ile uyumlu olmak adına tezin bundan sonraki kısımlarında “Makine Öğrenimi” kavramı “ML” biçiminde kısaltılacaktır.

yeteneğimiz ne yazık ki bu büyümeye aynı oranda ayak uyduramamaktadır. Bununla birlikte, bilgisayar sistemleri bize büyük miktarda bilginin otomatik olarak nasıl analiz edilebileceği hususunda kullanışlı araçlar sunabilmektedir. Örneğin, ML yaklaşımlarının temeli öznitelik (ya da özellik) seçimidir. Öznitelik seçimi görevi sayesinde göze çarpan veriler anlamlı biçimde yeniden betimlenir ve bir öğrenme algoritmasının eldeki verileri başarıyla analiz edebileceği ve tutarlı öngörüler sunabileceği faydalı durumlara yönlendirilmesi sağlanır. İşletme bilgi yönetimi bağlamında ele alındığında, *Büyük Veri*² (Buhl ve ark., 2013) müşterilerinin eğilimlerine odaklanan firmaların doğru pazarlama kararları almalarına ve yeni stratejiler geliştirmelerine ciddi katkılar sunabilmektedir.

Günümüzde yoğun bilimsel verilerin üretildiği kaynaklardan bir tanesi de Avrupa Nükleer Araştırma Merkezi'ndeki³ büyük hadron çarpıştırıcısı deneyleri olup buradaki araştırmalar 40 TB/s hızında sürekli bir veri akışı sunmaktadır (Krawczyk ve ark., 2022). Toplanan veriler bir dizi istatistik işlemlerden geçirilerek işlenebilir biçimde düzenlenir ve sonrasında çeşitli karşılaştırma metotları aracılığıyla ele alınır. Böylece, ilgili veri havuzundaki öznitelikler arasındaki ilişkiler araştırılır ve varsa bağlantıları gün yüzüne çıkartılır. Veri havuzundaki bazı öznitelikler ve istatistik analiz parametreleri üzerinde bir dizi özel ayarlamaların yapılmasından sonra gerçekleştirilen simülasyonlar yardımıyla farklı kararlara verilebilecek olası tepkileri görmek mümkün olabilmektedir. Büyük veri analizi yardımıyla firmalar müşterilerinin eğilimlerine göre topladıkları verileri bazı yöntemlerle inceleyerek maksimum fayda sağlayan bir istatistiki araç haline dönüştürebilmektedir.

Bir veri havuzunda bulunan örneklemelerin kaydedilen özelliklerinin sayısı seçilen veri hacmindeki agresif genişleme durumunda beklendiği üzere ciddi oranda artacaktır. Kayıt alanındaki dikkate değer artış literatürde *boyutsallık laneti*⁴ (Verleysen ve Francois, 2005) olarak bilinmektedir. Marron ve çalışma arkadaşı de Maine bir çeşit veri sıkıştırma yöntemiyle bu sorunun aşılabileceğini düşünmüş (Marron ve de Maine, 1967), ancak ilerleyen yıllarda veri sıkıştırmanın bazı verilerin kaybolmasına neden olduğu anlaşılmıştır (Heavens ve ark., 2000; Zhaoping, 2006; Suarjaya, 2012). Yapay Zekâ⁵ aracılığıyla yapılan veri analizlerinin son yıllarda önemli bir popülerlik kazanmasıyla birlikte bilim dünyasında öğrenebilen, planlayabilen, algılayabilen veya işleyebilen insan beynini modelleyebilmek için dikkate alınmaktadır (Adek and Ula, 2020). Bu ilginç olgu temel olarak görsel algılama, konuşma tanıma, karar verme ve dil çevirisi gibi insan zekâsına ihtiyaç duyan görevleri

² Bu kavram literatürde “Big Data” olarak bilinmektedir.

³ Okuyucunun kısaca CERN olarak bilinen araştırma merkezi hakkında ayrıntılı bilgiye www.home.cern adresinden ulaşması mümkündür.

⁴ Bu sorunun terminolojide bilinen adı “the curse of dimensionality” biçimindedir.

⁵ Bu kavram literatürde “Artificial Intelligence” olarak bilinmektedir ve kısaca “AI” biçiminde ifade edilmektedir. Alan terminolojisiyle uyumlu olmak için tezin bundan sonraki kısımlarında “Yapay Zekâ” kavramı “AI” biçiminde kısaltılacaktır.

gerçekleştirebilen bilgisayar mimarilerinin geliştirilmesidir (Rincy ve Gupta, 2020). ML, Sinir Ağları⁶ ve Derin Öğrenme⁷ kavramları bazen birbirinin yerine kullanılmaktadır, ancak aralarında bazı nüanslar olduğunu vurgulamak yerinde olacaktır. Bunlar AI'nın birer alt dalı olmakla birlikte esasen DL kavramı NNs kavramının bir alt dalı iken NNs yaklaşımı ise ML'nin bir alt dalıdır. AI yaklaşımı sadece işletme bilgi yönetimi (Bose ve Mahapatra, 2001) alanında değil finans (Dixon ve ark., 2020), meteoroloji (Stirnberg ve ark., 2021), nörobilim (Vogt, 2018), hesaplamalı biyoloji (Chicco, 2017), kuantum kimyası (Ramakrishnan ve von Lilienfeld, 2017), uzay aracı mühendisliği (İbrahim ve ark. 2019), astrofizik (Ntampaka ve ark., 2017; Caldeira ve ark., 2019; Salti ve ark., 2021; Salti ve Kangal, 2022) ve kozmoloji (Kangal ve ark., 2019; Escamilla-Rivera ve ark., 2020; Tilaver ve ark., 2021) gibi son derece farklı araştırma alanlarında da bilgi düzeyimizi artırmaya yönelik dikkate değer potansiyele sahiptir.

Sıradan bir ML yaklaşımı temel olarak gerçek dünya problemlerini betimleyen matematiksel denklemlere/fonksiyonlara odaklanmakla ilgilidir. Bu matematiksel denklemler/fonksiyonlar bazen *matematiksel modeller* bazen de sadece *modeller* olarak göz önüne alınmaktadır. Herhangi bir ML mimarisinin performansının kalbi öngörüdür. Temel ML görevlerinden öne çıkan bazılarını sıralamak gerekirse aşağıdaki gibi genel bir liste oluşturmak mümkündür (Zhang, 2010; Sharma ve Kumar, 2017; Jain ve ark., 2020; Mahesh, 2020):

- Uyarlama,
- Kümeleme ya da gruplandırma,
- Regresyon,
- Sınıflandırma,
- Örneklem ve sentez,
- Anormallik/Aykırlık/Kuralsızlık tespiti,
- Makine çevirisi,
- Benzerlik eşleştirme,
- Nedensel modelleme,
- Olasılık yoğunluğunun tahmini,
- Olasılık kütle fonksiyonunun tahmini,
- Eş oluşum kümelemesi,
- Bağlantı profili kurma.

⁶ “Sinir Ağları” kavramı literatürde “Neural Networks” olarak bilinmekte olup “NNs” şeklinde kısaltılmıştır. Tezin bundan sonraki kısımlarında literatüre bağlı kalmak adına “Sinir Ağları” kavramı “NNs” biçiminde kısaltılmış olarak kullanılacaktır.

⁷ Dilimizde “Derin Öğrenme” olarak bilinen kavramın literatürdeki karşılığı “Deep Learning” biçimindedir ve “DL” olarak kısaltılmıştır. Burada motive olarak, araştırmamızın bundan sonraki kısımlarında “Derin Öğrenme” kavramı “DL” biçiminde kısaltılmış halde kullanılacaktır.

Bazıları yukarıda listelenen ML uygulamaları çeşitli geliştirme fazlarından oluşan bir dizi işlem sonrasında yerine getirilebilmektedir. Bu bağlamda, ML yaklaşımlarının temel geliştirme evreleri aşağıda sıralanmıştır:

- Verileri toplama,
- Seçilen veri kümesinin ön-işleme evresinden geçirilmesi,
- Keşif bazlı veri analizi,
- Özellik oluşturma/çıkarma,
- Öznitelik seçimi,
- Boyut azaltma,
- Algoritma seçimi,
- Mimari eğitimi,
- Öğrenimi test etme,
- Verileri eşleştirme,
- Performansı raporlama,
- Olumsuz durumlarda makinenin farklı yöntemlerle tekrar eğitilmesi.

1.3. Öznitelik Seçimi ve Örneklem

ML yaklaşımları genel olarak öğrenilen bilgiyi betimlemekte kullanılan kodlamalar ile karakterize edilmektedir. Literatürde tamamlanan araştırmalar tek bir öğrenme algoritmasının her durumda başarılı olamadığını ve farklı öğrenme yaklaşımlarının bazen çok yakın çıkarımlar üretebildiğini ortaya koymuştur (Langley ve Simon, 1995). Bir öğrenme yaklaşımının performansına dikkate değer düzeyde etki edebilecek kritik faktör öğrenilecek görevi karakterize eden verilerin doğasıdır, zira ele alınan veri havuzu seçilen ML yaklaşımının üzerine inşa edildiği istatistiksel düzenliliği sunamazsa öğrenme başarısız olacaktır. Öğrenmeyi kolaylaştırabilecek biçimde ham verilerden yeni verilerin sentezlenmesi esasen mümkündür, ancak tam otomatik bir metodun zorlayıcı olması bu işin karmaşılaşmasına neden olmaktadır (Thornton, 1992). Diğer yandan, ham veri kümeleri sunulan ML algoritmaları ile çalışırken alakasız/gereksiz özniteliklerin veri grubundan kaldırılması ilgili veri kümelerine gizlenmiş düzen örgüsünü keşfetme görevini daha kolay ve daha az zaman alıcı hale getirilebilmektedir. Bu ilginç görev alan terminolojisinde *öznitelik seçimi*⁸ olarak bilinmektedir. Yapay veri kümelerini sentezleme sürecine nazaran öznitelik seçimi araçları başarılı biçimde tanımlanmıştır ve bu iş için kurgulanmış modeller tam otomatik ve hesaplamalı olarak izlenebilir bir süreç olma potansiyeline sahiptir. Öğrenme için öznitelik seçiminin önemli faydaları kabaca şöyle sıralanabilir:

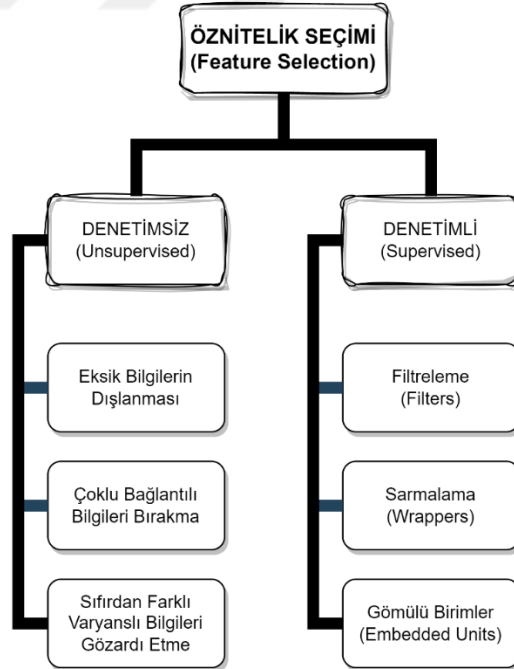
- Daha düzenli bilgi kümesi,
- Öğrenmeyi gerçekleştirmek için gereken veri miktarında azalma,

⁸ Literatürde bilinen adıyla “Feature Selection”.

- Görece daha kolay anlaşılır bilgi yığını,
- Geliştirilmiş tahmin tutarlılığı,
- Azaltılmış uygulama süresi,
- Düşürülmüş bilgisayar donanımı kapasitesi.

Son üç durum olan “geliştirilmiş tahmin tutarlılığı”, “azaltılmış uygulama süresi” ve “düşürülmüş bilgisayar donanımı kapasitesi” endüstriyel ve ticari *Veri Madenciliği*⁹ alanında özel öneme sahiptir. Depolama aygıtlarının maliyet ve boyutlarının düşmesinin yanında kapasitelerinin inanılmaz ölçüde artmasıyla birlikte kurumsal ve endüstriyel veri tabanlarının boyutları da artmıştır, öyle ki bu veriler hakkında eşzamanlı (paralel) çalışan özel donanım sistemlerinde işlem yapan paralelleştirilmiş ML mimarileri dışında herhangi bir yöntemle analiz yapmak artık mümkün olmamaktadır (John ve Langley, 1996). Standart bir ML yaklaşımının büyük veri gruplarına uygulanmasında öznitelik seçimi araçları verilerde bulunan en göze çarpan örüntüleri belirleyerek ayırım yapılmasına olanak sağlarken veri örnekleme araçları ise incelenen veri grubunun en iyi temsil örneklerini ortaya çıkararak yapay veriler sentezlenmesini sağlar (John ve Langley, 1996).

Bu tez çalışması öğrenilebilecek veri setinin büyüklüğüne/miktarına/karmaşıklığına bakılmaksızın öğrenme algoritmalarına yararlı olabilen bir aşama olan öznitelik seçimine özellikle dikkat çekmektedir. ML algoritmalarında kullanılan öznitelik seçme metotları tipik olarak “denetimli” olanlar ve “denetimsiz” olanlar biçiminde iki alt kategoriye ayrılmaktadır. Şekil 1.2’de öznitelik seçimi yöntemleri ve alt grupları sıralanmıştır.



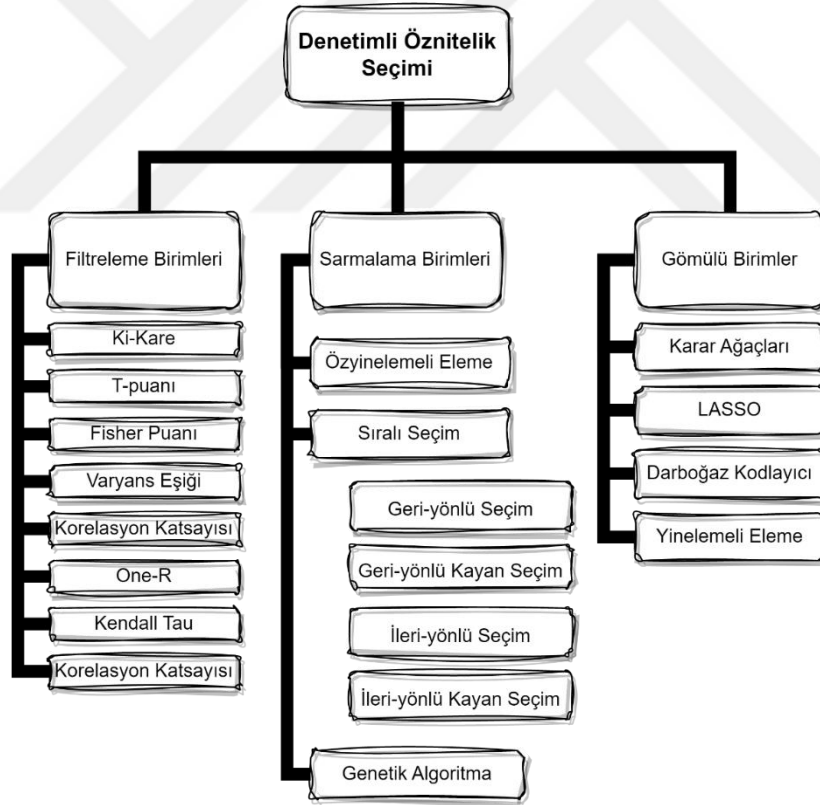
Şekil 1.2. Öznitelik seçim yöntemleri.

⁹ “Veri Madenciliği” sıra dışı desenler (örüntüler, paternler) ve ilişkileri ortaya çıkarmak amacıyla büyük veri tabanlarını araştırma sürecini betimleyen bir terim olup terminolojide “Data Mining” olarak bilinir.

Bilindiği üzere tez araştırmamız kapsamında bir tele-pazarlama veri kümesi analiz edilerek ilgili firmanın satışlarını etkileyen temel faktörlerin belirlenmesi hedeflenmiştir. Bu bağlamda araştırmamız kapsamında denetimli ML algoritmalarının kullanılması faydalı olacaktır. Şekil 1.2’den de görüleceği üzere denetimli öznelik seçimi metotları tipik olarak üç alt sınıfa ayrılmaktadır (Kohavi, 1995; Kohavi ve John, 1996; Thomas ve ark., 2006; Chandrashekar ve Sahin, 2014):

- Filtreler¹⁰: Verilere uygulanacak öğrenme algoritmasını kullanarak özneliklerin değerini ölçenler,
- Sarmalayıcılar¹¹: Özelliklerin doğasını genel veri özelliklerine dayalı belirleyici yöntemler kullanarak değerlendirenler,
- Gömülü birimler¹²: Bir öğrenme algoritmasının amaç işlevini ya da performansını optimize etmek için kullanılanlar.

Her üç kategoride de algoritmalar değerlendirme işlevlerinin karakterine ve öznelik alt küme uzayının oluşturulma biçimine göre daha da farklılaştırılabilmektedir. Şekil 1.3’te yukarıda bahsi geçen üç kategoriden önemli örneklerin bir listesi sunulmaktadır (Kohavi, 1995; Kohavi ve John, 1996; Thomas ve ark., 2006; Chandrashekar ve Sahin, 2014).



Şekil 1.3. Denetimli öznelik seçiminde önemli modeller.

¹⁰ Terminolojideki karşılığı “Filters” biçimindedir.

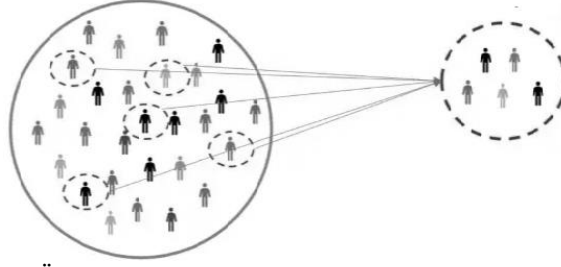
¹¹ Literatürde “Wrappers” olarak bilinmektedirler.

¹² Bu kavram alan terminolojisinde “Embedded Units” olarak anılmaktadır.

Denetimli öznitelik belirleme seçilen öğrenme mimarisine göre optimize edildiğinden sarmalayıcılar (ya da daha kısa bir biçimde sarıcılar) çoğunlukla algoritmanın filtreleme birimlerinden daha iyi sonuçlar üretmektedir. Bununla birlikte, dikkate alınan her bir öznitelik grubunu çalışmak için bir öğrenme yaklaşımı kullanıldığından sarmalayıcıların çalıştırılması masraflı (zaman, iş gücü, bilgisayar donanımı vs. bağlamında) olabilirken bu araçlar birçok özneliğe sahip büyük veri kümelerinde zorluklar ile karşılaşmamıza da neden olabilmektedirler. Öte yandan, öznitelik seçim süreci bir öğrenme mimarisi ile yakından bağlantılı olduğundan sarmalayıcılar algoritma filtrelerine göre daha az geneldir, ancak bir öğrenme yaklaşımından diğerine geçerken yeniden çalıştırılmaları gerekmektedir. Literatürden öznitelik seçimine yönelik filtreleme yaklaşımlarının avantajlarının dezavantajlarından daha fazla olduğu bilinmektedir ve genel olarak filtreleme modelleri sarmalayıcı yaklaşımlarına göre daha hızlı çalışmaktadır (Kohavi, 1995; Kohavi ve John, 1996). Her ne kadar filtreleme modellerinin çok sayıda özneliğe sahip veri tabanlarında ölçeklendirme şansı sarmalayıcılara oranla çok daha yüksek olsa da sarmalayıcı yöntemler "gerçek" sorun olan "sınıflandırıcı performansını optimize etmek" işini daha başarılı çözmektedir. Üçüncü kategoriyi oluşturan gömülü yöntemler bir öğrenme yaklaşımının hedef performansını optimize etmek için de kullanıldıklarından sarmalayıcı yöntemlere oldukça benzemekte olup (Thomas ve ark., 2006; Chandrashekar ve Sahin, 2014) sarmalayıcı yöntemlere göre en önemli fark öğrenme sırasında içsel bir model oluşturma metriği (bileşeni) kullanmalarıdır.

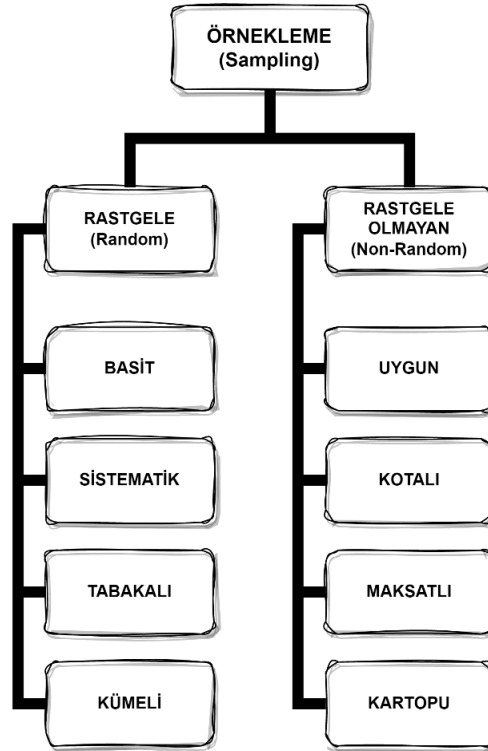
Analiz edilmesi gereken büyük bir veri kümesi olduğunda ve araştırmaya başlamak için heyecan duyuyorken bazen garip ve oldukça can sıkıcı olan bir durum aniden karşımıza çıkar: makinemiz veri kümesini yüklemeye çalışırken "bellek yetersiz" hatası verir! Büyük bir veri seti söz konusu olduğunda bu çoğu zaman karşılaşılan bir durumdur. Hesaplama açısından sınırlı donanıma sahip makineler kullanarak büyük miktarda veriyle uğraşmak veri biliminde karşılaşılan ciddi bir durumdur ve elbette bu sorun ek kaynak gücüyle çözülebilmektedir. Peki ya ek kaynak olanağı yoksa! Bu durumda yetersiz bellek sorunu nasıl aşılabılır? Verileri başarıyla temsil edebilecek bir alt küme belirlemenin ve onu analiz etmenin bir yolu var mıdır? Bunu yapmanın ana sonuca ulaşmak adına bir sakıncası var mıdır? Bu alt küme gerçekten tüm veri havuzunun iyi bir temsili olabilir mi? İşte tam da bu noktada yukarıdaki sorulara yanıt verebilecek *örnekleme*¹³ adı verilen ve büyük veri setleriyle başa çıkılmasına yardımcı olabilen bir istatistiksel yaklaşım ön plana çıkmaktadır. Şekil 1.4'te örnekleme yaklaşımı için temsili bir betimleme sunulmaktadır.

¹³ Terminolojideki karşılığı "sampling" biçimindedir. Her bir elemanını ayrı ayrı ele almak zorunda kalmadan, bir grubun örnek bir alt kümesinden yola çıkıp istatistiklere dayalı olarak ilgili ana grup hakkında bilgi alınmasını sağlayan yöntem "örnekleme" denir.



Şekil 1.4. Örneklemeye temsili bir örnek (Hossain, 2020).

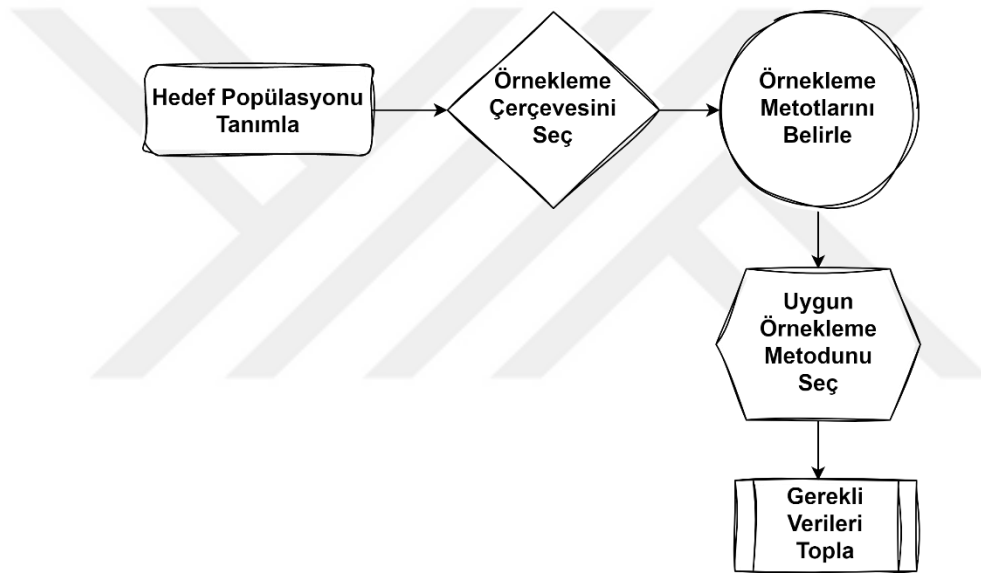
Ele alınan veri havuzuna yönelik araştırma sorusunun ne olduğu öğrenildikten sonra öne çıkan etken analiz faktörü araştırma yaklaşımını belirlemektir. Bu aşamayı ise “araştırmanın popülasyonunu belirlemek” ve “örnekleme seçmek” görevleri takip edecektir. Uygun (ya da doğru) örnekleme yöntemini seçmek istatistiksel bir analiz için hayati önem taşımaktadır, zira o araştırmanın başarısını (veya başarısızlığını) belirleyen bu aşamadır (Gangwal, 2019; Hossain, 2020). Yanlış örneklem seçimi ele alınan araştırma (ya da değerlendirme) sorusunda ulaşılması hedeflenen doğru bilgiyi sağlamayacaktır. Örneklem özellikle tam verimli analizi yapılamayacak ölçüde büyük veri havuzlarında (örneğin büyük veri analitiği uygulamalarında veya anketlerde) oldukça yararlı olmaktadır, çünkü temsili bir kümeyi ele alarak onu analiz etmek veri havuzunun (veya popülasyonun) tamamının araştırılmasından daha kolay ve uygun maliyetlidir. Diğer bir deyişle, tam ya da eksiksiz veri setlerini analiz etmeyle kıyaslandığında örnekleme temelli yaklaşımların daha yüksek analiz hızı gibi önemli bir avantajı vardır. Bütün örnekleme yöntemleri iki genel kategori altında toplanabilmektedir: olasılıkçı ve olasılık-dışı modeller. Şekil 1.5’te en çok tercih edilen yöntemler listelenmiştir.



Şekil 1.5. Sık kullanılan örnekleme yöntemleri.

Örnekleme yaparken popülasyonun parçası olan verilerin seçileceği yöntemin dikkatlice tanımlanması gerekmektedir (Gangwal, 2019; Suresha, 2021). Zira, örnek kümeyi kullanarak hakkında tahminler yürütülmek istenen ana veri setinin sahip olduğu parametreleri yorumlama işi çok daha kolay ve başarılı biçimde yerine getirilebilmektedir. Bir veri kümesinde örnek denekler (ya da elemanlar) belirlenirken dikkate alınması gereken temel ve önemli hususlar şunlardır (Gangwal, 2019; Hossain, 2020; Suresha, 2021):

- Örnek hedef: Temsil grubunun kullanılmasıyla tahmin edilmek istenilen popülasyon özelliğidir,
- Popülasyon (ya da nüfus): Gözlemlerin kuramsal olarak yapılabileceği kapsam veya alandır,
- Seçim kriterleri: Örnek gruptaki gözlemleri kabul etmek veya reddetmek için kullanılacak metodoloji ya da yöntemler grubudur,
- Örnek boyut: Örneği oluşturacak gözlem sayısıdır.



Şekil 1.6. Örnekleme çerçevesinin ana basamakları.

Şekil 1.6’da gösterildiği gibi sıradan bir örnekleme çerçevesinde beş ana basamak bulunmaktadır (Gangwal, 2019; Suresha, 2021):

- Basamak 1: Hedef popülasyonu tanımla!
 - Örnekleme sürecindeki ilk aşama hedef veri setini/kümesini açık bir şekilde tanımlamaktır.
- Basamak 2: Örnekleme çerçevesini seç!
 - Örnek kümenin alındığı veri havuzunu oluşturan elemanların/öğelerin/kişilerin düzenlenmiş bir listesidir.
- Basamak 3: Örnekleme metotlarını belirle!
 - Genel olarak olasılıklı örnekleme yöntemleri tercih edilmektedir, çünkü her seçim eşit değere sahiptir. Örneğin veri havuzu bileşenleri insanlar olsaydı grup

elemanlarının ırkına, cinsiyetine, topluluğuna ya da dinine bakılmaksızın herhangi bir kişi örneklem grubuna dahil edilebilecektir. Bir seçim anketi söz konusu olsaydı ülke genelinde farklı bölgelerden farklı örneklerin alınması daha tutarlı sonuçların elde edilmesini sağlayacaktır.

- **Basamak 4:** Uygun örnekleme metodunu seç!
 - Uygun örnekleme metodu belirlenirken ön plana çıkan etken faktör örnek grup boyutudur. Örneklem boyutu basitçe ifade etmek gerekirse istenilen doğruluk ve/veya kesinlik düzeyinde çıkarımlarda bulunmak için yeterli olacak birey/eleman sayısıdır. Örnek veri kümesinin boyutu ne kadar büyük olursa ham veri havuzu hakkındaki çıkarımlar da beklenildiği üzere daha doğru olacaktır. Örneğin bir ülkedeki seçimler öncesinde yapılan araştırmalarda anket firmaları bir siyasi partinin kazanabileceği meclis sandalye sayısını tahmin ederken faydalı olacağını düşündüğünden mümkün olduğunca çok bireyi örneklem havuzuna dahil etmek isteyecektir.
- **Basamak 5:** Gerekli verileri topla!
 - Ham veri havuzu incelenerek (ve gerekiyorsa grafiksel analizler de yapılarak) gizli kalmış ilave bilgiler de dahil olmak üzere veri setinin özellikleri düzenlenir. Biraz daha ayrıntılı açıklamak adına yine seçim anketlerini ele alalım. Kamuoyu yoklamalarında anket firmalarının oy kullanacak bireylere hangi siyasi partiye oy vereceklerini, bir önceki dönemde hangi partiye oy verdiklerini ve kesinlikle oy vermeyi düşünmedikleri bir partinin olup olmadığını sorması genellikle faydalı olmaktadır. Amaç toplumun seçimler öncesindeki eğilimi hakkında veri toplamak ve seçim sonuçlarına yönelik en başarılı tahmini yürütmektir.

1.4. Veri Dengeleme ve Sınıflandırma

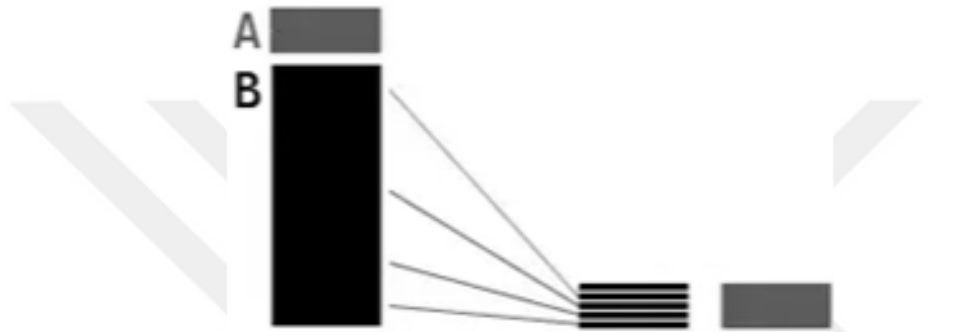
Dengesiz veri havuzu sorunu olarak bilinen kategori dengesizliği veri bilimi araştırmaları kapsamında dikkatlice ele alınması gereken bir problemdir. Sınıflandırma algoritmalarının çoğu ham veri setlerinin dengeli olduğunu kabul ederek işlem yapmaktadır. Sınıflandırma yapabilme yeteneği olan ML yaklaşımlarında temel hedef *doğru tahmin oranını* (ya da *tutarlılığı*¹⁴) maksimize etmektir. Ancak varsayılan dengeli veri dağılımı her zaman karşımıza çıkan bir durum değildir, aslında gerçek hayattaki veri setlerinde dengeli dağılım bulunmaz demek yerinde olacaktır. Veri havuzunda tanımlanan kategorilerden (sınıflardan) bazıları çok az elemana sahipken, bazıları da çok fazla sayıda eleman ile temsil ediliyor olabilir. Örneğin

¹⁴ Literatürde bilinen adıyla “Accuracy”. Bu kavram dilimize “Aküresi” olarak da çevrilmiştir.

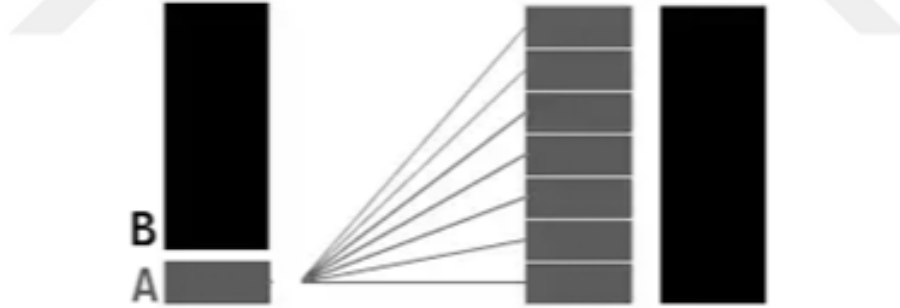
- Bir firmanın satışlarındaki “olumlu” ve “olumsuz” geri dönüşlerin toplam sayıları farklı (yani dengesiz) olabilir,
- Herhangi bir sınıftaki kız ve erkek öğrencilerin toplam sayıları farklı olabilir.

Böyle durumlarda sınıflandırma algoritması ciddi sorunlar üretmektedir. Karakteristik özellik bilgisi az olan veri seti elemanları için algoritma yeterince iyi düzeyde eğitilemediğinden modelin böyle kategoriler için hatalı öngörüler yürütmesi gerçekleşmesi olası sonuçlar arasındadır. Peki bu sorunun üstesinden nasıl geliriz? Temel yöntem kısaca şöyledir:

- Veri kümesindeki bilgilere bakılarak *veri azaltma*¹⁵ ya da *veri çoğaltma*¹⁶ işlemleri (bakınız Şekil 1.7 ve 1.8) sonrasında incelenen veri kategorileri dağılımı dengeli hale getirilir.



Şekil 1.7. Veri azaltma (aşağı örnekleme) yöntemiyle veri dengeleme.



Şekil 1.8. Veri çoğaltma (yukarı-örnekleme) yöntemiyle veri dengeleme.

Bu işlemler için çeşitli algoritma araçları inşa edilmiş olup tezin ilerleyen kısımlarında bazıları hakkında bilgiler verilecektir.

Nesnelerin önceden belirlenmiş bir kategori veya sınıf kümesinden birine göre nasıl sınıflandırılacağını öğrenmek psikoloji ve bilgisayar bilimlerindeki araştırmacıların yoğun ilgisini çeken bir zekâ özelliğidir. Kategorilerini temsil eden bir dizi nesnenin ortak "çekirdek" özelliklerinin belirlenmesi bir kişinin veya bilgisayar programının dikkatini odaklamada çok faydalı olmaktadır. Örneğin bir hayvanın zebra olup olmadığını belirlemek için insanlar kuyruğunu veya kulaklarını

¹⁵ Terminolojide “under-sampling” (aşağı-örnekleme) olarak bilinmektedir.

¹⁶ Literatürdeki karşılığı “over-sampling” (yukarı-örnekleme) biçimindedir.

incelemek yerine bedenindeki çizgileri kontrol etmeleri gerektiğini bilir. Bu nedenle beden çizgileri bizim algıladığımız (genelleştirdiğimiz) zebra kavramında güçlü bir şekilde yer almaktadır. Tabii ki kaplanlar için olduğu gibi zebalarda da tek başına şeritler bir sınıf tanımlı oluşturmak için yeterli değildir, ancak kesinlikle önemli özelliklerden bir tanesidir. Sınıflandırma yapabilmek (veya öğrenebilmek) insanlara ve bilgisayar programlarına karar verme gücü vermektedir ve bu kararların etkinliği sınıflandırma görevindeki performanstan doğrudan etkilenmektedir. ML alanında yukarıda açıklanan sınıflandırma görevi genellikle denetimli öğrenme olarak ele alınmaktadır. Denetimli öğrenme yaklaşımında belirli bir veri kümesi vardır ve örnek nesneler uygun kategori ile etiketlenir. Örneğin yukarıdaki örnekler kullanılarak algoritmaya zebranın ve kaplanın ne olup olmadıkları öğretilmektedir. Ana hedef yeni verilerin kategorilerden birine ait olarak tanımlanmasını sağlayacak eğitim nesnelerinden genelleme yapmaktır, yani sınıf tanımları oluşturmaktır. Denetimsiz öğrenme durumunda programa hangi hayvanların kaplan ya da zebra olduğu söylenmez. Genellikle denetimsiz öğrenmedeki amaç hangi nesnelerin birlikte gruplandırılması gerektiğine karar vermektir yani sınıfları eğitilenin kendisi oluşturur. Sınıflandırma öğreniminin başarısı büyük ölçüde eğitim için sağlanan verilerin kalitesine bağlıdır, çünkü bir algoritma yalnızca girdi verilerini öğrenebilir. Veriler yetersiz (veya ilgisiz) olursa sonuç verileri bunu yansıtacak ve yeni verilere uygulandıklarında yanlış sınıflandırmalara neden olacaktır.

1.5. Veri Tipleri, Önemli Veri Kaynakları ve Verilerin Sunumu

Veriler birçok biçimde olabilir, ancak makine öğrenimi modelleri dört temel veri türüne dayanmaktadır (bakınız Şekil 1.9).



Şekil 1.9. Temel veri tipleri.

- Nümerik veri tipi boyunuz, kilonuz veya telefon faturanızın maliyeti gibi ölçülebilir veri biçimidir. Sayıların ortalamasını almaya çalışarak veya bunları artan ya da azalan düzende sıralayarak bir veri kümesinin sayısal olup olmadığını belirlemek mümkündür. Kesin veya tam sayılar (örneğin bir sınıftaki 20 öğrenci) ayrık sayılar olarak ele alınırken, belirli bir aralığa (örneğin aylık %2,5 faiz oranı) girenler ise sürekli sayılar olarak kabul edilmektedir. Bir öğrenme algoritmasında kullanmadan önce bu tür verilerin belirli bir zaman ölçeğine bağlı olmadıklarını esasen yalnızca ham sayılar olduklarını unutmamakta fayda vardır.
- Kategorik veriler tanımlayıcı özelliklere göre sıralanırlar ve cinsiyet, etnik köken, doğum yeri, çalışılan sektör veya çeşitli başka etiketler bu veri tipine örnek olarak gösterilebilmektedir. Bu

veri tipleri ile çalışılırken toplanamayacaklarına, ortalamalarının alınamayacağına veya herhangi bir kronolojik sıraya konamayacaklarına dikkat edilmektedir. Kategorik veriler benzer (ya da farklı) özellikleri paylaşan bireyleri veya fikirleri sınıflandırmak için çok kullanışlıdır ve ML yaklaşımlarında veri analizini kolaylaştırmaktadır.

- Zamansal veri kümeleri belirli zaman noktalarında endekslenen verilerden oluşmaktadır ve çoğu zaman tutarlı aralıklarla toplanırlar. Bu tip verileri ile çalışırken onları haftadan haftaya, aydan aya, yıldan yıla veya arzu edilen herhangi bir zaman dilimi metriğine göre sıralamak veriler arasında karşılaştırma yapmayı kolaylaştıracaktır. Zaman serisi verilerinin bir başlangıç ve bitiş noktası olurken sayısal verilerin belirli zaman dilimlerinde geçerli kalma zorunlulukları yoktur.
- Metin tabanlı veriler ML yaklaşımlarına belirli bir düzeyde öngörü sağlayabilen sözcükler, cümleler veya paragraflardır. Bu verileri ML modellerinin kendi kendine yorumlaması oldukça zordur ve genellikle birlikte gruplandırılırlar. Sözcük tekrarlanma sıklığı, metin sınıflandırması veya duygu analizi gibi çeşitli yöntemler bu tip verilerin analizlerinde ön plana çıkmaktadır.

ML uygulamalarına yönelik verilerinin elde edilebileceği çok sayıda kaynak bulunmaktadır. Bu bağlamda araştırmalara yardımcı olabilecek en popüler veri kümesi kaynaklarına Tablo 1.1’de sunulan örnekleri vermek mümkündür.

Tablo 1.1. Popüler veri havuzu kaynakları.

Kaynak Logosu	Erişim Adresi	Kaynak Adı
	https://www.kaggle.com	Google’s Data Science Community
	https://datasetsearch.research.google.com	Google’s Dataset Research
	https://www.msropendata.com	Microsoft Research Open Data
	https://registry.opendata.aws	Open Data on Amazon Web Services
	https://archive.ics.uci.edu	UCI Machine Learning Repository
	https://data.gov	U.S. Government’s Open Data
	https://www.analyticsvidhya.com	India’s Data Science Community
	https://community.ibm.com	IBM’s AI and Data Science Community

Tablo 1.2. Örnek veri kümesi: golf oynamaya yönelik bilgiler.

Örnek No	Öznitelikler				Kategori
	Görünüm	Sıcaklık	Nem	Rüzgâr	
1	Güneşli	Sıcak	Yüksek	Yok	Oynanmaz
2	Güneşli	Sıcak	Yüksek	Var	Oynanmaz
3	Bulutlu	Sıcak	Yüksek	Yok	Oynanır
4	Yağmurlu	Ilıman	Yüksek	Yok	Oynanır
5	Yağmurlu	Serin	Normal	Yok	Oynanır
6	Yağmurlu	Serin	Normal	Var	Oynanmaz
7	Bulutlu	Serin	Normal	Var	Oynanır
8	Güneşli	Ilıman	Yüksek	Yok	Oynanır
9	Güneşli	Serin	Normal	Yok	Oynanır
10	Yağmurlu	Ilıman	Normal	Yok	Oynanır

Sıradan bir ML analizinde veri kümesi genellikle bir tablo ile temsil edilmektedir. Bu tabloda her örnek grubunu betimleyen bir etiketle birlikte ölçüm sonucu veya özelliği belirtilmektedir. Örnek olması adına Tablo 1.2.’de bir veri seti sunulmuştur. Bu veri kümesinde golf etkinliği için 10 farklı güne ait bazı bilgiler (Quinlan, 1986) olup ilgili tablodaki her satır o günün golf oynamaya uygun olup olmadığına işaret eden kategori etiketlerini ve etkin öznitelikler olan hava görünümü, hava sıcaklığı, havadaki nem durumu ve ortamdaki rüzgâr bilgisi için edinilen verileri içermektedir. Bir ML algoritmasının tipik uygulaması iki örnek grubu gerektirmektedir: *eğitim örnekleri*¹⁷ ve *test örnekleri*¹⁸. Eğitim için kullanılan örneklem grubu öğrenilen bilgi tanımlarını üretmek için kullanılırken öngörülerin doğruluğu değerlendirmek için ayrı bir test örnekleri seti gerekmektedir. Test aşamasında sınıf etiketleri algoritmaya sunulmaz. Algoritma girdi olarak bir test örneği ele alır ve çıktı olarak bir kategori etiketi (tahmin sınıfı) üretir.

1.6. Karar Destek Sistemi¹⁹

Tipik bir DSS yönetsel karar verme aşamasını desteklemek için bilgi teknolojisine odaklanır ve literatürde kişisel ve akıllı gibi çeşitli DSS türleri bulunmaktadır (Moro ve ark., 2014). Sıradan kişisel DSS yaklaşımları nispeten küçük ölçekli durumlarda (örneğin bir yöneticinin karar verme görevi) etkili roller üstlenirken akıllı DSS yöntemleri ise nihai kararları desteklemek için ML algoritmalarını kullanmaktadır (Arnott ve Pervan, 2008). *İş zekâsı*²⁰ bir diğer önemli DSS yaklaşımı olup genel olarak iş verilerinden yararlanarak karar verme görevlerini destekleyen ve veri ambarları/madenciliği gibi bilgi teknolojilerinden oluşan bir şemsiye terim olarak kabul edilmektedir (Turban ve ark., 2011).

¹⁷ Literatürdeki karşılığı “training data” biçimindedir.

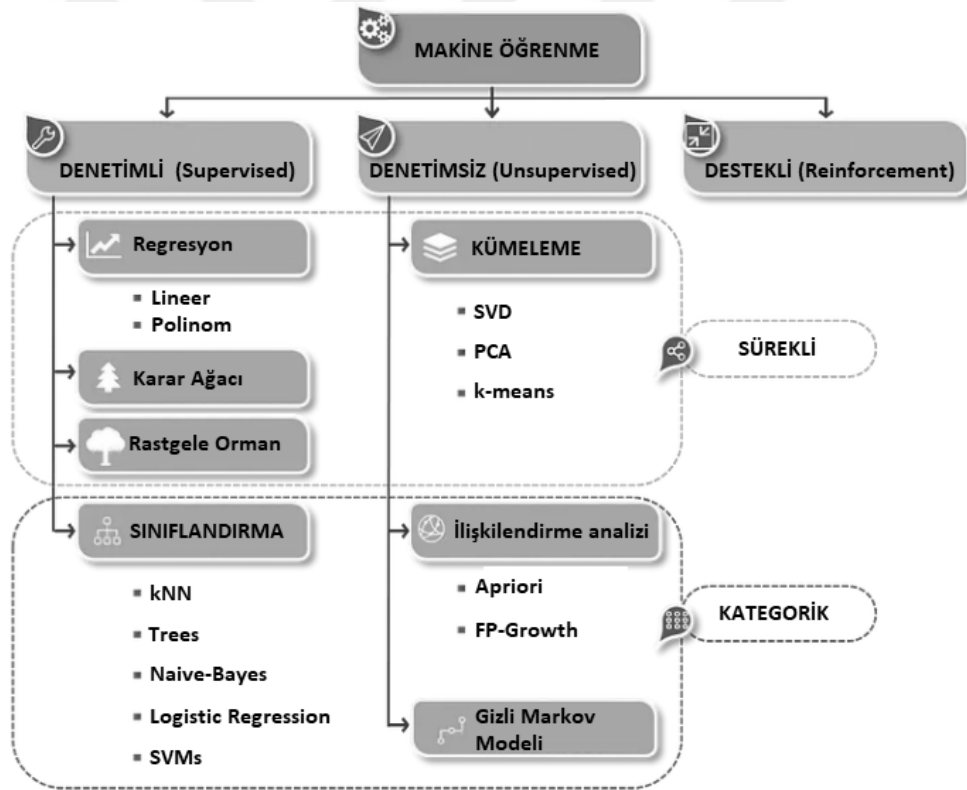
¹⁸ Terminolojide “test data” olarak bilinmektedir.

¹⁹ Bu kavram literatürde “decision support system” olarak anılmakta ve “DSS” olarak kısaltılmaktadır. Bu tez çalışması kapsamında terminolojiye bağlı kalınacak ve “karar destek sistemi” kavramı için “DSS” kısaltması kullanılacaktır.

²⁰ Alandaki karşılığı “business intelligent” biçimindedir.

1.7. Öğrenme yaklaşımları

ML algoritmalarına genel perspektiften bakıldığında denetimli, denetimsiz ve destekli öğrenme biçiminde üç ana kategoriye ayrılma durumu söz konusudur (Donalek, 2011; Benvenuto ve ark., 2018). Bir öğrenme algoritmasından elde edilen kavramsal sonuçlar genellikle öğrenme modelinin eldeki verilerden çıkardığı bilgi veya yaklaşım olarak ele alınmaktadır. Bu bağlamda, bilgi bilinen algoritmalarda farklı yöntemlerle ele alınabilmektedir. Örneğin, (i) *Naif Bayes Sınıflandırıcı*²¹ yaklaşımı bilgiyi olası toplamalar biçiminde inceler (Mitchell, 1997), (ii) *Uzun-Kısa Vadeli Bellek*²² mimarisi tahmin yapmak için anlamlı bilgileri yapay hücreler boyunca taşıyan bir iletişim hattı ve ağ hafızası kullanır (Yu ve ark. 2019), (iii) *Genetik Algoritma* bilgileri evrimsel mekanizmalara benzer evrelerde taşıyarak değerlendirir (Mitchell, 1996; Haupt ve Haupt, 1998), (iv) *C5.0* algoritması bilgiyi karar ağaçları²³ olarak betimler (Quinlan, 1987; Quinlan 1996; Harper, 2005; Hastie ve ark., 2008; Turban ve ark., 2011; Kumar ve Verma, 2012; Kowsari ve ark. 2019; İlham ve ark., 2019). Şekil 1.10’da temel öğrenme yaklaşımları için genel bir ayrım sunulmaktadır.



Şekil 1.10. Makine öğrenme yaklaşımlarına genel bir bakış (Sidhu, 2019).

²¹ Literatürdeki karşılığı “Naive Bayes Classifier” biçimindedir ve “NBC” biçiminde kısaltılmaktadır. Araştırmamızda “Naif Bayes Sınıflandırıcı” yaklaşımı için “NBC” kısaltması kullanılacaktır.

²² Terminolojideki karşılığı “Long Short-Term Memory” biçiminde olup kısaca “LSTM” olarak ifade edilmektedir. Araştırmamızda “Uzun Kısa-Vadeli Bellek” yaklaşımı “LSTM” biçiminde kısaltılarak ele alınacaktır.

²³ Literatürde “Decision Trees” şeklinde anılır ve kısa biçimi “DTs” şeklindedir.

1.7.1. NBC

Bu algoritma incelenen bir örneğin hangi kategoriden olduğuna karar vermek için Bayes formülünün naif (saf ya da basitleştirilmiş) bir biçimini kullanmaktadır (Mitchell, 1997):

$$p(X_i|y_1, y_2, \dots, y_n) = \frac{p(X_i)}{p(y_1, y_2, \dots, y_n)} \prod_{j=1}^n p(y_j|X_i). \quad (1.1)$$

Örneklemin sahip olduğu parametre değerlerine göre her sınıf için nihai dağılım olasılığı hesaplanır ve veri en yüksek olasılığa sahip kategoriye atanır. Yukarıdaki denklem X sınıfının nihai olasılık değerini temsil etmektedir, yani burada verilen öznitelik değerlerini içeren $\langle y_1, y_2, \dots, y_n \rangle$ grubu için sınıflandırılacak özneliliğin gözlemlenen olasılığı hesaplanmaktadır. Denklemin sağ tarafındaki payda genellikle göz ardı edilmektedir, zira esasen sınıfların son olasılıklarının toplamının 1 olması gerektiği koşulundan kolayca hesaplanabilen bir sabittir. NBC ile öğrenmek oldukça basit ve sadedir: eğitim örneklerinden yukarıdaki denklemin sağ tarafındaki olasılıklar kolayca tahmin edilebilmektedir. Burada elde edilen nicelik aslında potansiyel sınıfların her biri için olası özetlerdir ve veri setinde sayısal öznitelikler varsa normal bir dağılımın kabul edilmesi yaygın bir uygulamadır.

Tablo 1.2.'de 10 farklı gün için koşulların golf oynamaya uygun olup olmadığını betimleyen veriler sunulmuştur. Şimdi bu veri setindeki öznitelikler ve sınıflar arasındaki ilişkileri irdelemek için frekans dağılımlarına bakalım. Tablo 1.2.'de sunulan veriler kullanılarak elde edilen golf oynamaya yönelik olasılık durumları Tablo 1.3, Tablo 1.4, Tablo 1.5 ve Tablo 1.6'da ayrıntılı biçimde sırasıyla hava durumuna, hava sıcaklığına, havadaki nem miktarına ve ortamdaki rüzgâr durumuna göre sunulmaktadır.

Tablo 1.3. Hava durumuna göre golf verileri.

	Oynanır	Oynanmaz	Toplam
Güneşli	2	2	4
Yağmurlu	3	1	4
Bulutlu	2	0	2
Toplam	7	3	10

Tablo 1.4. Sıcaklığa göre golf verileri.

	Oynanır	Oynanmaz	Toplam
Sıcak	1	2	3
Ilıman	3	0	3
Serin	3	1	4
Toplam	7	3	10

Tablo 1.5. Nem oranına göre golf verileri.

	Oynanır	Oynanmaz	Toplam
Normal	4	1	5
Yüksek	3	2	5
Toplam	7	3	10

Tablo 1.6. Rüzgâr durumuna göre golf verileri.

	Oynanır	Oynanmaz	Toplam
Var	1	2	3
Yok	6	1	7
Toplam	7	3	10

Şimdi bir sabah uyandığımızı ve günün golf oynamak için uygun olup olmadığını belirlemeyi dilediğimizi hayal edelim. O gün için havanın güneşli, sıcak, normal nemli ve rüzgârsız olduğunu düşünelim. Şimdi, Tablo 1.3'ten Tablo 1.6'ya kadar sunulan olasılıkları Denklem 1.1'de kullanarak her bir kategori için nihai olasılığı hesaplayalım:

$$\begin{aligned}
 p(\text{Oynanmaz}|\text{güneşli, sıcak, normal nem, rüzgar yok}) &= p(\text{Oynanmaz}) \\
 &\times p(\text{Güneşli}|\text{oynanmaz}) \\
 &\times p(\text{Sıcak}|\text{oynanmaz}) \\
 &\times p(\text{Normal nem}|\text{oynanmaz}) \\
 &\times p(\text{Rüzgârsız}|\text{oynanmaz}) \\
 &= \frac{3}{10} \times \frac{2}{3} \times \frac{2}{3} \times \frac{1}{3} \times \frac{1}{3} \\
 &\approx \% 1.5
 \end{aligned} \tag{1.2}$$

Sıradaki adımda ise öyle bir gündeki “Oynanır” seçeneği için olasılık değerini hesaplayalım. Yukarıdaki değerden büyük ya da küçük olması nihai karara ulaşmamızı sağlayacaktır.

$$\begin{aligned}
 p(\text{Oynanır}|\text{güneşli, sıcak, normal nem, rüzgar yok}) &= p(\text{Oynanır}) \\
 &\times p(\text{Güneşli}|\text{oynanır}) \\
 &\times p(\text{Sıcak}|\text{oynanır}) \\
 &\times p(\text{Normal nem}|\text{oynanır}) \\
 &\times p(\text{Rüzgârsız}|\text{oynanır}) \\
 &= \frac{7}{10} \times \frac{2}{7} \times \frac{1}{7} \times \frac{4}{7} \times \frac{6}{7} \\
 &= \% 1.4
 \end{aligned} \tag{1.3}$$

Sonuç açıkça ortaya çıkmıştır! Öyle bir günde golf oynayamazdık, çünkü $p(\text{Oynanır}) = \%1.4$ olasılık değeri aksi durumu betimleyen $p(\text{Oynanmaz}) = \%1.5$ değerinden küçüktür.

Öznitelik değerlerinin veri havuzu içerisinde birbirlerinden bağımsız olduğu varsayımından dolayı NBC yaklaşımının tahmin performansı eğitim verilerindeki gereksiz özniteliklerin varlığından olumsuz etkilenebilmektedir (Langley ve Sage, 1994c). Bununla birlikte, özellikler arasındaki güçlü korelasyonların performansı düşüreceği ancak NBC yaklaşımının verilerde orta düzeyde bağımlılıklar olduğunda yine de çok iyi performans gösterebileceği ispatlanmıştır (Domingos ve Pazzani, 1996).

İlimli bağımlılıkların yanlış olasılık tahminlerine yol açacağı doğrudur, ancak bu olasılıklar hatalı sınıflandırmaya neden olacak kadar "yanlış" değildir (Hall, 1999).

1.7.2 LSTM ve “Bırakma” Aracı

LSTM mekanizması *Konvolüsyonel Sinir Ağı*²⁴ veya *Tekrarlayan Sinir Ağı*²⁵ tarafından sağlanan bazı gizli katmanlar yardımıyla karmaşık veri kümelerini incelemektedir (Ntampaka ve ark., 2015; Yu ve ark., 2019). Bu ilginç algoritma, sınıflandırma ve örüntü tanıma problemlerini analiz ederken oldukça başarılı sonuçlar vermektedir. CNN tabanlı algoritmalar görüntüler gibi uzamsal veriler söz konusu olduğunda RNN tabanlı algoritmalarla göre daha güçlü sonuçlar vermektedir (Krizhevsky, 2009). Her ne kadar RNN tabanlı yaklaşımlar CNN tabanlı algoritmalarla kıyasla daha az özellik uyumluluğu sağlıyor gibi görünse de zamansal (veya sıralı) veriler için çok tutarlı tahminler sunabilmektedir (Keren ve Schuller, 2016).

Uzun ve kısa süreli bellek kullanarak sinyal değişimlerini takip eden LSTM ağı tekrarlayan RNN yaklaşımının genişletilmiş bir biçimi olarak da dikkate alınmaktadır (Urtnasan ve ark., 2020). Tipik bir RNN tabanlı yinelenen LSTM modülü birbirleriyle sıra dışı bir şekilde etkileşime giren NN katmanlarından yararlanılarak oluşturulmaktadır (Kumar ve ark., 2018). Bu algoritmada $z^{(t)}$ giriş verileri ile birlikte önceki yuvadan aktarılan bilgi aracılığıyla gizli katmanlardaki tekrarlayan nöronlar için çıkış verilerini betimleyen $h^{(t)}$ niceliği hesaplanmaktadır (Salti ve Kangal, 2022):

$$h^{(t)} = F[\Psi_{hh} \odot h^{(t-1)} + \Psi_{zh} \odot z^{(t)} + b_h]. \quad (1.4)$$

Yukarıdaki ifadede yazılan F , Ψ_{hh} , Ψ_{zh} ve b_h terimleri sırasıyla aktivasyon fonksiyonunu, gizli katmandan gizli katmana ağırlık matrisini, giriş katmanının gizli katmana ağırlık matrisini ve sapmayı temsil etmektedir. Ayrıca, $\langle \rangle$ işareti bir vektörü, $\odot$ sembolü bir matris çarpımını, t ise zaman adımlarını betimlemektedir. İlk veri noktası için bu yaklaşımda $t=1$ değeri kullanılmaktadır, böylece karşılık gelen çıktıyı (ya da sonuç verisini) betimleyen $T^{(t)}$ aşağıdaki ilişki yardımıyla elde edilebilmektedir:

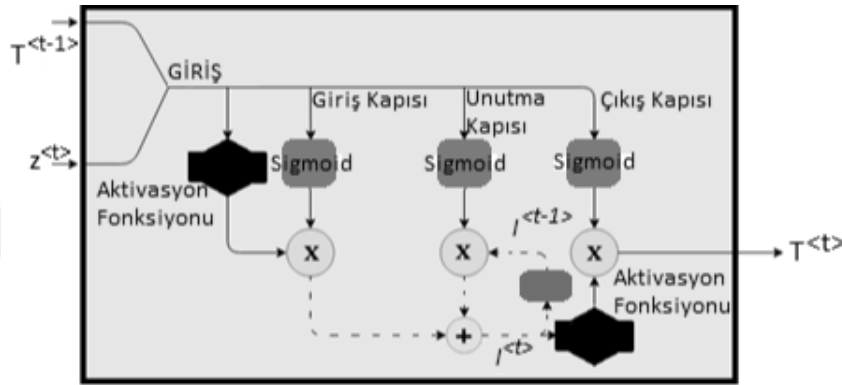
$$T^{(t)} = F[\Psi_{hT} \odot h^{(t)} + b_T]. \quad (1.5)$$

Burada yazılan Ψ_{hT} ve b_T sırasıyla giriş katmanından çıkış katmanına tekrarlayan ağırlıkların matrisleri ve ilgili noktalara karşılık gelen yanılgi miktarıdır. Görüldüğü üzere modül bu süreçte hem t zamanındaki giriş verisini açıklayan $z^{(t)}$ niceliğini hem de önceki durumdan gelen (yani yapay sinir ağlarında aktarılan) bilgiyi kullanarak çalışmaktadır.

²⁴ Literatürde “Convolutional Neural Network” ya da kısaca “CNN” olarak bilinmektedir.

²⁵ Terminolojideki karşılığı “Recurrent Neural Network” ya da kısaca “RNN” biçimindedir.

Sıradan bir LSTM modülünün en dikkat çeken (ya da can sıkıcı) dezavantajı sunduğu öngörülerdeki belirsizlikleri yakalamak için bir mekanizma sağlamamasıdır. Yani algoritma sonuçlarına yönelik hata aralığı vermeden, ilgili çıktıları kabul etmemizi beklemektedir. Her ne kadar tipik bir *Bayesyan Sinir Ağı*²⁶ algoritması belirsizlik hesabı için matematiksel bir çerçeve sağlayabilse de bu yaklaşım engelleyici hesaplama maliyetleri çıkarmaktadır (Cox, 1946; Malakoff, 1999). Öte yandan, BNN temelli algoritmalarının en önemli eksiklikleri arasında *ezberleme*²⁷ ve *uzak-öngörü*²⁸ sorunları da yer almaktadır. Bu ciddi sorunların üstesinden gelebilmek amacıyla yakın zamanda *Bırakma*²⁹ adlı bir istatistik aracı sunulmuştur (Gal ve Ghahramani, 2016).



Şekil 1.11. Tipik bir LSTM modülü diyagramı (Salti ve Kangal, 2022).

Şekil 1.11'de gösterildiği üzere sıradan bir LSTM modülü bir adet ana birim *Hücre*³⁰, bir adet *giriş kapısı*³¹, bir adet *unutma kapısı*³², bir adet *çıkış kapısı*³³, üç adet *sigmoid* tipi kapı aktivasyon fonksiyonu ve iki adet *çıkış aktivasyon fonksiyonu* içermektedir (Graves ve ark. 2009; Sak ve ark. 2014). Hücre olarak anılan birim verileri keyfi zaman aralıklarında çağırırken kapılar ise hücre içine ve hücre dışına veri akışını düzenlemektedir. Bu nedenle, hücreye gömülü olan “ara faz - unutma kapısı” döngüsü bilgi akışında çok önemli bir adım olarak görev almaktadır. Görüldüğü üzere LSTM modülleri ayrıca bir dahili faz değişkeni ($l^{(t)}$) içermektedir ve bu değişkenin bir zaman adımı gecikmeli formu olan $l^{(t-1)}$ etkili bir yineleme katmanı oluşturmak için girdi verilerine eklenmektedir (Ntampaka ve ark., 2015; Yu ve ark., 2019). Burada çarpma yerine toplanmanın kullanılması gradyanların kaybolma riskini

²⁶ Bu yaklaşımın alan terminolojisindeki karşılığı “Bayesian Neural Network” biçiminde olup kısaca “BNN” olarak anılmaktadır.

²⁷ Kavramın literatürdeki karşılığı “over-fitting” biçimindedir.

²⁸ Terminolojide “under-fitting” olarak bilinmektedir.

²⁹ Bu kavram ilgili kaynak çalışmada “Dropout” şeklinde anılmaktadır.

³⁰ Birimin literatürdeki adı “Cell”’dir.

³¹ Kavram “input gate” biçiminde bilinmektedir.

³² Birim terminolojide “forget gate” olarak adlandırılmaktadır.

³³ Literatürde “output gate” şeklinde anılmaktadır.

azaltmaktadır. Yanıtlardan elde edilen bilgilerin zaman içinde iletildiği LSTM-Bırakma modülündeki birimler matematiksel olarak şöyle betimlenmektedir (Salti ve Kangal, 2022):

$$i^{(t)} = \Sigma[\Psi_{zi} \odot z^{(t)} + \Psi_{hi} \odot h^{(t-1)} + b_i] \odot m_i, \quad (1.6)$$

$$f^{(t)} = \Sigma[\Psi_{zf} \odot z^{(t)} + \Psi_{hf} \odot h^{(t-1)} + b_f] \odot m_f, \quad (1.7)$$

$$l^{(t)} = F[\Psi_{zl} \odot z^{(t)} + \Psi_{hl} \odot h^{(t-1)} + b_l] \odot m_l, \quad (1.8)$$

$$c^{(t)} = (f^{(t)} \otimes c^{(t-1)} + i^{(t)} \otimes l^{(t)}) \odot m_c, \quad (1.9)$$

$$o^{(t)} = \Sigma[\Psi_{zo} \odot z^{(t)} + \Psi_{ho} \odot h^{(t-1)} + b_o] \odot m_o, \quad (1.10)$$

$$h^{(t)} = (o^{(t)} \otimes Fc^{(t)}) \odot m_h. \quad (1.11)$$

Yukarıdaki denklemlerde yazılan $\otimes$ sembolü doğrudan matris çarpımını göstermektedir. Ayrıca, m_i , m_f , m_l , m_c , m_o ve m_h nicelikleri sırasıyla giriş kapısı, unutma kapısı, olası yeni durum, değerlendirme katmanı, hücre çıkış kapısı ve nihai veri için İkili-Bırakma maske vektörlerine karşılık gelmektedir. Son olarak, sigmoid fonksiyonu ise şöyle tanımlanmaktadır:

$$\Sigma = \frac{1}{1+e^{-z}}. \quad (1.12)$$

Denklem (1.6) modülün hücre biriminin hangi verilerin uzun vadede hatırlanması gerektiğini anlamasını sağlayan $i^{(t)}$ giriş kapısını ifade etmektedir. Hücre biriminin belirli verileri unutmasını sağlayan ve unutma kapısı olarak bilinen birimi betimleyen $f^{(t)}$ niceliğinin matematiksel tanımı denklem (1.7) yazılmıştır. Algoritmada yeni bilgi durumları denklem (1.8) ile temsil edilmektedir. Denklem (1.9) bir null (sıfır) vektör verdiğinde modül önceki durum hakkında hiçbir şey hatırlamayacaktır. Öte yandan, denklem (1.9) sonuç olarak 1 değerini verdiğinde ise algoritma her şeyi hatırlayacaktır. Çıkış kapısını betimleyen $o^{(t)}$ niceliği denklem (1.10) ile ifade edilir ve bir bilginin çıkış verisi olup olmayacağını belirlemektedir. Son olarak, denklem (1.11) ise LSTM-Bırakma modülünün nihai çıkışı bilgisini temsil etmektedir.

Optimize edici³⁴, aktivasyon fonksiyonu, bırakma olasılığı, yapay nöron sayısı, parti boyutu³⁵ ve dönem sayısı³⁶ tahmin sürecini derinden etkileyerek algoritmayı başarılı veya başarısız hale getirebildikleri için LSTM-Bırakma benzeri derin yapılı sinir ağı modüllerinin çok önemli hiper parametreleri (bileşenleri) olarak bilinmektedirler. Ne yazık ki sıradan bir LSTM-Bırakma derin yapılı makine öğrenimi algoritması kendi başına en uygun hiper parametre seçimini belirleyememektedir. Bir dizi ön-işleme uygulamaları sonrasında bu parametreler için en uygun seçim bizzat analizi yapan kişi tarafından bilgisayara öğretilmelidir ya da verilmelidir:

- Optimizer:

LSTM-Bırakma mimarisi esas olarak gerçek veri ile tahmin edilen çıktı arasındaki farkı azaltmayı amaçlar ve bu görevi maliyet³⁷ fonksiyonu yardımıyla başarmaktadır (Goodfellow ve ark., 2016). Bu amaçla, optimize edici ağırlık parametrelerini güncellerken maliyet fonksiyonu ise optimize ediciye modülün tutarlılığını iyileştirmenin doğru yolunu göstermektedir. Literatürdeki

- Nesterov İvmelendirilmiş Gradyan³⁸ (Nesterov, 1983),
- Uyarlanabilir An³⁹ (Kingma ve Ba, 2015),
- Nesterov İvmelendirilmiş Uyarlanabilir An Tahmini⁴⁰ (Dozat, 2016),
- Adagrad (Duchi, 2011),
- AdaDelta (Zeiler, 2012),
- AdaMax (Kingma ve Ba, 2015),
- AMSGrad (Reddi ve ark., 2018)

modelleri iyi bilinen optimize edici varyantlarından bazılarıdır.

- Aktivasyon Fonksiyonları:

Sıradan bir LSTM-Bırakma modelinde çeşitli aktivasyon fonksiyonları kullanılabilmektedir. En çok kullanılan formülasyonlar şunlardır:

- Doğrultucu Doğrusal Birim⁴¹ (Nair ve Hinton, 2010),
- Sızmalı Doğrultucu Doğrusal Birim⁴² (Maas ve ark. 2013),

³⁴ Bu birim literatürde “Optimizer” şeklinde anılmaktadır.

³⁵ Kavramının terminolojideki karşılığı “Batch size” biçimindedir.

³⁶ Literatürde “number of epochs” ya da “epoch number” olarak bilinmektedir.

³⁷ “Maliyet” kavramının alandaki karşılığı “Cost” biçimindedir.

³⁸ Bu modelin terminolojik adı “Nesterov Accelerated Gradient” şeklindedir.

³⁹ “Adaptive Moment” modelinin dilimizdeki karşılığı “Uyarlanabilir An” olup bu model literatürde “ADAM” biçiminde kısaltılmış olarak betimlenmektedir.

⁴⁰ “Nesterov Accelerated Adaptive Moment Estimation” modelinin dilimizdeki karşılığı “Nesterov Hızlandırılmış Uyarlanabilir An Tahmini” olup bu model literatürde “NADAM” biçiminde kısaltılmış olarak betimlenmektedir.

⁴¹ Literatürde “Rectifier Linear Unit” olarak bilinmekte olup “ReLU” biçiminde kısaltılmaktadır.

⁴² Terminolojide “Leaky Rectifier Linear Unit” olarak bilinmekte olup “LReLU” biçiminde kısaltılmaktadır.

- Üstel Doğrusal Birim⁴³ (Clevert ve ark., 2015),
- Ölçekli Üstel Doğrusal Birim⁴⁴ (Klambauer ve ark. 2017),
- SoftPlus,
- Hiperbolik tanjant.

Bu fonksiyonlar aşağıdaki matematiksel ifadelerle temsil edilmektedir:

$$(ReLU, LReLU) \Rightarrow F = \begin{cases} ax & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \quad (1.13)$$

$$(ELU, SELU) \Rightarrow F = \lambda \begin{cases} \beta[e^{-x} - 1] & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \quad (1.14)$$

$$(SoftPlus) \Rightarrow F = \log[1 + e^x] \quad (1.15)$$

$$(TanH) \Rightarrow F = \tanh x = \frac{2}{1+e^{-2x}} - 1 \quad (1.16)$$

Burada ReLU için $a = 0$, LReLU için $a = 0.01$, ELU için $\beta = \lambda = 1$ ve son olarak SELU fonksiyonu için $(\beta, \lambda) = (1.6733, 1.0507)$ olması gerektiğini vurgulamak yerinde olacaktır.

- Bırakma olasılığı:

Bu hiper parametrenin genel olarak kabul edilen tanımı "sinir ağı katmanlarında bir düğümü eğitme olasılığı" biçimindedir. Bırakma olasılığı değeri $p_b=1.0$ olursa kullanılan yaklaşımda veri bırakma yok demektir, buna karşın bırakma olasılığı değerinin $p_b=0.0$ olması ise veri katmanından herhangi bir çıkışın olmadığı duruma işaret etmektedir (Srivastava ve ark., 2014).

- Parti boyutu/Dönem sayısı:

Bu parametreler sırasıyla mimari iletim hattı boyunca yayılan örneklerin sayısını ve tüm eğitim sürecinin bir adım ileri (veya geri) geçişini belirtmektedir. Parti boyutu değeri avantajlı sonuçlar elde etmek için genellikle tüm numunelerin sayısından daha küçük alınmaktadır. Öte yandan, dönem sayısı ise eğitim-öğrenme sürecinde modülün ağırlıklarının kaç kez değiştirilmesi gerektiğini belirler ve değeri geleneksel olarak büyük (100'ler ya da 1000'ler mertebesinde) alınmaktadır.

LSTM-Bırakma modelinde kullanılacak hiper parametreler ayarlanırken farklı kombinasyonlarda inşa edilen mimarinin yaptığı tahminlerin gerçek verilerle karşılaştırılmasıyla R -kare değerleri hesaplanmaktadır (en yüksek R -kare değeri en uygun bileşen kombinasyonunu işaret etmektedir). İstatistik alanında çok önemli rolleri olan R -kare aracı temel olarak bir eğilimin uyum iyiliğini ölçmektedir ve şu şekilde tanımlanır:

⁴³ Alanda "Exponential Linear Unit" olarak bilinmektedir ve "ELU" biçiminde kısaltılır.

⁴⁴ Bu modelin terminolojideki karşılığı "Scaled Exponential Linear Unit" biçiminde olup kısa olarak "SELU" şeklinde kullanılmaktadır.

$$R^2 = 1 - \frac{\sum_i (T_i - \hat{T})^2}{\sum_i (T_i - \bar{T})^2} . \quad (1.17)$$

Yukarıda yazılan T_i gözlemsel (deneysel) verileri temsil ederken $\hat{T}$ ise tahmin edilen değeri göstermektedir. R -kare değeri için önemli ve unutulmaması gereken bir kısıtlama bulunmaktadır:

$$0 \leq R^2 \leq 1 . \quad (1.18)$$

Burada $R^2 = 1$ değerinin gerçek ve tahmin edilen veriler arasında herhangi bir sapmanın olmadığı duruma karşılık geldiğine dikkat edilmelidir.

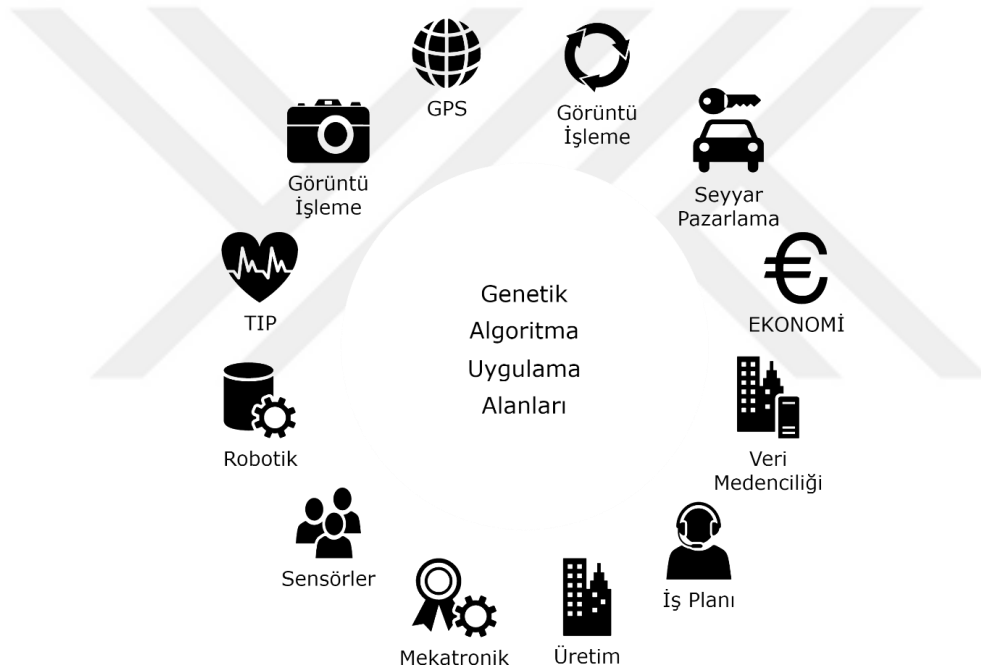
LSTM yaklaşımın önemli uygulamaları olarak aşağıdaki örnekleri vermek mümkündür:

- İşletme yönetimi alanında çeşitli tahmin görevleri (Tax ve ark. 2017),
- El yazısı tanıma (Graves ve ark., 2007; Graves ve Schmidhuber, 2009),
- Robot kontrolü (Mayer ve ark., 2006),
- Ritim öğrenme (Gers ve ark., 2002),
- Konuşma tanıma (Graves, 2005; Fernández ve ark., 2007; Graves, 2013),
- Gramer öğrenimi (Gers ve Schmidhuber, 2001; Schmidhuber ve ark., 2002; Perez-Ortiz ve ark., 2003),
- Astrofizikte CMB sıcaklık dağılımını kuramsal olarak modelleme (Arjona, 2020; Salti ve ark., 2021; Salti ve Kangal, 2022),
- Evrenbilimde kozmik Hubble parametresini kuramsal olarak modelleme (Escamilla-Rivera ve ark., 2020; Tilaver ve ark., 2021; Salti ve ark., 2022),
- Protein homoloji tespiti (Hochreiter ve ark., 2007),
- Müzik kompozisyonu, besteleme (Eck ve Schmidhuber, 2002),
- İnsan eylemi tanıma (Baccouche ve ark., 2011),
- Zaman serisi tahmini (Wierstra ve ark., 2005),
- Tıbbi bakım yollarında tahmin (Choi ve ark., 2016),
- İlaç tasarımı (Gupta ve ark., 2018),
- Proteinlerin hücre altı lokalizasyonunu tahmin etme (Thireou ve Reczko, 2007),
- Anlamsal ayrıştırma (Jia ve Liang, 2016),
- Havaalanı yolcu yönetimi (Orsini ve ark., 2019),
- Kısa vadeli trafik tahmini (Zhao ve ark. 2017),
- İşaret dili tanıma (Huang ve ark., 2018),
- Piyasa Tahmini (Saiful ve Hossain, 2020),
- Nesne eş-bölümlendirme (Wang ve ark., 2018; Duan ve ark., 2018).

Derin yapıli sinir ağı modelleri arasında önemli yeri olan LSTM algoritmasının kullanıldığı alanlarda elde edilen dikkate değer düzeydeki tutarlı sonuçlar ML mimarilerinin önemini açıkça ortaya koymaktadır. Yukarıdaki örnekler LSTM modülünün uygulamalarına yönelik bahsedilebilecek çalışmalardan sadece bazılarıdır ve bunların sayısını artırmak oldukça kolaydır. Bu durum LSTM algoritmasının en çok kullanılan (ya da tercih edilen) ML mimarileri arasına girmesine neden olmaktadır.

1.7.3 Genetik Algoritma

Bu yaklaşım da tıpkı LSTM mimarisi gibi oldukça farklı alanlarda tutarlı analiz sonuçları verebilmektedir. Genel bir çerçevede genetik algoritmanın kullanım alanları Şekil 1.12’de özetlenmektedir.



Şekil 1.12. Genetik algoritma uygulama alanlarına örnekler.

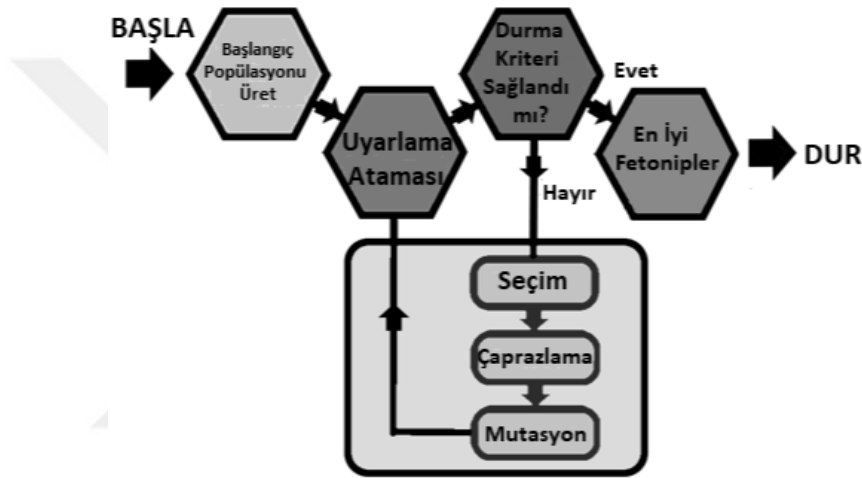
Herhangi bir alanda inşa edilen kuramsal bir modelin ürettiği sonuçlarının gerçek gözlem verileriyle en iyi biçimde uyuşması için modeldeki keyfi parametrelerin değerlerinin dikkatlice belirlenmesi gerekmektedir. Yukarıda bahsi geçen alanlara ek olarak genetik ML yaklaşımı kuramsal bir modeli düzenlemek/ayarlamak/fit etmek için de kullanılabilir (Mitchell, 1996; Haupt ve Haupt, 1998).

Genetik algoritma modeli temel olarak yinelemeli bir prosedürdür ve beş ana bölümden oluşmaktadır:

- Başlangıç popülasyonunun belirlenmesi,
- Optimize edilecek bir uygunluk fonksiyonunun seçilmesi,

- Hangi genotiplerin (veya kromozomların) çoğaltılması gerektiğini belirleyen bir seçim operatörünün belirlenmesi,
- Yeni nesil genotipler üretmek için bir çaprazlama operatörünün seçilmesi,
- Yeni bir nesil için genotipleri rastgele mutasyona uğratan bir mutasyon operatörünün tanımlanması.

Doğal seçim temelli bu algoritma bir aday çözüm popülasyonu (buna genellikle fenotip denir) ile başlamaktadır. Aday çözüm genellikle bir genotip oluşturmak için süreç içerisinde bir dizi oluşturan ve genler olarak adlandırılan parametreler ile temsil edilmektedir. Her bir fenotip için bir uygunluk puanı üretildikten sonra uygunluk fonksiyonu ilgili fenotipin ne kadar uygun olduğunu belirlemek ve bazı fenotipleri ebeveyn olarak seçmek için etkili bir yöntem sağlamaktadır.



Şekil 1.13. Genetik aloritmada genel akış şeması (Salti ve ark. 2022).

Sıradan bir genetik öğrenme algoritmasının genel işleyiş prosedürü Şekil 1.13'te temsili olarak gösterilmektedir. Sürecin “Seçim⁴⁵” aşaması yalnızca en uygun fenotipleri belirlemekle kalmaz, aynı zamanda genlerin bir sonraki nesle aktarmalarına olanak sunmaktadır. Bir sonraki aşamada “Çaprazlama⁴⁶” ve “Mutasyon” operatörlerinin olası bir kombinasyonu yardımıyla seçilen veri elemanları arasından yeni bir fenotip popülasyonu üretilir. Nihai nesil evresi için bir sonlandırma durumu elde edilene kadar süreç tekrarlanmaktadır.

Bir önceki alt başlıkta bahsedilen LSTM-Bırakma yaklaşımında olduğu gibi genetik algoritma da sunduğu tahmin sonuçlarına doğrudan etki edebilecek kritik parametreler içermektedir: uyarlama⁴⁷ fonksiyonu, boyut⁴⁸, popülasyon büyüklüğü, mutasyon oranı ve çaprazlama değeri. Boyut sayısına örnek vermek gerekirse ayarlanması gereken bir kuramsal model söz konusuysa ilgili modeldeki keyfi

⁴⁵ Selection.

⁴⁶ Crossover.

⁴⁷ Fitness.

⁴⁸ Dimension.

parametre sayısı algoritmanın boyut sayısı olarak alınır. Popülasyon büyüklüğüne veri setinin uzunluğuna bakılarak karar verilirken, mutasyon oranı ve çaprazlama değeri genellikle sırasıyla 0,05 ve 0,95 olarak seçilmektedir. Son olarak, uyarlama fonksiyonu olarak da ki-kare⁴⁹ fonksiyonu kullanılabilir.

$$\chi_H^2(\theta) = \sum_{i=1}^n \frac{[T_K(z_i|\theta) - T_D(z_i)]^2}{\sigma_i^2} . \quad (1.19)$$

Burada yazılan T_K , T_D , θ ve σ_i terimleri sırasıyla kuramsal çıktıyı, deneysel çıktıyı, keyfi parametreleri ve sapma miktarlarını betimlemektedir.

Genetik yaklaşım tıpkı derin yapılı LSTM algoritması gibi tahmin sonuçları ile gerçek veriler arasındaki sapma miktarlarını hesaplayamamaktadır. Bu sorunu çözebilmenin çeşitli yolları bulunmaktadır:

- Önyükleme⁵⁰ yaklaşımı (Press ve ark., 2007),
- Om istatistiği (Nesseris ve Shafieloo, 2010),
- Yol⁵¹ integrali temelli yöntem (Nesseris ve Garcia-Bellido, 2012),
- Fisher Bilgi Matrisi⁵² yaklaşımı (Arjona, 2020).

1.7.4. FIM

Bu yaklaşım hem veri tahminleri yürütmek hem de istatistiksel bir analizdeki belirsizlikleri hesaplamak için kullanılabilir. Temel olarak, bir parametre seti olan θ 'nın dağılımı ele alınan model olan M ve seçilmiş deneysel veri kümesi olan D 'nin aracılığıyla tartışılarak nihai⁵³ olasılık öngörüsü yapılmaktadır (Wolz ve ark., 2012):

$$\Pi(\theta) = \xi(\theta|D; M) . \quad (1.20)$$

Bu miktar Bayes teoreminden yola çıkılarak muhtemellik⁵⁴ dağılımı $L(\theta) = \xi(\theta|D; M)$, önceki olasılık $\psi(\theta) = \xi(\theta|M)$ ve Bayes kanıtı $Y(\theta) = \xi(D|M)$ 'nin bir fonksiyonu cinsinden yazılabilmektedir (Wolz ve ark., 2012):

$$\Pi(\theta) = \frac{1}{Y} L(\theta) \psi(\theta) . \quad (1.21)$$

⁴⁹ Chi-squared.

⁵⁰ Bootstrap.

⁵¹ Path.

⁵² Literatürde "Fisher Information Matrix" olarak bilinir ve "FIM" biçiminde kısaltılmaktadır.

⁵³ "Posterior" ya da "Latter".

⁵⁴ Likelihood.

Sabit bir önerme için bir parametrenin tahmini değerini araştırdığımızda Bayes kanıtı önemsiz bir normalleştirme faktörüne dönüşmektedir. Bu nedenle, sıralı olan veri setlerinin analizinde bu nicelik göz ardı edilebilmektedir. Böylece, olasılık dağılımı $\Pi(\theta)$ için Fisher bilgi matrisi şu şekilde tanımlanmaktadır (Kendall ve Stuart, 1969; Tegmark ve ark., 1997):

$$F_{\mu\nu} = \left\langle -\frac{\partial^2 \ln \Pi(\theta)}{\partial \theta_\mu \partial \theta_\nu} \right\rangle. \quad (1.22)$$

Buradaki θ_μ niceliğinin modeldeki serbest parametre setine (örneğin 4 tane keyfi parametre içeren bir model için $\theta_\mu = (\theta_1, \theta_2, \theta_3, \theta_4) = (X, Y, A, B)$ biçimindedir) karşılık geldiğini söylemiştik. Muhtemellik ve nihai olasılık kavramları geniş ve tekdüze öncelikler sebebiyle birbirinin yerine de kullanılabilmektedir (Wolz ve ark., 2012). Ters⁵⁵ FIM genellikle maksimum olasılık tahmincisi ile aynı sınıfta olan olasılığın ortalama eğriliğini ifade etmektedir. 1945 ve 1946 yıllarında, Cramér ve Rao birbirinden habersiz olarak nokta tahminlerinin varyansına sade bir alt sınır elde etme problemini incelemiştir (Rao, 1945; Cramér 1946). Böylece, Cramér-Rao eşitsizliği⁵⁶ olarak bilinen ilke sayesinde FIM’de tanımlanan θ_μ parametre kümesindeki keyfi sabitlerin hata payı için bir sınır elde edilebilmektedir (Wolz ve ark., 2012):

$$\Delta\theta_\mu \geq \frac{1}{\sqrt{F_{\mu\mu}}}. \quad (1.23)$$

Denklem (1.22) ve denklem (1.23) tarafından ortaya çıkarılan özellikler, FIM’in olasılığın birinci dereceden iyimser bir yaklaşımı olarak kullanılabileceğini vurgulamaktadır (Wolz ve ark., 2012). FIM yaklaşımını $\theta_\mu = (\theta_1, \theta_2, \theta_3, \theta_4) = (X, Y, A, B)$ parametre setini içeren bir kuramsal modele uygulamak için karşılık gelen bilgi matrisi şu şekilde yazılabilmektedir:

$$F_{\mu\nu} = \begin{pmatrix} \frac{\partial^2 \chi_H^2}{\partial X^2} & \frac{\partial^2 \chi_H^2}{\partial X \partial Y} & \frac{\partial^2 \chi_H^2}{\partial X \partial A} & \frac{\partial^2 \chi_H^2}{\partial X \partial B} \\ \frac{\partial^2 \chi_H^2}{\partial Y \partial X} & \frac{\partial^2 \chi_H^2}{\partial Y^2} & \frac{\partial^2 \chi_H^2}{\partial Y \partial A} & \frac{\partial^2 \chi_H^2}{\partial Y \partial B} \\ \frac{\partial^2 \chi_H^2}{\partial A \partial X} & \frac{\partial^2 \chi_H^2}{\partial A \partial Y} & \frac{\partial^2 \chi_H^2}{\partial A^2} & \frac{\partial^2 \chi_H^2}{\partial A \partial B} \\ \frac{\partial^2 \chi_H^2}{\partial B \partial X} & \frac{\partial^2 \chi_H^2}{\partial B \partial Y} & \frac{\partial^2 \chi_H^2}{\partial B \partial A} & \frac{\partial^2 \chi_H^2}{\partial B^2} \end{pmatrix}. \quad (1.24)$$

1.7.5. DTs

Karar ağacı yaklaşımları bir dizi sade mekanizmayı uygulayarak toplu ve büyük/geniş veri havuzlarını tutarlı bir biçimde gruplara ayırmak için kullanılmaktadır. Bu analiz yöntemi temel olarak

⁵⁵ Inverse.

⁵⁶ “Cramér–Rao bound” ya da “Cramér–Rao inequality”.

sınıflandırma yapabilen veri madenciliği modelidir ve literatürde (Quinlan, 1986; 1987; 1996; Berry ve Linoff, 2004)

- C4.5,
- C5.0,
- Yinelemeli Ayırıcı 3⁵⁷
- Sınıflandırma ve Regresyon Ağacı⁵⁸,
- Ki-kare Otomatik Etkileşim Tespiti⁵⁹,
- Hızlı Tarafsız Verimli İstatistiksel Ağaç⁶⁰

gibi çeşitli DTs temelli ML algoritmaları bulunmaktadır. Mantıksal kuralları harekete geçiren sistemlerin yanı sıra DTs algoritmaları da pratikte oldukça popülerdir. Bunun temel nedeni kısmen sağlamlıkları, veri işleme hızları ve kullanıcıların yorumlayabileceği açık kavram betimlemelerini kolayca üretebilmeleridir. Mimariyi oluşturan paketler⁶¹ algoritma düğümleri tarafından keşfedilen belirli bir çıktı alanını tahmin etmek için ağaç yapılarını temsil etmektedirler. Ağaç modelleri doğrudan ağaç oluşturma düğümünden ya da dolaylı olarak etkileşimli ağaç oluşturunucudan kurgulanabilmekte ve algoritmanın model paketleri belirtilen amaca bağlı olarak farklı seçenekler içerebilmektedir (Quinlan, 1986; 1987; 1996; Berry ve Linoff, 2004): *Tek ağaç modeli paketleri ve Artırma, paketleme ve çok büyük veri kümeleri için model paketleri.*

Ağaç oluşturma tekniğinde karışık formda öznitelikler içeren bir veri havuzunun daha küçük ve olabildiğince daha düzenli alt gruplara ayrılması amaçlanmakta ve bu iş için de ilgili veri seti elemanlarının değerleri/durumları kullanılmaktadır (Shahnaz, 2006). Eğitim verilerinden bir ağaç oluşturmak istendiğinde nazari ölçüm kullanan DTs algoritmaları kılavuz olarak dinamik bir yaklaşım kullanılmaktadırlar. Bu yöntem için ortaya konan çeşitli algoritmaların olması oluşturulabilecek dallar için farklı (hangi yöntemlerin nasıl kullanılacağı konusunda/temelinde) önerileri ortaya çıkarmaktadır. Ancak özünde ağacın kökü için bir öznitelik seçilir ve eğitim verileri ilgili özniteliğin değerlerine karşılık gelen alt kümelere bölünür.

Tablo 1.2’de sunulan ve golf oynamaya yönelik karar verilerini içeren bilgi grubu göz önüne alındığında Şekil 1.14’teki gibi bir karar ağacı oluşturmak mümkün olmaktadır (Hall, 1999). Burada, ağaçtaki düğümler özelliklere karşılık gelirken dallar ise ilişkili değerleri/durumları temsil etmektedir.

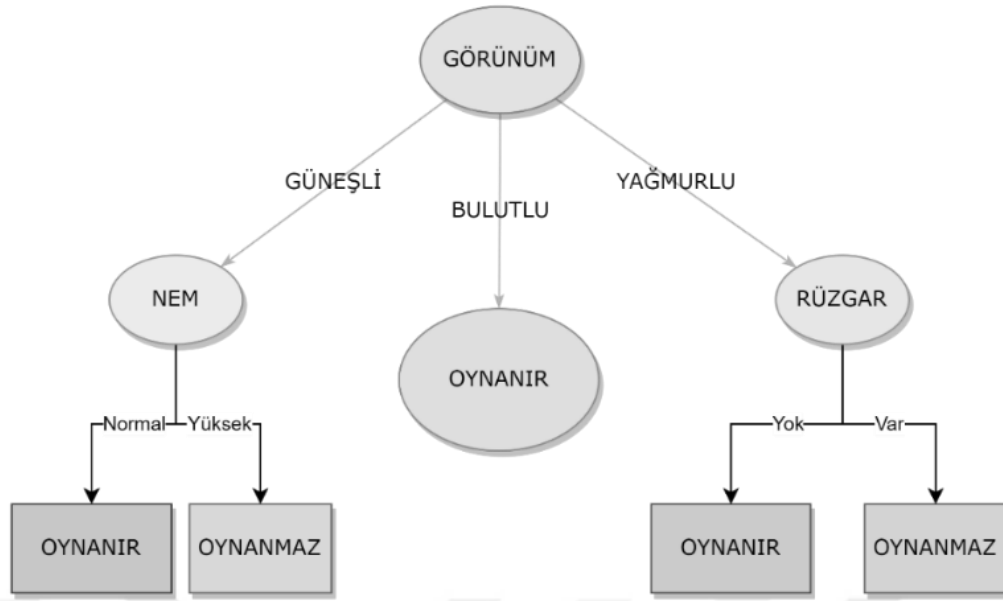
⁵⁷ Alan terminolojisindeki karşılığı “Iterative Dichotomiser 3” şeklinde olup “ID3” olarak kısaltılmaktadır.

⁵⁸ Literatürde “Classification & Regression Tree” ya da “CHART” olarak bilinmektedir.

⁵⁹ Terminolojide “Chi-square Automatic Interaction Detection” ya da “CHAID” olarak bilinmektedir.

⁶⁰ Literatürde “Quick Unbiased Efficient Statistical Tree” ya da “QUEST” şeklinde anılmaktadır.

⁶¹ Nuggets.



Şekil 1.14. Golf veri kümesi için bir karar ağacı.

Görüldüğü üzere, ağacın yaprakları farklı sınıflara karşılık gelmektedir. Yeni bir örneği sınıflandırmak için ağacın düğümlerinde test edilen özellikler incelenir ve örnekte gözlemlenen değerlerine karşılık gelen dallar izlenir. Bir yaprağa ulaşıldığında süreç sonlandırılır ve yapraktaki sınıf örneğe atanır. Örnek bir günü (mesela güneşli bir gökyüzü, sıcak bir hava, normal nem durumu ve rüzgârsız bir ortam olsun) sınıflandırmak için Şekil 1.14’te verilen karar ağacını kullanmak istediğimizde başlangıçta ağacın kökündeki özelliğin (hava görünümünün) incelenmesi gerekmektedir. Yeni örnekte “görünüm” değeri “güneşli” olduğundan sol dal izlenmelidir. Daha sonra, havadaki nem durumu değerlendirilmelidir: bu durumda yeni örnek için “normal nem” durumu söz konusu olduğundan sağdaki dal takip edilmelidir. Bu işlem bizi bir yaprak düğüme getirmiştir, dolayısıyla ilgili örneklem “Oynanır” sınıfına atanacaktır.

1.7.6. Örnek-tabanlı Öğrenim⁶²

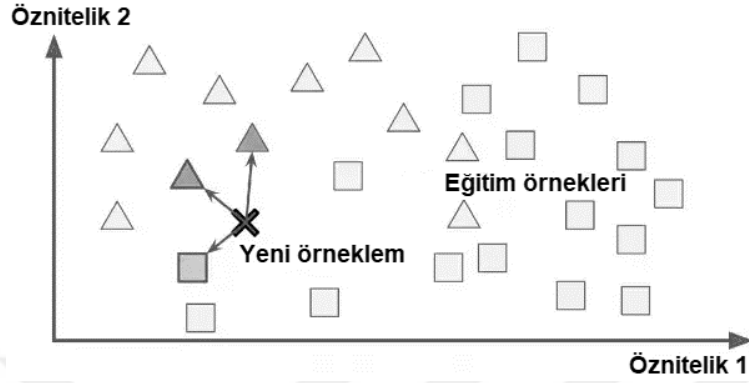
Bu tür algoritmalar veri kümesinden edinilen bilgiyi belirli durumlar veya deneyimler biçiminde ele almaktadırlar. Saklanan bilgiyi almak için verimli eşleştirme yöntemlerine güvendiklerinden yeni durumlara da uygulanabilmektedirler (Hall, 1999). Örneğe dayalı ML algoritmaları çoğunlukla hesaplama açısından sadedirler ve çeşitli varyasyonları genellikle insan öğreniminin farklı modelleri olarak kabul edilmektedir (Cunningham, 1997). Diğer yandan, örnek bazlı öğrenme yaklaşımları bazen “tembel” ML algoritmaları olarak anılmaktadırlar, çünkü öğrenme işi sınıflandırma anına kadar ertelenmekte ve yetenek kapasitenin çoğu eşleştirme evresinde harcanmaktadır.

IBL yaklaşımı aslında *en yakın komşu*⁶³ olarak bilinen en basit benzerliğe dayalı ML algoritmasının sade bir uygulamasıdır. Burada, sınıflandırılacak olan örneğe en yakın örneklem (bir

⁶² Literatürde “Instance-based Learning” ya da “IBL” şeklinde anılmaktadır.

⁶³ Nearest Neighbor.

Öklid mesafe ölçüsüne göre) aranır ve yeni örnek kullanılan verinin sınıfına atanır (bakınız Şekil 1.15). En benzer örneklerin (şekilde görüldüğü gibi 2 Üçgen ve 1 Kare) çoğunluğuna (şekildeki örnekte çoğunluğun üçgenler olduğu görülmektedir) baktığımızda yeni örnek bir üçgen olarak sınıflandırılacaktır.



Şekil 1.15. IBL yaklaşımında yakın komşuluk testinin bir temsili (Agrawal, 2020).

IBL algoritmasının kullandığı mesafe ölçüm metriği kapalı formda şöyle ifade edilmektedir

$$d(x, y) = \sqrt{\sum_{i=1}^n f(x_i, y_i)} . \quad (1.25)$$

Bu denklem yardımıyla x ve y örnekleri arasındaki mesafe hesaplanır (yukarıdaki denklemde yazılan x_i ve y_i sırasıyla x ve y örneklerinin i . değerlerini temsil etmektedir). Sayısal değerli öznitelikler için ilgili fonksiyon

$$f(x_i, y_i) = (x_i - y_i)^2 . \quad (1.26)$$

biçimindedir. Diğer yandan, sembolik (kategorik) değerli veriler içinse

$$f(x_i, y_i) = \begin{cases} 0 & x_i \text{ ve } y_i \text{ aynı ise} \\ 1 & x_i \text{ ve } y_i \text{ farklı ise} \end{cases} \quad (1.27)$$

tanımı kullanılmaktadır.

Tablo 1.2'deki golf oyun verilerinin ve denklem 1.25'in yardımıyla hesaplanan/belirlenen mesafe değerleri örnek bir gün (mesela güneşli bir gökyüzü, sıcak bir hava, normal nem durumu ve rüzgârsız bir ortam) için Tablo 1.7'de sunulmaktadır. Bu durumda, örnek güne eşit derecede yakın olan iki adet (#1 ve #9 günleri) örneklem olduğu görülmektedir, dolayısıyla burada keyfi bir seçim yapılabilir.

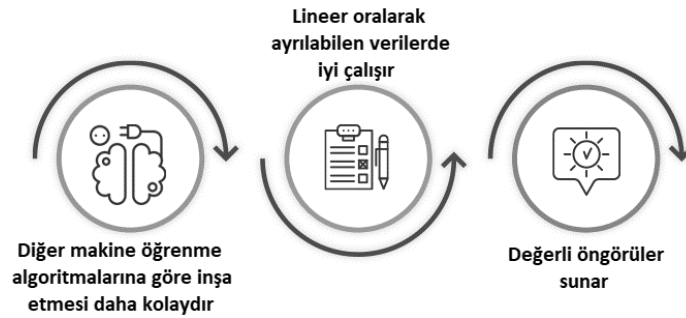
Tablo 1.7. Golf verileri için hesaplanan mesafe değerleri.

Örneklem	Mesafe	En Yakın Komşuluk
#1	1	☒
#2	2	☐
#3	2	☐
#4	3	☐
#5	2	☐
#6	3	☐
#7	2	☐
#8	2	☐
#9	1	☒
#10	2	☐

Basit *en yakın komşu* algoritmasının eğitim verilerindeki ilgisiz özelliklerin varlığından olumsuz etkilendiği bilinmektedir. En yakın komşuluk yaklaşımı ilgisiz bilgilerin varlığında da öğrenebilir ancak bu durumda daha fazla eğitim verisine ihtiyaç duyulmaktadır. Literatürde farklı çalışmalarda belirli bir tutarlılık düzeyine ulaşılması ya da bunun sürdürülmesi istendiğinde bir veri havuzundaki ilgisiz niteliklerin sayısının artmasıyla gereken eğitim verisi miktarının da katlanarak arttığı gösterilmiştir (Aha, 1991; Langley ve Sage, 1994a; 1994b). Dolayısıyla, eğitim verilerinin sınırlı olduğu durumlarda ilgisiz özniteliklerin veri havuzundan kaldırılması en yakın komşunun tahmin edilmesine yönelik performansı iyileştirmektedir. Ayrıca, sınıflandırılacak her örneklemin incelenen eğitim setindeki durumlarının her birisiyle ayrı ayrı karşılaştırılması gerektiğinden en yakın komşuluk tabanlı algoritmalar görece yavaş çalışmaktadırlar. Yani, sonuca ulaşmaları daha uzun zaman gerektirmektedir. Öznitelik seçimi eğitim durumlarının sayısını azaltabilir, çünkü daha az öznitelik (özellikle veri seti elemanları kategorik olduğunda) daha az farklı örneklem durumu anlamına gelmektedir. Gereken eğitim örneklemelerinin sayısını azaltma işlemi kabul edilebilir bir hata oranı sunmasının yanı sıra kullanılan öğrenme algoritmasının hızını önemli ölçüde arttırabilmektedir (Hall, 1999).

1.7.7. Lojistik Regresyon⁶⁴

Bu algoritmanın en göze çarpan üç özelliği yukarıda Şekil 1.16’da vurgulanmaktadır.

**Şekil 1.16.** LR yaklaşımın önemli avantajları (Kanade, 2022).

⁶⁴ Alan çalışmalarında “Logistic Regression” ya da kısaca “LR” olarak bilinmektedir.

Bazen incelenen veri setinde bulunan bilgiler tamamen kategorik/nominal/ordinal olur (yani bilgi havuzunda hiç sayısal ya da nümerik veri yoktur), böyle durumlarda parametrik olmayan yaklaşımlar kullanılmaktadır (Yakut ve Gemici, 2017). LR yaklaşımı sosyal bilimler ve tıp alanı dahil çeşitli alanlarda başarılı analizler yapılmasına olanak sunmaktadır. Örneğin, yaralı hastalardaki ölüm/yaşam durumunu tahmin edebilmek amacıyla sıklıkla kullanılan *Travma ve Yaralanma Şiddet Skoru*⁶⁵ bu teknik kullanılarak tanımlanmıştır (Boyd ve ark., 1987). LR yaklaşımı ayrıca belirli bir sürecin, sistemin veya ürünün başarısızlık olasılığını tahmin etmek amacıyla mühendislik problemlerinde (Strano ve Colosimo, 2006; Palei ve Das, 2009) ve bir müşterinin bir ürünü satın alma veya bir aboneliği durdurma eğiliminin tahmini gibi pazarlama uygulamalarında kullanılabilmektedir (Berry ve Linoff, 1997).

Basit bir örnek olması amacıyla bir açıklayıcı değişken ve iki kategori içeren sırdan bir LR problemi ele alalım. Bir sınıfta 15 öğrenci bulunmaktadır ve bu öğrenci grubu bir sınava çalışmak için 0 ile 5 saat arasında zaman dilimleri harcamıştır. Sorumuz şöyle olsun:

- *Sınava hazırlanmak için harcanan süre bir öğrencinin sınavda başarılı olma olasılığını nasıl etkileyecektir?*

Bu probleme yanıt bulabilmek amacıyla LR yönteminin tercih edilebilmesinin temel nedeni bağımlı değişken (dersi geçme/kalma durumu) için başarılı ve başarısız değerlerinin "1" ve "0" gibi kardinal yani sayısal değerlerle temsil edilebilmeleridir. Diğer yandan, bu ders için “geçti” ve “kaldı” durumları sınavdan alınan nota göre (0-100 aralığındaki kardinal veriler) düzenlenebilmektedir. Böylece, problem basit bir regresyon analizinde ele alınabilir hale getirilmiştir. Hangi öğrencinin sınava hazırlanırken ne kadar saat sayısı harcadığı ve her bir öğrencinin dersten geçme ve kalma durumu Tablo 1.8’de sunulmuştur.

Tablo 1.8. Sınava giren öğrencilerin hazırlık ve sonuç bilgileri.

Öğrenci No	Harcanan Saat Sayısı (x_i)	Başarı Durumu (y_i)
#1	0.50	0
#2	0.75	0
#3	0.5	0
#4	1.25	1
#5	1	0
#6	1.50	1
#7	1.75	1
#8	2	1
#9	1.50	0
#10	2.5	1
#11	3	1
#12	3.25	0
#13	5	1
#14	4.5	0
#15	4.23	1

⁶⁵ Literatürdeki karşılığı “Trauma and Injury Severity Score” ya da daha kısa halede “TRISS” biçimindedir.

Burada, çalışılan saat sayısından (x_i) ve sınav sonuçlarından ($y_i = 1$ geçti, $y_i = 0$ kaldı) oluşan verilere yönelik bir lojistik fonksiyon tanımlanması gerekmektedir. Görüldüğü üzere, tablodaki veri noktaları $i=1$ 'den $i=15$ 'e kadar değerler alan i alt simgesi ile indekslenilmektedir. Ayrıca, x değişkeni "açıklayıcı parametre" olarak ele alınırken y değişkeni ise sırasıyla "1" ve "0" kardinal değerleriyle betimlenen "geçti" ve "kaldı" kategorilerini betimleyen "kategorik değişken" olarak kullanılır. Bu örnekten anlaşıldığı üzere LR istatistiğinde bağımlı değişkenin kategorik olarak ikili⁶⁶ ya da çoklu kategorili olması durumunda bağımsız değişkenlerle sebep-sonuç ilişkisi tartışılmaktadır (Özdamar, 2004; Yakut ve Gemici, 2017). Bu noktada, LR yaklaşımında bağımsız parametrelerin normal dağılıma uygun olması ve bütün kategoriler için kovaryans matrislerinin aynı olması varsayımının aranmadığını vurgulamak yerinde olacaktır (Kalaycı, 2008; Yakut ve Gemici, 2017). Sonuç olarak, LR yönteminin temel amacı bağımlı parametrenin değerinin öngörülmesi yerine onun "1" değerini alma olasılığının hesaplanmasıdır (Alpar, 2013). Dolayısıyla, elde edilecek nihai veri esasen bir olasılık değeridir (yani bu değer $[0,1]$ aralığında bir gerçel sayı olmalıdır).

Lojistik fonksiyon genel olarak şöyle betimlenmektedir (Hosmer ve Lemeshow, 2000):

$$\pi(x) = \frac{1}{1 + e^{-\frac{\mu - x}{s}}} \quad (1.28)$$

Buradaki μ bir konum parametresi⁶⁷ iken s ise bir ölçek parametresidir. Yukarıdaki ilişki farklı olarak şu biçimde de yazılabilmektedir (Hosmer ve Lemeshow, 2000):

$$\pi(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \quad (1.29)$$

öyle ki burada

$$\beta_0 = -\frac{\mu}{s} \quad (1.30)$$

olup kesişme noktası olarak bilinmektedir⁶⁸. Öte yandan,

$$\beta_1 = \frac{1}{s} \quad (1.31)$$

⁶⁶ Binary veya dichotomous.

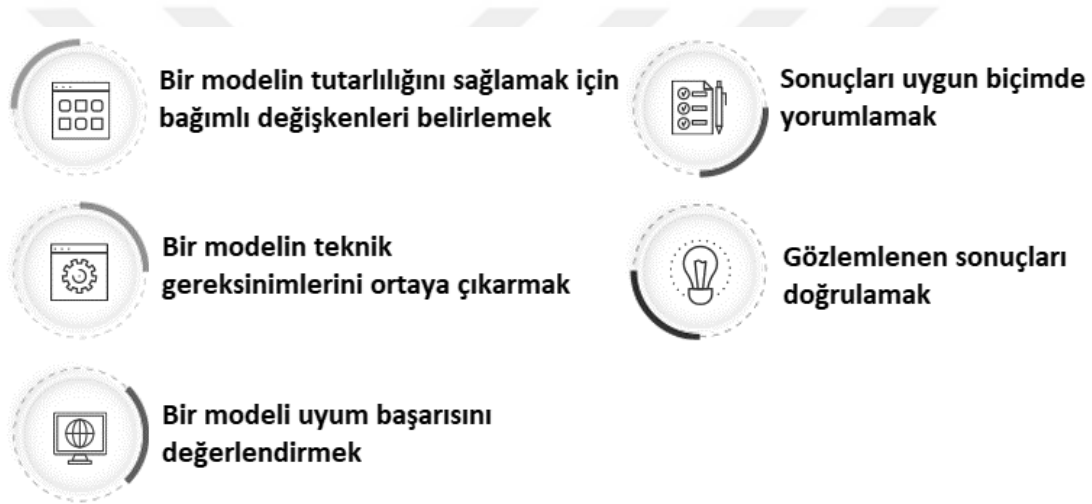
⁶⁷ İlgili olasılık eğrisinin orta noktası $\pi(\mu) = \frac{1}{2}$ 'dir.

⁶⁸ Bu nokta aslında dikey kesişme noktası ya da $y = \beta_0 + \beta_1 x$ eğrisinin y -eksenini kestiği noktadır.

ise ters ölçek parametresi⁶⁹ olarak adlandırılmaktadır ve aslında x 'e bağlı olarak değişen *logaritmik-tuhaflik*⁷⁰ fonksiyonun eğimidir. *Tuhaflik*⁷¹ değeri ise kısaca bir durumun gerçekleşme ihtimalinin gerçekleşmeme ihtimaline oranı şeklinde tanımlanmaktadır (Yakut ve Gemici, 2017). Diğer bir ifade şekliyle, $x=1$ değeri için “tuhaflik” değeri şöyle yazılır (Hosmer ve Lemeshow, 2000; Menard, 2002):

$$Odds(x = 1) = \frac{\pi(1)}{1-\pi(1)}. \quad (1.32)$$

Kısaca LR olarak adlandırılan yapay öğrenme yaklaşımının en başarılı olduğu (ya da en tutarlı öngörülerini sunduğu) uygulama alanlarından öne çıkanlar Şekil 1.17’de vurgulanmıştır. Buradan, LR yönteminin oldukça farklı durumlarda kullanılabilen faydalı bir öğrenme yaklaşımı olduğu görülmektedir.



Şekil 1.17. LR yaklaşımına uygun alanlar (Kanade, 2022).

1.7.8. Destek Vektör Makineleri⁷²

Başarılı sınıflandırma ve regresyon analizleri yapabilen bir yaklaşım olup AT&T Bell laboratuvarlarında geliştirilmiştir (Vapnik ve Chervonenkis, 1974; Boser ve ark., 1992; Cortes ve Vapnik, 1995). En yaygın kullanım alanları Şekil 1.18’de gösterilmektedir.

⁶⁹ Veya “oran parametresi”.

⁷⁰ Dilimize böyle çevirdiğimiz kavramın literatürdeki bilinen adı “Log-odds” şeklindedir.

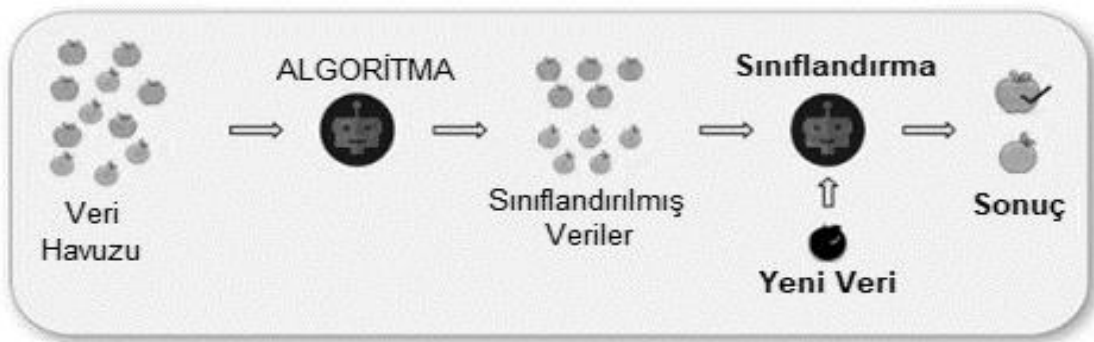
⁷¹ Bu niceliğin terminolojik karşılık “Odds” biçimindedir.

⁷² Literatürdeki karşılığı “Support Vector Machines” ya da “Maximum Margin Classifier” biçiminde olup kısa formda genellikle “SVMs” olarak betimlenmektedir.



Şekil 1.18. Destek Vektör Makinelerinin yaygın kullanım alanları.

Terminolojik bir tanımla ifade etmek gerekirse sıradan bir SVMs modeli sınıflandırma, regresyon ve sıra-dışı verilerin tespiti görevleri için kullanılabilen yüksek (ya da sonsuz) boyutlu bir alanda bir hiper-düzlem (veya hiper-düzlemler) kümesi oluşturan yöntemdir (Vapnik ve Chervonenkis, 1974; Boser ve ark., 1992; Cortes ve Vapnik, 1995). Sezgisel bağlamda herhangi bir kategorinin en yakın eğitim-veri noktasına (fonksiyonel sınır) en uzak mesafeye sahip olan hiper-düzlem başarılı bir ayırım sunacaktır, zira marj ne kadar büyük olursa sınıflandırıcının genelleme hatası aynı oranda düşük kalacaktır. İstatistiksel öğrenme çerçevelerine dayanan SVMs yöntemi en sağlam tahmin algoritmaları arasında gösterilmektedir (Vapnik ve Chervonenkis, 1974; Vapnik, 1995). SVMs yaklaşımında herhangi bir veri seti n -boyutlu bir vektör (sayılardan oluşan bir liste) olarak ele alınır ve veri noktalarının $(n-1)$ -boyutlu bir hiper-düzlemle⁷³ ayrılıp ayrılmayacağı sorgulanır (Vapnik ve Chervonenkis, 1974; Boser ve ark., 1992). Lineer (ya da çizgisel) biçimde ayrılabilen veri havuzlarında çok başarılı sonuçlar üretebildiğinden genellikle bir *çizgisel sınıflandırıcı* tipi olarak kabul edilmektedir (Cortes ve Vapnik, 1995). Şimdi, elimizdeki verilerin bileşenlerinin iki kategoriye ayrıldığını ve amacımızın da bize verilen yeni bir veri noktasının hangi kategoriye gireceğine karar vermek olduğunu farz edelim. Bu durumu Şekil 1.19'daki gibi sade bir görselleştirme yöntemiyle özetlemek mümkündür.



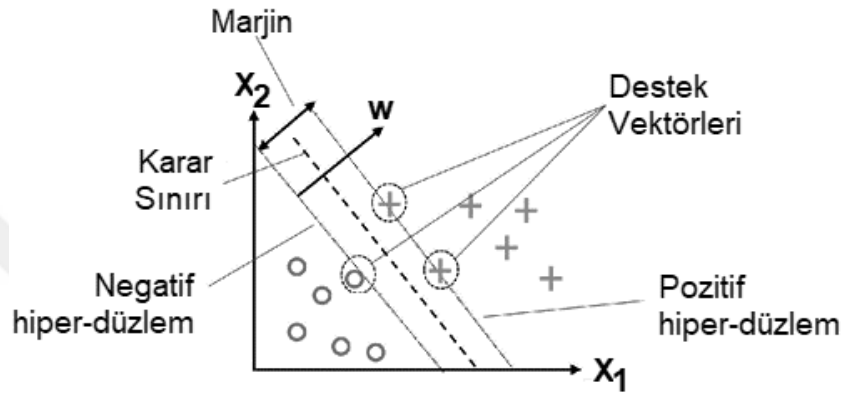
Şekil 1.19. Sınıflandırma analizinin temsili akışı (Simplilearn, 2020).

SVMs yönteminde en doğru hiper-düzlem elimizdeki iki sınıf arasında en büyük ayrımı⁷⁴ yapandır (Boser ve ark., 1992; Cortes ve Vapnik, 1995), dolayısıyla bir hiper-düzlem en yakın veri noktasına olan

⁷³ Bu yaklaşımda verileri sınıflandırabilecek birçok hiper-düzlem olabilmektedir.

⁷⁴ Düzlemler arasındaki boşluk (ya da marj, marjin) ne kadar fazla olursa sınıflandırılan veriler arasındaki ayırım o kadar net ortaya çıkacaktır.

mesafeyi maksimize edecek şekilde seçilmektedir (Vapnik, 1995). Eğer analiz sonucunda bu özelliği sağlayan bir hiper-düzleme rastlanırsa buna maksimum-marjlı hiper-düzlem denir, böylece tanımlanmış olduğu sınıflandırıcı yaklaşımı ise “maksimum-marjlı sınıflandırıcı”⁷⁵ olarak kabul edilecektir (Boser ve ark., 1992; Cortes ve Vapnik, 1995; Vapnik, 1995; Hamel, 2009). Orijinal problem sonlu-boyutlu bir uzayda ifade edilmiştir, lakin ayırım yapılacak veri kümelerinin böyle sonlu-boyutlu uzaylarda çizgisel olarak ayıramaz olduğu sıklıkla görülmektedir. Buradan yola çıkarak çizgisel veri ayırımını kolaylaştırmak amacıyla sonlu-boyutlu uzayın daha yüksek boyutlu bir uzaya haritalanması önerilmiştir (Vapnik, 1995; Hamel, 2009).



Şekil 1.20. SVMs yaklaşımının temsili bir şeması (Kumar, 2020).

Bu yaklaşımı daha kolay anlayabilmek adına Şekil 1.20’de temsili bir grafik yardımıyla ilgili analizin temelleri ifade edilmektedir. Ortadaki kesikli çizgi karar sınırı olup veriler arasındaki ayrımı ortaya koyan optimal karar düzlemidir, W ise hiper-düzlemin normal (yani düzleme dik) vektörüdür ve normalize edilmesi gerekli değildir. İkili sınıflandırıcı analizlerinde veri gruplarını birbirinden ayırmak için buradaki gibi iki adet destekli hiper-düzleme gereksinim duyulmaktadır⁷⁶ (Yakut ve Gemici, 2017): pozitif hiper-düzlem ve negatif hiper-düzlem.

SVMs temelli analizde kullanılmak üzere

$$(x_1, y_1), \dots, (x_n, y_n) \text{ öyle ki } x \in R, y \in \{-1, +1\} \quad (1.33)$$

biçiminde veri çiftlerinden oluşan eğitim seti verildiğinde farklı sınıflara ait veriler bir adet hiper-düzlem aracılığıyla ayrıştırılabilmektedir (Vapnik, 1995). Bu yaklaşımda, herhangi bir hiper-düzlem aşağıdaki denklemi sağlayan x_i noktalarının bir kümesi olarak tanımlanmaktadır:

$$W^T x - b = 0. \quad (1.34)$$

⁷⁵ Ya da “optimal kararlılık algılayıcısı”.

⁷⁶ Bu iki düzlem arasındaki boşluk doğal olarak ayırım marjı olacaktır.

Burada yazılan b terimi sabit bir sayı olup $b/\|w\|$ parametresi hiper-düzlemin normal vektörü boyunca orijinden kaymasını belirtmektedir (Vapnik, 1995; Hamel, 2009). Diğer yandan, çalışılan eğitim verileri *çizgisel olarak birbirinden ayrılabilir*⁷⁷ şekil 1.20’de gösterildiği gibi iki veri sınıfını ayıran iki paralel hiper-düzlem de seçilebilmektedir.

Normalize (veya standardize) edilmiş bir veri kümesi için aşağıdaki koşullar sağlanmaktadır:

- $W^T X - b = +1$:

Düzlemde veya üzerinde bir yerdeki herhangi bir nokta⁷⁸ “+1” etiketi ile sınıflandırılır.

- $W^T X - b = -1$:

Düzlemde veya altında bir yerdeki herhangi bir nokta⁷⁹ “-1” etiketi ile sınıflandırılır.

Yukarıda tanımlanan iki farklı hiper-düzlem arasındaki mesafe geometrik olarak $2/\|W\|$ olduğundan⁸⁰ düzlemler arasındaki mesafeyi maksimize etmek için $\|W\|$ ’yı minimize etmemiz gerekmektedir (Yakut ve Gemici, 2017). Ayrıca, veri noktalarının marj içine düşmesini de önlememiz gerektiğinden şu kısıtlamaları kabul etmemiz gerekmektedir (Campbell ve Ying, 2011):

$$W^T x_i - b \geq 1 \quad (\text{eğer } y_i = 1 \text{ ise}), \quad (1.35)$$

$$W^T x_i - b \leq -1 \quad (\text{eğer } y_i = -1 \text{ ise}). \quad (1.36)$$

İlgili kısıtlamalar her bir veri noktasının marjın doğru tarafında olması gerektiğini belirtmektedir. Bu kısıtlamalar şu biçimde de ifade edilebilmektedir (Campbell ve Ying, 2011)

$$y_i(W^T x_i - b) \geq 1 \quad (\text{tüm } 1 < i < n \text{ değerleri için}). \quad (1.37)$$

Sonuç olarak, optimizasyon problemini betimlemek için aşağıdaki denklem yazılabilmektedir (Vapnik, 1995; Burges, 1998; Alpaydın, 2011; Campbell ve Ying, 2011):

$$\min_{W,b} \frac{\|W\|^2}{2} \quad \text{ancak şöyle kısıtla } y_i(W^T x_i - b) \geq 1, \forall i \in \{1, \dots, n\}. \quad (1.38)$$

Yazılan geometrik tanımlamanın önemli bir sonucu maksimum boşluklu bir hiper-düzleminin tamamen ona en yakın olan x_i noktaları tarafından belirlenmesidir. Bu nedenle, x_i ’ler destek vektörleri olarak anılmaktadır. Bu algoritma her ne kadar çizgisel olarak ayrılabilen veri kümelerinde başarılı olsa da *çizgisel-olmayan biçimde birbirinden ayrılan*⁸¹ veri kümeleri için de tanımlanabilmektedir.

⁷⁷ Bu durum literatürde “hard-margin” olarak bilinmektedir.

⁷⁸ Şekil 1.20’de daire içine konan artı (+) işaretiyle temsil edilen veriler.

⁷⁹ Şekil 1.20’de içi boş daireler ile temsil edilen veriler.

⁸⁰ Çizgisel cebirden bilindiği üzere bir noktadan bir düzleme olan mesafe kolayca hesaplanabilmektedir.

⁸¹ Bu durum literatürde “soft-margin” adıyla bilinmektedir.

SVMs yaklaşımı çeşitli “gerçek dünya” problemlerinin üstesinden gelmek için kullanılabilir. Önemli kullanım alanlarına yönelik örnekleri kısaca incelemek amacıyla okuyucunun aşağıdaki listeyi kontrol etmesi faydalı olacaktır:

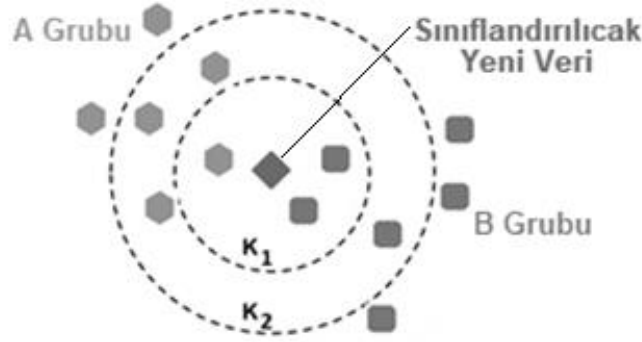
- Metin sınıflandırma: Analizin temel uygulamaları tümevarımsal ya da tümdengelsel etiketli eğitim örneklerine olan ihtiyacı önemli ölçüde azaltabilmektedir (Joachims, 1998). Sığ semantik ayrıştırma için bazı yöntemler esasen SVMs yaklaşımına dayanmaktadır (Pradhan, 2004).
- Görüntü sınıflandırma: Yapılan uygulamalar yalnızca 3-4 tur alaka düzeyi geri bildiriminden sonra geleneksel sorgu iyileştirme şemalarına göre SVMs yaklaşımlarından önemli ölçüde daha yüksek arama doğruluğu elde edilebileceğini göstermektedir. Bu durum modifiye edilmiş SVMs yaklaşımını (Barghout, 2015) kullananlar da dahil olmak üzere *görüntü bölümlene*⁸² sistemleri için de geçerlidir.
- Uydu verilerinin sınıflandırılması: Denetimli SVMs yöntemi *Sentetik Açıklıklı Radar*⁸³ gibi uydu verilerinin sınıflandırılmasında önemli sonuçlar vermektedir (Maity, 2016).
- Yazı karakterlerini tanıma: Elle yazılmış karakterler SVMs algoritmasıyla tanınabilmektedir (DeCoste, 2002; Maitra ve ark., 2015).

1.7.9. kNN

Öğrenme mimarisinin seçilen eğitim verilerine dayandığı örnek-tabanlı bir algoritma olup esasen IBL yaklaşımının daha genel biçimidir. Yaklaşım parametrik değildir ve “tembel” model olarak da bilinmektedir. İstekli öğrenmeden farklı olarak tembel öğrenmede bir eğitim aşaması yoktur. Peki algoritma nasıl çalışır? Algoritma sisteme yeni veri eklendiğinde veri havuzundan öğrendiği örüntülere bakarak bir karar mekanizmasını devreye sokmaktadır. Diğer bir deyişle, bir tahmin yapmamız gerektiğinde algoritma tüm veri kümesindeki en yakın (ya da benzer) komşu veri noktalarını arayacaktır. Sınıfı bilinmeyen yeni veri örneğinin hangi sınıfa yerleştirileceği en yakın komşularının çoğunluğu tarafından belirlenmektedir.

⁸² Dilimize böyle çevrilen yöntemin orijinal adı “image segmentation” biçimindedir.

⁸³ Kısaca “SAR” olarak bilinmekte olup literatürde “Sythetic Aperture Radar” adıyla anılmaktadır.



Şekil 1.21. kNN sınıflandırması için temsili bir görsel.

kNN yaklaşımının genel bir temsili Şekil. 1.21’de sunulmaktadır. Burada gösterilen K_1 kod adlı bölgede⁸⁴ kare şekliyle betimlenen veri noktalarının⁸⁵ sayısının diğer verilere nazaran daha fazla olması yeni örnekleminin⁸⁶ B grubuna konması gerektiğini göstermektedir. Ardından, K_2 bölgesine⁸⁷ baktığımızda bu defa altıgen biçimli şekille temsil edilen veri noktalarının⁸⁸ çoğunlukta olduğu görülmektedir, dolayısıyla yeni veri noktası A grubunun bir üyesi olacaktır.

Sınıflandırma işlerinde oldukça başarılı olan kNN yaklaşımı için en önemli görev en yakın komşuları belirleme aşamasında her bir test verisi için tüm eğitim verilerine olan mesafeyi ve/veya benzerliği ölçme yöntemidir (Zhu ve ark., 2014). Literatürde iki nokta arasındaki mesafenin ölçümünü yapabilmek için önerilmiş çeşitli formülasyonlar bulunmaktadır. Konudan sapmamak adına mesafe ölçüm modelleri bir sonraki alt-başlık altında ayrıntılı biçimde ele alınacaktır.

Yaklaşımın adının başında yazılan k çok önemli bir parametredir ve algoritmaya en yakın kaç komşu üzerinden hesaplama yapılacağını söylemektedir. Bu niceliğin alacağı değer algoritmanın tahmin sonucuna doğrudan etki etmektedir, lakin yaklaşımın kendisi en uygun k değerini kendi başına hesaplayamamaktadır yani en doğru/uygun/faydalı k değeri algoritmaya dışarıdan verilmelidir. Bu durum kNN modelinin de derin yapıli LSTM yaklaşımına benzer bir ön-ayarlılama aşaması gerektirdiğine işaret etmektedir. Dikkat edilmesi gereken noktalar şunlardır;

- Eğer $k=1$ olursa algoritmanın verileri “öğrenme” yerine “ezberleme” olasılığı artabilir,
- Çok büyük değerler kullanılırsa “uydurma” durumu gerçekleşebilir yani bu defa mekanizma bir anlam ifade etmeyen çok genel öngörüler sunacaktır.

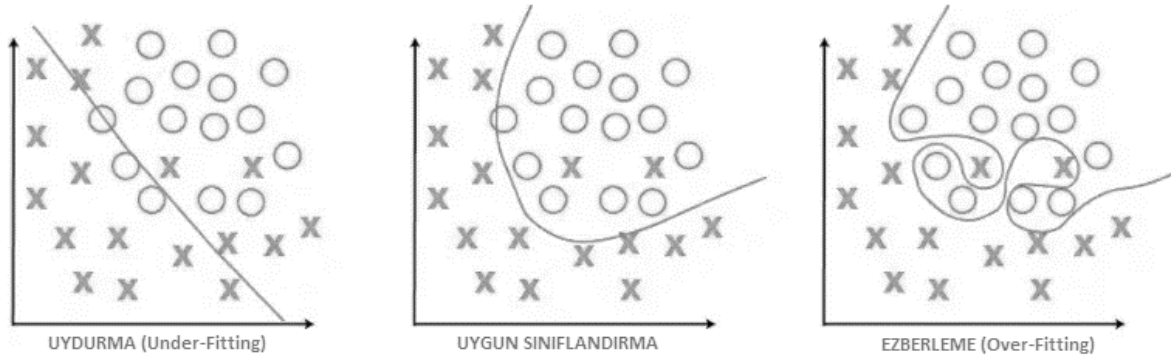
⁸⁴ İçteki çemberin sınırladığı alan.

⁸⁵ B grubu olarak adlandırılan kare şekil kategorisi.

⁸⁶ Prizma şekliyle betimlenen bilgi.

⁸⁷ Dıştaki çemberin sınırladığı alan.

⁸⁸ A grubu olarak adlandırılan altıgen şekil kategorisi.



Şekil 1.22. Sınıflandırma durumları (Hussain, 2020).

Şekil 1.22’de sınıflandırma analizi için temel olası durumlar temsili bir yöntemle resmedilmiştir⁸⁹. “Uydurma” durumunun betimlendiği en soldaki grafikten yapılan analizin çok genel olduğu ve bir ayırım yapmak için yeterli çıkarımın elde edilmediği görülmektedir. En sağda sunulan “ezberleme” durumunda ise zorlamalı bir durum olduğu açıkça anlaşılmaktadır, zira tahminler doğru olamayacak kadar iyi gözükmektedir. Ortadaki grafikte ise makul/kabul edilebilir/uygun olarak nitelendirilebilecek bir sonucu görmek mümkündür. Buradan anlaşılabileceği üzere optimum k değerine ulaşmak kNN temelli bir araştırmanın kritik bir evresi olarak karşımızda durmaktadır. kNN yaklaşımında bir diğer önemli kısım ise tamamlanan ML analizinin performansını test etmektir (Caruana ve Niculescu-Mizil, 2004). Bir yapay öğrenme sürecinin performans değerlendirmesi bir dizi metrik kriter aracılığıyla yapılabilmektedir ve literatürde çeşitli değerlendirme yöntemleri önerilmiştir. Bu modellerden öne çıkanlar ayrı bir başlık altında tezin ilerleyen kısımlarında sunulacaktır, ancak sunumda bütünlük sağlamak adına kNN algoritmasında kritik öneme sahip olan ve tez kapsamında sıklıkla kullanılan bazı kriterlerden burada bahsetmek yerinde olacaktır.

Değerlendirme metrikleri algoritmanın eğitim-test aşaması sonrasında en uygun k değerinin belirlenmesi için karşılaştırma temelli kullanışlı bir yöntem kullanabilme olanağı sunmaktadır. Bu bağlamda, *Karışıklık Matrisi*⁹⁰ bilinen önemli test araçları arasında yer almakta ve bileşenleri bazı değerlendirme metriklerinde kullanılmaktadır. Bu matrisin bileşenleri yapılan bir sınıflandırma analizinin sonuçlarının *Doğru*⁹¹, *Yanlış*⁹², *Pozitif (ya da olumlu)* ve *Negatif* (veya olumsuz) seçeneklerine göre değerlendirildiği bir bakış açısıyla ele alınmaktadır. Şekil 1.23’te karışıklık matrisinin genel bir temsili ve bazı önemli performans test metrikleri sunulmaktadır.

⁸⁹ “Uydurma” ve “Ezberleme” olarak çevrilen kavramların literatürdeki adları faydalı olması adına şekilde parantez içerisinde vurgulanmıştır.

⁹⁰ Literatürde kullanılan adıyla “Confusion Matrix”.

⁹¹ Orijinal haliyle “True” olarak bilinmekte olup “Var” ve “Evet” veri seçeneklerini de bu kategoriye sokmak mümkündür.

⁹² Terminolojideki karşılığı “False” kelimesi olup “Yok” ve “Hayır” veri seçenekleri de bu etiket ismiyle nitelendirilebilmektedir.

		Tahmini Veri			Toplamlar
		Pozitif (1)	Negatif (0)		
Gerçek Veri	Pozitif (1)	DP	YN	Hassasiyet ⁹³ (Pozitif Hatırlama ⁹⁴) $\frac{DP}{DP+YN}$	Gerçek Pozitifler $P = DP + YN$
	Negatif (0)	YP	DN	Özgüllük ⁹⁵ (Negatif Hatırlama) $\frac{YN}{DP+DN}$	Gerçek Negatifler $N = YP + DN$
		Pozitif Kesinlik ⁹⁶ $\frac{DP}{DP+YP}$	Negatif Kesinlik $\frac{DN}{DN+YN}$	Tutarlılık ⁹⁷ $\frac{DP+DN}{DP+YN+YP+DN}$	
Toplamlar		Tahmini Pozitifler $P = DP + YP$	Tahmini Negatifler $N = YN + DN$		

Şekil 1.23. Karışıklık matrisi ve bazı performans metrikleri.

Şekil 1.23'te verilen karışıklık matrisi (renkli kısımlar) için satırlar gerçek verilere karşılık gelirken sütunlar ise tahmini verileri ifade etmektedir. Ayrıca, DP, YN, YP ve DN kısaltmaları sırasıyla Doğru Pozitif (seçilen model tarafından doğru olarak pozitiflik tahmini yapılırken gerçek verilerde de pozitiflik vardır), Yanlış Negatif (verilerde gerçek bir negatiflik olup seçilen algoritma tarafından doğru bir şekilde negatiflik tahmini yapılmıştır), Yanlış Pozitif (verilerde gerçek bir olumsuzluk vardır ancak seçilen yaklaşım bunu pozitif olarak tahmin etmiştir) ve Gerçek Negatif (verilerde gerçek bir pozitiflik vardır ancak model bunu negatif olarak tahmin etmiştir) anlamına gelmektedir. Şekil 1.23'te ayrıntılı olarak betimlenen performans metriklerine ek olarak F1-puanı⁹⁸ olarak bilinen önemli bir değerlendirme metriği daha bulunmaktadır. Bu metrik 0 ile 1 arasında değerler alır ve 1 değeri en iyi başarımlı duruma işaret eder. Karışıklık matrisi bileşenleri yardımıyla tanımlanan “kesinlik” ve “hatırlama” metrikleri kullanılarak F1-puanı şöyle hesaplanmaktadır:

⁹³ Literatürde “Sensitivity” olarak anılmaktadır.

⁹⁴ İstatistikte “Recall” adıyla bilinmektedir.

⁹⁵ Orijinal adı “Specificity” şeklinde yazılır.

⁹⁶ Alan terminolojisinde “Precision” kavramı olarak bilinir.

⁹⁷ Bu kavramın orijinal adı “Accuracy” şeklinde yazılırken dilimize bazı kaynaklarda “Aküresi” olarak aktarılmıştır.

⁹⁸ Literatürde “F-measure”, “F-score” ya da “F1-score” olarak bilinmektedir.

$$F1 - \text{puanı} = 2 \frac{\text{Kesinlik} \times \text{Hatırlama}}{\text{Kesinlik} + \text{Hatırlama}} \quad (1.39)$$

Önemli ML yaklaşımları arasında gösterilen kNN algoritmasının sıklıkla kullanıldığı “geçek dünya” problemlerinin bazı örnekleri şunlardır (Jayaswal, 2020; Arslan, 2020):

- Metin madenciliği (anlamsal olarak benzer belgelerin aranması):
Her belge bir vektör olarak kabul edilir ve analiz sonucunda dokümanların birbirine yakın olması aynı konuları içerdiği anlamına gelecektir.
- Aykırı değerleri tespit etmek:
Örneğin kredi kartı dolandırıcılığı tespiti bu yöntemle yapılabilmektedir.
- Öneri sistemleri (Amazon, Netflix, Facebook, Google, Youtube, vs.):
kNN yaklaşımı çok yüksek boyutlu ve karmaşık veriler için uygun olmayabilir, ancak bazı sistemler için mükemmel bir mekanizmadır.
- Müşteri Temerrüt Riski⁹⁹:
Bir müşterinin kullanacağı krediyi ödeme imkanının olup olmadığı ile ilgili olarak bir tahmin yürütülmesidir.
- Müşteri Kaybı¹⁰⁰:
Almakta olduğu servisi ya da hizmeti bir müşterinin devam ettirip ettirmeyeceği bu yöntemle öngörülebilmektedir.
- Spam elektronik postaların filtrelenmesinde kullanılabilmektedir.
- Tıp: Örneğin bir kişinin hastalığına özel ilaç belirleme.

Yukarıdaki durumların dışında kNN yaklaşımı aşağıda sıralan alanlarda da etkin biçimde kullanılabilmektedir (Hussain, 2020; Arslan, 2020):

- Tarım,
- Yüz tanıma ve el yazısı tanıma,
- Biyometrik tanımlama.

1.8. Mesafe Ölçümü Modelleri

IBL ve kNN gibi en yakın komşuluk temelli algoritmalar için veri noktaları arasındaki mesafenin nasıl ölçüleceği çok önemlidir. İki farklı öznitelik grubunu betimleyen $A = (x_1, x_2, \dots, x_k)$ ve $B = (y_1, y_2, \dots, y_k)$ vektör bileşenleri arasındaki mesafe $d(x, y)$ biçiminde adlandırılan ve negatif olmayan gerçel değerlere sahip bir fonksiyondur (Alfeilat ve ark., 2019). Bu fonksiyon aşağıdaki koşullardan oluşan belirli sayıda özelliğe uyarsa bir metrik olduğu varsayılır (Deza ve Deza, 2009):

- Pozitiflik (negatif olmama): Mesafe sıfırdan büyük veya sıfıra eşit olmalıdır;

⁹⁹ Dilimize böyle çevrilen kavramın literatürdeki karşılığı şöyledir: “Customer Default Risk”.

¹⁰⁰ Terminolojik karşılığı “Customer Churn” şeklinde yazılır.

$$d(x, y) \geq 0. \quad (1.40.)$$

- Ayrılmazların kimliği: x noktası y 'ye eşit olduğunda mesafe sifıra eşittir;

$$d(x, y) = 0 \leftrightarrow x = y. \quad (1.41.)$$

- Simetri: x ile y arasındaki mesafe y ile x arasındaki mesafeye eşittir;

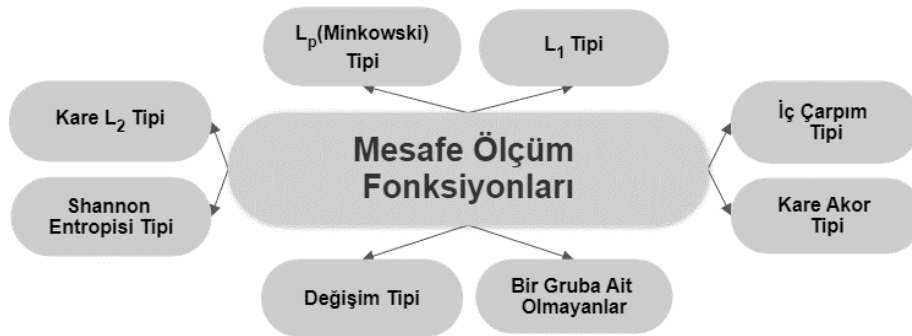
$$d(x, y) = d(y, x). \quad (1.42.)$$

- Üçgen eşitsizliği: Arada üçüncü bir z noktasının var olduğunu varsayarsak, $d(x, y)$ mesafesi her zaman $d(x, z)$ mesafesi ile $d(z, y)$ mesafesinin toplamından küçük veya ona eşittir,

$$d(x, y) = d(x, z) + d(z, y). \quad (1.43.)$$

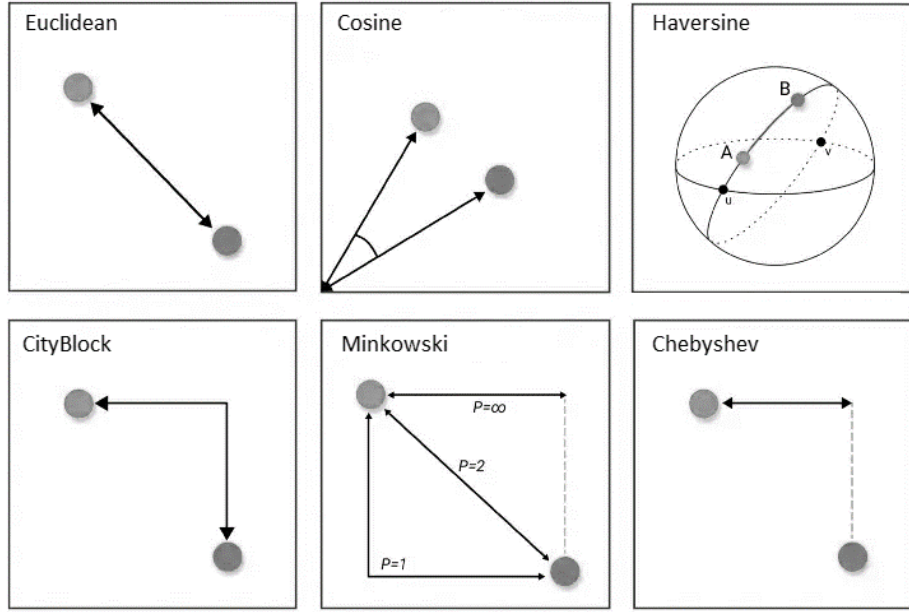
Ayrıca, $0 \leq d(x, y) \leq 1$ olduğunda benzerlik ölçüsü şu şekilde tanımlanır (Alfeilat ve ark., 2019):

$$s(x, y) = 1 - d(x, y). \quad (1.44.)$$



Şekil 1.24. Popüler mesafe ölçüm fonksiyonu tipleri.

Literatürde onlarca mesafe ölçüm formülü önerilmiştir (Cha, 2007; Deza ve Deza, 2009; Alfeilat ve ark., 2019; Prasath ve ark., 2019) ve bu modeller Şekil 1.24'te görüldüğü gibi sekiz kategori altında sınıflandırılabilir. Diğer yandan, mesafe ölçüm modelleri inşa edilirken tercih edilen çeşitli yöntemler için bazı örnekler Şekil 1.25'te sunulmuştur.



Şekil 1.25. Bazı mesafe ölçüm yöntemleri (Grootendorst, 2021).

Şimdi en bilinen mesafe ölçüm modellerini biraz daha ayrıntılı betimleyelim.

- L_p (Minkowski) tipi mesafe modelleri:

Bu mesafe ailesi üç tane özel modelleme içermektedir. Aralarındaki ayrım p 'nin değerinde gizlenmektedir ve genel olarak aile şöyle betimlenir (Prasath ve ark., 2019)

$$d_{Minkowski}(x, y) = (\sum_{i=1}^n |x_i - y_i|^p)^{\frac{1}{p}}. \quad (1.45)$$

Burada p 'nin alacağı değere göre modelin ismi değişmektedir. Yukarıda $p=1$ olursa CityBlock¹⁰¹ tipi mesafe elde edilmektedir, yani

$$d_{CityBlock}(x, y) = \sum_{i=1}^n |x_i - y_i|, \quad (1.46)$$

olmaktadır. Diğer yandan, $p=2$ olduğunda ise Euclidean mesafe modeline ulaşılmaktadır, böylece

$$d_{Euclidean}(x, y) = (\sum_{i=1}^n |x_i - y_i|^2)^{\frac{1}{2}} \quad (1.47)$$

denklemini elde edilecektir. Son olarak, $p=\infty$ seçildiğinde ise Chebyshev¹⁰² modeli tanımlanmaktadır (Grabusts, 2011; Premaratne, 2014; Todeschini ve ark., 2015):

$$d_{Chebyshev}(x, y) = \max_i |x_i - y_i|. \quad (1.48)$$

¹⁰¹ Bu model literatürde "Manhattan", "Taxicab", "Rectilinear" ya da " L_1 -norm" olarak da bilinmektedir.

¹⁰² Literatürde "Maximum Value Distance (Maksimum Değer Mesafesi)", "Lagrange" veya "Chessboard" adıyla da anılmaktadır.

- L_1 tipi mesafe modelleri:

Bu mesafe ailesi temel olarak veri noktaları arasındaki mutlak farkı bulmaya bağlıdır ve Lorentzian, Canberra, Sorensen, Soergel, Kulczynski ve Ortalama Karakter¹⁰³ olarak bilinen mesafe modellerini içermektedir (Prasath ve ark., 2019). Lorentz mesafesinde iki vektör arasındaki mutlak fark doğal logaritma ile temsil edilmektedir. Formülasyondaki logaritma ölçeği alt aralığı genişletip üst aralığı sıkıştırmaktadır, bu nedenle Lorentz mesafe modeli küçük değişikliklere karşı hassastır.

$$d_{Lorentz}(x, y) = \sum_{i=1}^n \ln(1 + |x_i - y_i|). \quad (1.49)$$

Hesaplanacak nihai sonucun negatif olmama özelliğinin sağlanması ve logaritmanın sıfır olmasını önlemek için ifade “+1” eklentisi içermektedir. Canberra mesafe modeli esasen Manhattan mesafesinin ağırlıklı bir versiyonudur (Williams ve Lance, 1966; Lance ve Williams, 1967). Burada, x ve y vektörlerinin özellik değerleri arasındaki mutlak fark mutlak özellik değerlerinin toplamına bölünmektedir (Akila ve Chandra, 2013). Bu mesafe aslında pozitif değerler için kullanılır ve sıfıra yakın küçük değişikliklere karşı çok duyarlıdır, dolayısıyla modelin bu özelliği artan sayıda değişkenle daha yüksek boyutlu uzayda daha belirgin hale gelmektedir (Prasath ve ark., 2019). Canberra mesafesi genellikle bir orijinin etrafına dağılmış veriler için kullanılmaktadır:

$$d_{Canberra}(x, y) = \sum_{i=1}^n \frac{|x_i - y_i|}{|x_i| + |y_i|}. \quad (1.50)$$

Sorensen¹⁰⁴ mesafesi (Sorensen, 1948) ekoloji, çevre bilimleri ve ilgili alanlarda bağlantıları (ilişkileri) ifade etmek için en sık uygulanan ölçüm yöntemidir (Prasath ve ark., 2019). Aslında x ve y vektörlerinin öz nitelik değerleri arasındaki toplam farkların öz nitelik değerleriyle standartlaştırıldığı değiştirilmiş bir Manhattan metriğidir (Szmıdt, 2013):

$$d_{Sorensen}(x, y) = \frac{\sum_{i=1}^n |x_i - y_i|}{\sum_{i=1}^n x_i + y_i}. \quad (1.51)$$

Tüm vektör değerleri pozitif olduğunda bu ölçü sıfır ile bir arasında değer almaktadır. Soergel¹⁰⁵ mesafesi evrimsel mesafeyi hesaplamak için yaygın olarak kullanılan mesafe ölçüleri arasında olup (Chetty ve ark., 2008) ikili değişkenler için bu uzaklık tanımı Tanimoto¹⁰⁶ benzerlik katsayısının tamamlayıcısı ile aynıdır (Zhou ve ark. 2008). Bu mesafe negatif olmayan değerlere sahip tüm

¹⁰³ Literatürdeki adı “Mean Character” olarak yazılır.

¹⁰⁴ Diğer adı Bray-Curtis modelidir.

¹⁰⁵ Bu yaklaşım literatürde “Ruzicka” modeli olarak da anılmaktadır.

¹⁰⁶ Ya da “Jaccard” modeli olarak bilinir.

öznitelikler tarafından sağlanan dört metrik özelliğinin tümüne de uymakta (Willett ve ark., 1998) ve şöyle tanımlanmaktadır:

$$d_{Soergel}(x, y) = \frac{\sum_{i=1}^n |x_i - y_i|}{\sum_{i=1}^n \max(x_i, y_i)}. \quad (1.52)$$

Kulczynski mesafesinin tanımı Soergel mesafesine benzemektedir, ancak “maksimum” fonksiyonu yerine burada “minimum” fonksiyonunu kullanmaktadır (Prasath ve ark., 2019):

$$d_{Kulczynski}(x, y) = \frac{\sum_{i=1}^n |x_i - y_i|}{\sum_{i=1}^n \min(x_i, y_i)}. \quad (1.53)$$

Ortalama Karakter Mesafesi¹⁰⁷ kısaca OKM olarak adlandırılabilir ve şöyle betimlenmektedir (Prasath ve ark., 2019):

$$d_{OKM}(x, y) = \frac{\sum_{i=1}^n |x_i - y_i|}{n}. \quad (1.54)$$

- İç çarpım mesafe ölçüleri:

Bu mesafe ölçümü ailesi Jaccard, Cosine¹⁰⁸, Dice ve Chord¹⁰⁹ modellerini içermektedir. Jaccard mesafesi temel olarak örnek kümeler arasındaki farklılığı ölçmektedir (Jaccard, 1901). Bu ölçüm modeli esasen Jaccard benzerlik katsayısının tamamlayıcısıdır ve Jaccard katsayısının 1’den çıkarılmasıyla elde edilmektedir (Cesare ve Xiang, 2012):

$$d_{Jaccard}(x, y) = \frac{\sum_{i=1}^n (x_i - y_i)^2}{\sum_{i=1}^n x_i^2 + \sum_{i=1}^n y_i^2 - \sum_{i=1}^n (x_i - y_i)}. \quad (1.55)$$

Açısal mesafe olarak da adlandırılan Cosine mesafesi, iki vektör arasındaki açıyı ölçen kosinüs benzerliğinden türetilmektedir:

$$d_{Cosine}(x, y) = 1 - \frac{\sum_{i=1}^k x_i y_i}{\sqrt{\sum_{i=1}^k x_i^2} \sqrt{\sum_{i=1}^k y_i^2}}. \quad (1.56)$$

Dice mesafesi temel olarak Dice benzerliğinin türetilir (Dice, 1945) ve sıfıra yakın değerlere karşı duyarlı olabilmektedir. Bu uzaklık tam bir metrik değildir, çünkü üçgen eşitsizliği özelliği sağlamamaktadır. Model genellikle belgelerden ve biyolojik taksonomiden bilgi alımında yaygın olarak kullanılmaktadır (Prasath ve ark., 2019):

¹⁰⁷ Literatürde “Ortalama Manhattan” veya “Gower Mesafesi” olarak da bilinmektedir.

¹⁰⁸ Kosinüs.

¹⁰⁹ Akort.

$$d_{Dice}(x, y) = 1 - \frac{2 \sum_{i=1}^k x_i y_i}{\sum_{i=1}^k x_i^2 + \sum_{i=1}^k y_i^2}. \quad (1.57)$$

Orloci tarafından topluluk bileşimi verilerini (Legendre ve Legendre, 2012) analiz etmek için önerilen (Orloci, 1967) Chord modeli Euclidean modelin bir nevi modifiye edilmiş formudur (Gan, 2007):

$$d_{Chord}(x, y) = \sqrt{2 - \frac{2 \sum_{i=1}^k x_i y_i}{\sum_{i=1}^k x_i^2 \sum_{i=1}^k y_i^2}}. \quad (1.58)$$

Bu mesafe ölçüsü yarıçapı 1 olan bir hiper-küre içindeki iki normalize edilmiş noktayı birleştiren kirişin uzunluğu olarak tanımlanmıştır ve model genellikle sürekli verileri kategorize etmek amacıyla kullanılmaktadır (Shirkhorshidi, 2015).

- Kare akort mesafe ölçüleri:

Bu aileye ait mesafeler geometrikler¹¹⁰ cinsinden tanımlanmaktadır ve negatif değerli özellik vektörü ile kullanılamazlar. Bhattacharyya mesafesi iki olasılık dağılımının benzerliğini ölçmektedir ve şöyle tanımlanır (Bhattachayya, 1943)

$$d_{Bhattachayya}(x, y) = -\ln\left(\sum_{i=1}^k \sqrt{x_i y_i}\right). \quad (1.59)$$

Squared-Chord yaklaşımı daha çok paleontolojide ve polen araştırmalarında kullanılmaktadır. Örnek iki vektör boyunca her noktadaki karekök farkının karesinin toplamı alınmaktadır, zira bu durum daha farklı özellik için farkı arttırmaktadır (Prasath ve ark., 2019):

$$d_{Squared-Chord}(x, y) = \sum_{i=1}^n \sqrt{(x_i - y_i)^2}. \quad (1.60)$$

Matusita mesafesi Squared-Chord yaklaşımında sunulan tanımın kareköküdür:

$$d_{Squared-Chord}(x, y) = \sqrt{\sum_{i=1}^n \sqrt{(x_i - y_i)^2}}. \quad (1.61)$$

Hellinger mesafesi¹¹¹ (Hellinger, 1909; Abbad ve Tairi, 2016). İki olasılık dağılımı arasındaki benzerliği ölçmek için kullanılan bir metrik olup Bhattacharyya modeli ile yakından ilişkilidir:

$$d_{Hellinger}(x, y) = \sqrt{2 \sum_{i=1}^n (\sqrt{x_i} - \sqrt{y_i})^2}. \quad (1.62)$$

¹¹⁰ Burada geometrik ortalama kastedilmektedir. İki değerin geometrik ortalaması çarpımlarının kareköküdür.

¹¹¹ Literatürde Jeffries - Matusita mesafesi olarak da bilinmektedir.

- Kare L_2 tipi mesafeler:

Oldukça geniş bir ailedir. En bilinen örnekleri

- Kare Euclidian (KaE) (Prasath ve ark., 2019),
- Clark (Clark, 1952),
- Neyman Ki-Kare (NeKiK) (Neyman, 1949),
- Pearson Ki-Kare (PeKiK) (Pearson, 1900),
- Kare Ki-Kare¹¹² (KaKiK) (Prasath ve ark., 2019),
- Olasılıklı Simetrik¹¹³ Ki-Kare (OSKiK) (Prasath ve ark., 2019),
- Divergence (Prasath ve ark., 2019),
- Katkılı Simetrik¹¹⁴ Ki-Kare (KaSKiK) (Prasath ve ark., 2019),
- Avarage (Shirkhorshidi ve ark., 2015),
- Ortalama Sansürlü¹¹⁵ Euclidean (OSE) (Prasath ve ark., 2019)
- Kareli Ki-Kare Tip-2 (KaKiK2) (Prasath ve ark., 2019)

modelleridir. Bu modelleri betimleyen matematiksel bağıntılar aşağıda sırasıyla listelenmiştir:

$$d_{KaE}(x, y) = \sum_{i=1}^n (x_i - y_i)^2, \quad (1.63)$$

$$d_{Clark}(x, y) = \sqrt{\sum_{i=1}^n \left(\frac{x_i - y_i}{|x_i| + |y_i|} \right)^2}, \quad (1.64)$$

$$d_{NeKiK}(x, y) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{x_i}, \quad (1.65)$$

$$d_{PeKiK}(x, y) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{y_i}, \quad (1.66)$$

$$d_{KaKiK}(x, y) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{x_i + y_i}, \quad (1.67)$$

$$d_{OSKiK}(x, y) = 2 \sum_{i=1}^n \frac{(x_i - y_i)^2}{x_i + y_i}, \quad (1.68)$$

$$d_{Divergence}(x, y) = 2 \sum_{i=1}^n \frac{(x_i - y_i)^2}{(x_i + y_i)^2}, \quad (1.69)$$

$$d_{KaSKiK}(x, y) = 2 \sum_{i=1}^n \frac{(x_i - y_i)^2 (x_i + y_i)}{x_i y_i}, \quad (1.70)$$

¹¹² Squared Chi-Squared.

¹¹³ Probabilistic Symmetric.

¹¹⁴ Additive Symmetric.

¹¹⁵ Mean Censored.

$$d_{Average}(x, y) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2}, \quad (1.71)$$

$$d_{OSE}(x, y) = \sqrt{\frac{\sum_{i=1}^n (x_i - y_i)^2}{\sum_{i=1}^n 1_{x_i^2 + y_i^2 \neq 0}}}, \quad (1.72)$$

$$d_{KaKiK2}(x, y) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{|x_i + y_i|}. \quad (1.73)$$

- Shannon entropisi temelli modeller:

Bu aileye ait uzaklık ölçüleri Shannon entropisi (Shannon, 1948) ile yakından ilişkilidir. Bu aileye giren önemli mesafe modelleri ise şunlardır:

- Kullback-Leibler¹¹⁶ (Kullback ve Leibler, 1951),
- Jeffreys (Jeffreys, 1946),
- K-diverjansı¹¹⁷ (Prasath ve ark., 2019),
- Topsoe¹¹⁸ (Topsoe, 2000),
- Jensen-Shannon (Prasath ve ark., 2019),
- Jensen Farkı (Sibson, 1969).

Kullback-Leibler modeli temel olarak iki olasılık dağılımı arasındaki farkı ölçmektedir. Bu mesafe modeli simetrik olmadığından ve üçgen eşitsizlik özelliğini sağlamadığından metrik bir ölçü değildir ve literatürde “yarı” mesafe olarak kabul edilmektedir (Prasath ve ark., 2019). Kullback-Leibler mesafesi sorgu genişletme, dil modelleri ve sınıflandırma gibi çeşitli doğal dil uygulamalarında kullanılabilmektedir (Pinto ve ark., 2007). Diğer yandan, Jeffreys ve Topsoe modelleri aslında Kullback-Leibler mesafesinin simetrik biçimleridir. Topsoe mesafesi Jensen-Shannon sapmasının iki katıdır ve bir metrik değildir, ancak karekökü bir metrik ölçüdür. Jensen-Shannon mesafesi temel olarak Jensen-Shannon sapma miktarının karekökü olup K-sapması niceliğini simetrik yapmak için ortalama yöntemi kullanan Topsoe mesafesinin ise yarısı kadardır. Modelleri temsil eden matematiksel tanımlar şunlardır:

$$d_{Kullback-Leibler}(x, y) = \sum_{i=1}^n x_i \ln \left(\frac{x_i}{y_i} \right), \quad (1.74)$$

$$d_{Jeffreys}(x, y) = \sum_{i=1}^n (x_i - y_i) \ln \left(\frac{x_i}{y_i} \right), \quad (1.75)$$

¹¹⁶ Literatürde ayrıca “KL divergence”, “relative entropy” ya da “information deviation” adlarıyla anılmaktadır.

¹¹⁷ “J-divergence” ya da “KL2-distance” olarak da bilinmektedir.

¹¹⁸ Diğer adı “information statistics (bilgi istatistiği)” biçimindedir.

$$d_{K-divergans}(x, y) = \sum_{i=1}^n x_i \ln \left(\frac{2x_i}{x_i+y_i} \right), \quad (1.76)$$

$$d_{Topsoe}(x, y) = \sum_{i=1}^n x_i \ln \left(\frac{2x_i}{x_i+y_i} \right) + \sum_{i=1}^n y_i \ln \left(\frac{2y_i}{x_i+y_i} \right), \quad (1.77)$$

$$d_{Jensen-Shannon}(x, y) = \frac{1}{2} \left[\sum_{i=1}^n x_i \ln \left(\frac{2x_i}{x_i+y_i} \right) + \sum_{i=1}^n y_i \ln \left(\frac{2y_i}{x_i+y_i} \right) \right], \quad (1.78)$$

$$d_{Jensen}(x, y) = \frac{1}{2} \left[\sum_{i=1}^n \frac{x_i \ln x_i + y_i \ln y_i}{2} - \frac{x_i+y_i}{2} \ln \left(\frac{x_i+y_i}{2} \right) \right]. \quad (1.79)$$

- Değişim mesafesi ölçümleri:

Söz-dizimsel¹¹⁹ ilişkiden türetilen bu mesafe ailesi Vicis-Wave Hedges¹²⁰ (VWH), Vicis Symmetric (VS), Maximum Symmetric Ki-Kare (MaSKiK) ve Minimum Simetrik Ki-Kare (MiSKiK) mesafeleri olmak üzere dört modelden oluşmaktadır. Vicis-Wave Hedges modeli literatürde sıkıştırılmış görüntü alma (Hatzigiorgaki ve Skodras, 2003), içerik tabanlı video alma (Patel ve Meshram, 2012), zaman serisi sınıflandırması (Giusti ve Batista, 2013), görüntü keskinliği (Macklem, 2002) ve parmak izi tanıma (Bharkad ve Kokare, 2011) konularında kullanılmıştır. Vicis Simetrik mesafesi üç formülle tanımlanmaktadır. Yukarıda bahsi geçen yaklaşımlar matematiksel olarak aşağıdaki gibi tanımlanır:

$$d_{VWH}(x, y) = \sum_{i=1}^n \frac{|x_i - y_i|}{\min(x_i, y_i)}, \quad (1.80)$$

$$d_{VS1}(x, y) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{(\min(x_i, y_i))^2}, \quad (1.81)$$

$$d_{VS2}(x, y) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{\min(x_i, y_i)}, \quad (1.82)$$

$$d_{VS3}(x, y) = \sum_{i=1}^n \frac{(x_i - y_i)^2}{\max(x_i, y_i)}, \quad (1.83)$$

$$d_{MaSKiK}(x, y) = \max \left(\sum_{i=1}^n \frac{(x_i - y_i)^2}{x_i}, \sum_{i=1}^n \frac{(x_i - y_i)^2}{y_i} \right), \quad (1.84)$$

$$d_{MiSKiK}(x, y) = \min \left(\sum_{i=1}^n \frac{(x_i - y_i)^2}{x_i}, \sum_{i=1}^n \frac{(x_i - y_i)^2}{y_i} \right). \quad (1.85)$$

¹¹⁹ Syntactic.

¹²⁰ Literatürde "Wave-Hedges" adıyla da bilinmektedir.

- Bir gruba ait olmayan modeller:

Bu metrikler esasen birden çok fikri veya buraya kadar olan kısımlarda bahsi geçen mesafe modellerini temel alan yaklaşımlardır. Bu modellerden öne çıkanlar şöyle listelenebilir (Prasath ve ark., 2019):

- Ortalama-(L_1, L_∞) (kısaca OLL),
- Kumar-Johnson, Taneja (Taneja, 1995),
- Korelasyon¹²¹,
- Ki-Kare Pearson (KiKP),
- Hamming (Hamming, 1958),
- Hausdorff,
- Ki-kare İstatistiği¹²² (KiKİ),
- Whittaker'ın Çağrışım Endeksi¹²³ (WÇE) (Whittaker, 1952),
- Meehl,
- Motyka,
- Hassanat (Hassanat, 2014).

Buradaki Ortalama-(L_1, L_∞) mesafesi temel olarak Manhattan ve Chebyshev mesafelerinin ortalaması biçiminde tanımlanmaktadır. Pearson mesafesi iki vektör arasındaki doğrusal ilişkiyi ölçen Pearson korelasyon katsayısından türetilmiştir (Fulekar, 2009). Hamming mesafesi iki vektör arasındaki uyumsuzluk sayısını ölçen bir mesafe ölçüsü olup çoğunlukla nominal veriler ile birlikte Sicim¹²⁴ ve Bit-wise analizleri için kullanılmaktadır, ancak sayısal veriler için de yararlı olabilmektedir (Prasath ve ark., 2019). Bahsi geçen modeller matematiksel olarak şöyle formülize edilir:

$$d_{OLL}(x, y) = \frac{\sum_{i=1}^n |x_i - y_i| + \max_i |x_i - y_i|}{2}, \quad (1.86)$$

$$d_{Kumar-Johnson}(x, y) = \sum_{i=1}^n \frac{(x_i^2 + y_i^2)^2}{2^3 \sqrt{x_i y_i}}, \quad (1.87)$$

$$d_{Taneja}(x, y) = \sum_{i=1}^n \frac{x_i + y_i}{2} \ln \frac{x_i + y_i}{2 \sqrt{x_i y_i}}, \quad (1.88)$$

$$d_{Pearson}(x, y) = 1 - \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2 \times \sum_{i=1}^n (y_i - \bar{y})^2}, \quad (1.89)$$

¹²¹ Orijinal adı "Correlation" şeklinde yazılmaktadır.

¹²² Terminolojideki karşılığıyla "Chi-Squared Statistics"

¹²³ Literatürde bilinen adıyla "Whittaker's index of association".

¹²⁴ İngilizcedeki karşılığı ve terminolojideki adı "String" olarak yazılır.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad (1.90)$$

$$d_{Korelasyon}(x, y) = \frac{1}{2} \left(1 - \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2 \times \sum_{i=1}^n (y_i - \bar{y})^2} \right), \quad (1.91)$$

$$d_{KiKP}(x, y) = \frac{1}{2} \left(1 - \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2 \times \sum_{i=1}^n (y_i - \bar{y})^2} \right)^2, \quad (1.92)$$

$$d_{Hamming}(x, y) = \sum_{i=1}^n 1_{x_i \neq y_i}, \quad (1.93)$$

$$d_{Hausdorff}(x, y) = \max[h(x, y), h(y, x)], \quad (1.94)$$

$$h(x, y) = \max_{x_i \in x} \min_{y_i \in y} \|x_i - y_i\|, \quad (1.95)$$

$$d_{KiKl}(x, y) = \sum_{i=1}^n \frac{x_i - m_i}{m_i}, \quad (1.96)$$

$$m_i = \frac{x_i + y_i}{2}, \quad (1.97)$$

$$d_{W\check{C}H}(x, y) = \frac{1}{2} \sum_{i=1}^n \left| \frac{x_i}{\sum_{i=1}^n x_i} - \frac{y_i}{\sum_{i=1}^n y_i} \right|, \quad (1.98)$$

$$d_{Meehl}(x, y) = \sum_{i=1}^{n-1} (x_i - y_i - x_{i+1} + y_{i+1})^2, \quad (1.99)$$

$$d_{Motyka}(x, y) = \frac{\sum_{i=1}^n \max(x_i, y_i)}{\sum_{i=1}^n (x_i + y_i)}, \quad (1.100)$$

$$d_{Hassanat}(x, y) = \sum_{i=1}^n H(x_i, y_i), \quad (1.101)$$

$$H(x_i, y_i) = \begin{cases} 1 - \frac{1 + \min(x_i, y_i)}{1 + \max(x_i, y_i)} & \min(x_i, y_i) \geq 0 \\ 1 - \frac{1 + \min(x_i, y_i) + |\min(x_i, y_i)|}{1 + \max(x_i, y_i) + |\min(x_i, y_i)|} & \min(x_i, y_i) < 0. \end{cases} \quad (1.102)$$

Hassanat mesafesi $[0, 1]$ aralığında sınırlanmıştır (Hassanat, 2014). Minimumun sonlu olduğu varsayımıyla maksimum değer sonsuza yaklaştığında veya maksimumun sonlu olduğu varsayılarak minimum değer sonluda eksiye yaklaştığı zaman 1'e ulaşır (Prasath ve ark., 2019):

$$\left. \begin{aligned} \lim_{\max(A_i, B_i) \rightarrow \infty} H(A_i, B_i) \\ \lim_{\max(A_i, B_i) \rightarrow -\infty} H(A_i, B_i) \end{aligned} \right\} = 1. \quad (1.103)$$

Tüm metrik özellikleri sağlayan bu mesafe tam bir metriktir (Hassanat, 2014). Bu metrikte iki değer arasındaki fark ne olursa olsun mesafe 0 ile 1 aralığında olacaktır. Yani maksimum uzaklık test edilen vektörlerin boyutuna yaklaşır, dolayısıyla boyutlardaki artış en kötü durumda mesafeyi lineer olarak artırır (Prasath ve ark., 2019).

1.9. Performans Metrikleri

Performans ölçümleri her ML ardışık düzeninin önemli bir parçası olup size yaptığınız bir analizde ilerleme kaydedip kaydetmediğinizi söyler ve çalışmanıza bir puan verirler. Zira, kullanılan ML yaklaşımının eldeki verileri nasıl gördüğünü bilmek önemlidir ve tüm yapay öğrenme yaklaşımları performansını değerlendirmek için bir metriğe ihtiyaç duymaktadır. Tablo 1.1.'de Kaggle ve diğer popüler veri seti kaynakları sunulmuştur. Bu platformlarda zaman zaman bazı veri kümelerinin analiz edilmesine yönelik para ödüllü yarışmalar düzenlenmektedir ve katılımcılardan ilgili veriyi en başarılı biçimde okuyabilen ve sonrasında tutarlı tahminler sunabilen yapay zekâ mimarisini inşa etmeleri beklenmektedir. Böylesi yarışmalarda bir değerlendirme bölümü bulunur ve burada yarışmaya katılan kişinin önerdiği algoritmanın performansının değerlendirildiği bir ölçüt sunulur. Metrik ile *kayıp*¹²⁵ fonksiyonu karıştırılmamalıdır, zira aralarında bir nüans yani ince bir ayrım bulunmaktadır. “Kayıp” fonksiyonu bir modelin performansının bir nevi ölçüsünü gösterir, ancak esasen bir ML mimarisini *Dereceli Açılma*¹²⁶ gibi bir tür optimizasyon tekniği yardımıyla eğitmek için kullanılmaktadır ve genellikle kullanılan modelin parametrelerine göre farklılaşır. Diğer yandan, metrikler de bir modelin eğitim ve/veya test esnasında performansını izlemek ve ölçmek için kullanılır, ancak ayırt edilebilir olmaları gerekmemektedir. Performans ölçütleri tıpkı ML yaklaşımlarında olduğu gibi *regresyon* ve *sınıflandırma* biçiminde gruplandırılabilir (Bajaj, 2023). Her iki durum için de düzinelerce ölçüm modeli vardır, ancak bu tez çalışmasında en çok kullanılan yaklaşımlardan bahsedilecektir.

- **Regresyon metrikleri**

- Ortalama Karesel Hata¹²⁷:

Regresyon problemleri için kullanılan en popüler metrik olup temel olarak hedef değer ile regresyon modeli tarafından tahmin edilen değer arasındaki kare farkının ortalamasını hesaplamaktadır:

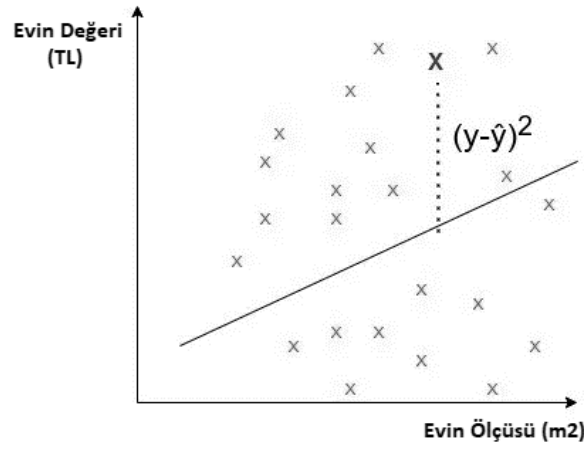
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2. \quad (1.104)$$

¹²⁵ Loss.

¹²⁶ Gradient Descent.

¹²⁷ İstatistik alanında “Mean Squared Error” olarak bilinir ve kısaca MSE biçiminde ifade edilmektedir.

Burada n , y_i ve $\hat{y}$ sırasıyla veri sayısı, gerçek (ya da deneysel) baz-değer ve tahmin değeridir. Şekil 1.26’da bu yöntemi temsil eden bir örnek sunulmuştur.

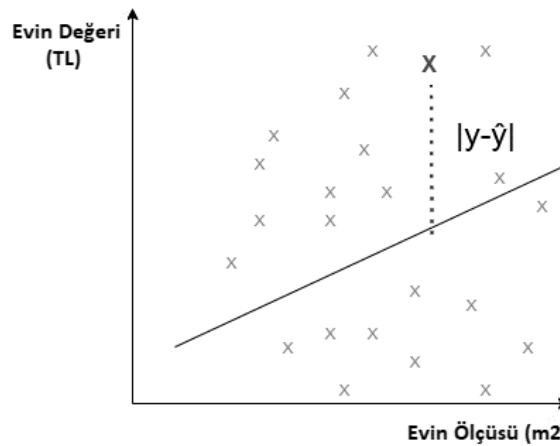


Şekil 1.26. MSE hesabı için örnek bir grafik (Bajaj, 2023).

Ek olarak, MSE metriği ile ilgili birkaç önemli noktadan daha bahsetmek yerinde olacaktır (Bajaj, 2023):

- Türevlenebilir olduğundan aslında daha iyi optimize edilebilir.
- Küçük hataları bile karesini alarak abartır, bu da esasen modelin ne kadar kötü olduğunun abartılmasına yol açar.
- Hata yorumlaması kare alma faktörü (ölçek) göz önünde bulundurularak yapılmalıdır. Örneğin, Mersin için ele alınan bir konut regresyon probleminde $MSE=33.05$ elde edilmiş olsun. Burada hata payı öncelikle (fiyat)²-eksenine karşılık gelecektir.
- Kareleme faktörü sebebiyle temel olarak diğer metriklerle göre aykırı değerlere daha yatkındır.
 - o Ortalama Mutlak Hata¹²⁸:

Basitçe ifade etmek gerekirse “gerçek dünya” değerleri ile tahmin edilen veri sonuçları arasındaki farkın ortalamasıdır (genel betimleme için bakınız Şekil 1.27).



Şekil 1.27. MAE hesabı için örnek bir grafik (Bajaj, 2023).

¹²⁸ Alanda “Mean Absolute Error” olarak bilinir ve MAE biçiminde kısaltılmaktadır.

Bu yaklaşım matematiksel olarak şu şekilde betimlenmektedir:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}|. \quad (1.105)$$

Yukarıdaki denklemde yazılan n , y_i ve $\hat{y}$ terimleri sırasıyla toplam veri sayısı, gerçek değer ve tahmini değere karşılık gelmektedir. MAE hesabında öne çıkan anahtar özellikleri şöyle sıralamak mümkündür (Bajaj, 2023):

- Hataları abartmadığı için aykırı değerlere karşı MSE'ye göre farklı davranır.
- Esasen bize tahminlerin gerçek çıktıdan ne kadar uzak olduğuna dair bir ölçüt vermektedir. Bununla birlikte, MAE değerler arasındaki farkın mutlak değerini kullandığından hatanın yönü¹²⁹ hakkında bir fikir vermez.
- MAE türevlenebilen MSE'nin aksine türevlenemez.
 - o Karekök Ortalama Hata¹³⁰:

Bu metrik hedef değer ile regresyon tipi tahmin yaklaşımı tarafından verilen değer arasındaki kare farkının ortalamasının kareköküne karşılık gelmektedir. Matematiksel olarak şu şekilde tanımlanır:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y})^2}. \quad (1.106)$$

Burada vurgulanması gereken birkaç önemli nokta bulunmaktadır (Bajaj, 2023):

- MSE'nin türevlenebilirlik özelliğini korur.
- MSE tarafından yapılan daha küçük hataların abartılması durumunu karekök alarak tartışır.
- Ölçek artık rasgele değişkenle aynı mertebede olduğundan, hata yorumlaması sorunsuz bir şekilde yapılabilir.
- Ölçek faktörleri temelde normalleştirildiğinden aykırı değerler söz konusu olduğunda sorun çıkarmaya daha az eğilimlidir.
 - o R-kare Değeri:

Bu metrik hakkında LSTM algoritmasının tanıtıldığı kısımda yeterince ayrıntılı bilgi sunulmuştur.

- o Düzeltilmiş R-kare Değeri¹³¹:

Vanilya R-kare¹³² metriğinin yanıltıcı¹³³ bir huyu vardır ve araştırmacıyı puan arttıkça modelin geliştiğine inandırır. Bu yanıltıcı durum bir model verilere gereğinden fazla uyduğu zaman meydana

¹²⁹ Yani verilerin değerleri eksik mi yoksa fazla mı tahmin edilmiştir.

¹³⁰ İstatistik alanında "Root Mean Squared Error" olarak bilinir ve kısaca RMSE biçiminde ifade edilmektedir.

¹³¹ Terminolojide "Adjusted R-squared" olarak bilinmektedir.

¹³² Literatürde "Vanilla R-squared" adıyla bilinmektedir.

¹³³ Araştırmacıyı kandıran ya da şeytanca.

gelmektedir ve elde edilen başarı oranı %100 olur, ancak aslında öğrenme gerçekleşmemiştir. Bu olumsuzluğun önüne geçmek için R -kare bağımsız değişkenlerin sayısı ile yeniden ayarlanır. Düzeltilmiş R -kare değeri olarak anılan bu yaklaşım yalnızca gerçek bir iyileşme varsa etkisini gösterdiğinden verdiği değer daima R -kare değerinden düşük olur ve matematiksel olarak şöyle betimlenmektedir (Bajaj, 2023):

$$R_D^2 = 1 - \frac{n-1}{n-k-1} (1 - R^2). \quad (1.107)$$

Buradaki n ve k terimleri sırasıyla gözlem sayısı ve bağımsız değişken sayısı olarak bilinmektedir.

- **Sınıflandırma metrikleri**

Temel hedefin verilerin sınıflandırılması olduğu problemler dünyanın önde gelen AI araştırmaları alanlarındandır. Kullanım durumlarını vurgulamak gerekirse neredeyse tüm üretim ve endüstriyel ortamlarda mevcuttur diyebiliriz: konuşma tanıma, yüz tanıma, el yazısı tanıma, metin sınıflandırma bazı önemli örneklerdir ve liste kolayca uzatılabilmektedir. Sınıflandırma modellerinin ayırık çıktıları olur, bu nedenle ayırık örneklem gruplarını bir yöntem uygulayarak karşılaştıran bir metriğe ihtiyaç duyulmaktadır. Bu metrikler bir algoritmanın performansını değerlendirerek onun ne kadar iyi/başarılı/tutarlı (veya ne kadar kötü/başarısız/tutarsız) olduğunu söyler, ancak her bir metrik bu işi farklı bir yöntemle yapmaktadır.

- **Karışıklık Matrisi ve F1-puanı:**

Bu kavramlar hakkında bazı önemli bilgiler daha önce kNN modelinin tanıtıldığı kısımda verilmiştir, ancak bu kısımda paylaşacağımız bazı diğer metriklerin tanımında ihtiyaç duyulacağından karışıklık matrisi daha kısa ama farklı formatta yeniden sunulmuştur. Karışıklık matrisi en sade biçimiyle Şekil 1.28'deki gibi betimlenir ve buradaki D , Y , P ve N harflerinin sırasıyla Doğru, Yanlış, Pozitif ve Negatif kelimelerini temsil eder.

		Tahmini Verileri	
		Pozitif (1)	Negatif (0)
Gerçek Verileri	Pozitif (1)	DP	YN
	Negatif (0)	YP	DN

Şekil 1.28. Karışıklık matrisinin sade bir temsili.

Hassasiyet, Özgüllük, Kesinlik, Tutarlılık ve F1-puanı kavramları karışık matrisi kullanılarak hesaplanmaktadır.

○ Jaccard endeksi:

Doğru tahmin kümesi ile gerçek değer kümesi kesişim kümesinin eleman sayısının birleşim kümesinin eleman sayısına oranı biçiminde tanımlanmaktadır. Bu nicelik de 0 ile 1 arasında değerler almaktadır ve tıpkı F1-puanında olduğu gibi 1 değeri en başarılı sonuç anlamına gelmektedir.

○ LogLoss değeri:

Tahminlerin olasılıkları üzerinden hesaplanmakta olup 0 ile 1 arasında değer almaktadır. Jaccard endeksi ve F1-puanı metriğinden farklı olarak LogLoss niceliğinin alacağı 0 değeri en başarılı durumu göstermektedir.

○ Doğru Pozitif Oran (DPO)¹³⁴:

Bu metrikte tıpkı F1-puanı gibi Şekil 1.28 sunulan karışıklık matrisi yardımıyla hesaplanmaktadır:

$$DPO = \frac{DP}{DP+YN} \quad (1.108)$$

Sezgisel olarak DPO/hatırlama¹³⁵ tüm pozitif veri noktalarının sayısının doğru şekilde pozitif kabul edilen pozitif veri noktalarının sayısına oranına karşılık gelmektedir. DPO değeri ne kadar yüksek olursa o kadar az oranda pozitif veri noktasını gözden kaçırmış oluruz.

○ Yanlış Pozitif Oran (YPO)¹³⁶:

Karışıklık matrisi yardımıyla bu metrik şöyle hesaplanır

$$YPO = \frac{YP}{YP+DN} \quad (1.109)$$

YPO/atık¹³⁷ sezgisel olarak yanlışlıkla pozitif kabul edilen negatif veri noktalarının sayısının tüm negatif veri noktalarının sayısına oranıdır. YPO değeri ne kadar yüksek olursa o kadar çok oranda negatif veri noktasını yanlış sınıflandırmışız demektir.

○ Alıcı İşletim Özellikleri Eğrisi Altındaki Alan¹³⁸:

Bir ROC eğrisi DPO ve YPO arasındaki ilişkiye göre çizilen bir grafiktir (bakınız Şekil 1.29). AUC değeri (altta kalan alan) 1'e ne kadar yakınsa model o oranda geliştirilmelidir.

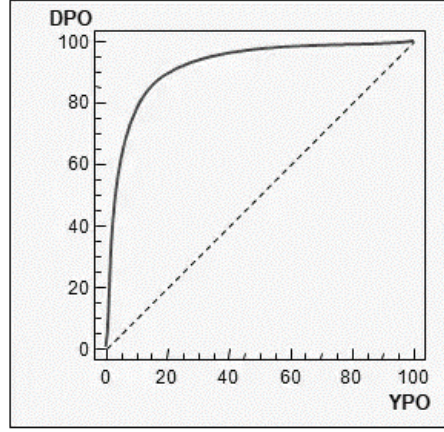
¹³⁴ True Positive Rate (TPR).

¹³⁵ Recall.

¹³⁶ "False Positive Rate" ya da kısaca "FPR"

¹³⁷ Fallout

¹³⁸ Literatürdeki karşılığı "Area Under Receiver Operating Characteristics Curve" biçimindedir ve kısaca "AUROC" olarak bilinmektedir.



Şekil 1.29. Örnek bir AUROC eğrisi (Singh, 2020).

2. KAYNAK ARAŞTIRMALARI

Telefonla pazarlamanın satış görevlilerinin yeni müşteriler edinmek, ürünlerini satmak ve hizmetlerini telefon üzerinden sağlamak için potansiyel müşterilerle iletişim kurduğu bir tür doğrudan pazarlama yöntemi (Kotler ve Keller, 2015) olduğunu vurgulamıştık. Sadece doğrudan pazarlama veri tabanları potansiyel müşteri bilgilerine ulaşmamıza yardımcı olduğundan bir şirketin bireysel özelliklerine veya alışveriş sırasındaki tutumlarına göre satışı ve/veya teklifi kabul etme olasılığı en yüksek olan potansiyel müşteri grubunu başarılı bir şekilde tahmin etmesi çok önemlidir (Mustapha ve Alsufyani, 2019). Son yıllarda popüler olmasıyla birlikte ML uygulamalarının tele-pazarlama dahil birçok alanda önemli ölçüde ilgi çektiği ve çeşitli araştırmalara konu olduğu görülmektedir. Bu tez çalışmasının temel amacı herhangi bir şirketin potansiyel müşterilerinin kararlarını öngörmek (yani olumlu ürün/hizmet satış oranını arttırmak) için ML algoritmalarını kullanmasının faydalı olup olmayacağını araştırmaktır. ML yaklaşımlarının tele-pazarlama problemlerine uygulanmasının son birkaç yılda dikkate değer ölçüde ön plana çıktığı görülmektedir, dolayısıyla bu tez çalışması kapsamında ele alınan konunun oldukça yeni ve özgün olduğunu vurgulamak yerinde olacaktır. Bu bölümünde literatürde öne çıkan ve tele-pazarlama performansı için ML yaklaşımlarını ele alan bazı güncel çalışmalar hakkında kısa ve önemli bilgiler sunulacaktır.

Çeşitli alanlarda birçok bilgi-keşif yaklaşımı geliştirilmiştir, ancak çok az model bilginin karar problemlerinde kullanımını açık bir şekilde ele almaktadır. Bu durumdan motive olan Ahn ve Ezawa *Bayesian Ağ Öğrenme*¹³⁹ modelinden çıkarılan bilgileri kullanarak gerçek zamanlı telefonla pazarlama işlemleri için bir karar destek sistemi sunmuştur (Ahn ve Ezawa, 1997). Daha spesifik açıklamak gerekirse araştırmalarını şöyle ifade etmek mümkündür: AT&T firmasının müşteri-iletişim çalışanları için bir telefon indirim planının tanıtımına ilişkin bir prototip karar destek sistemi geliştirilmiştir. İnşa edilen sistem temel olarak gerçek zamanlı karar desteği sağlamak için bir Bayesian ağ öğrenme mekanizması (bilgi keşfetme süreci) ile karar verme tekniğinin (etki diyagramı¹⁴⁰) birleştirilmesinden oluşmaktadır. Burada müşterinin önceki promosyon/yanıt geçmişinden faydalanarak yanıt verme olasılığını tahmin etmek için bir öğrenme yaklaşımı kullanılırken öngörülen olasılığı eylemlerle ilgili maliyet ve fayda ile entegre etmek için de etki diyagramı çerçevesinden faydalanılmıştır. Yazarlar öncelikle Bayesian ağ öğrenme modelinin sunduğu karar destek önerilerinin yanıltıcı olabileceğini göstermiş, sonrasında ise bu öğrenme modelinin etki diyagramları gibi titiz karar verme teknikleriyle ilişkilendirilmesinden sonra geliştirilen karar destek sisteminin akıllı bir karar tavsiyesi sağladığını ortaya koymuştur.

Kredi kartı dolandırıcılığı vakaları birçok ülkede ciddi zararlara neden olduğundan finansal kuruluşlar önleyici tedbirler almak adına *Dolandırıcılık Tespit Sistemleri*¹⁴¹ (DoTeS) kullanmaktadır.

¹³⁹ *Bayesian Network Learning*.

¹⁴⁰ *Influence Diagram*.

¹⁴¹ *Fraud Detection Systems (FDS)*.

Her ne kadar DoTeS dolandırıcılığı azaltmada iyi sonuçlar göstermiş olsa da böylesi sistemler tarafından işaretlenen vakaların önemli bir kısmı “yanlış tespitler” olarak çıkmaktadır, bu da önemli araştırma maliyetlerine ve kart sahibi rahatsızlığına neden olmaktadır. Bu bağlamda, Soheila önde gelen Kanada bankalarından biri tarafından sağlanan verileri analiz edilerek bir çeşit *nihai işleme sistemi* önermeyi (yani böylece mevcut DoTeS yaklaşımlarını geliştirmeyi) amaçlamıştır (Soheila, 2000). Özelliklerin varyasyonlarına, kombinasyonlarına ve eğitim sınıfı dağılımlarına dayanarak bu parametrelerin geliştirilen prototipin performansı üzerindeki etkisini ortaya çıkarmak amacıyla farklı deney setleri oluşturulmuştur. Sonuç olarak, önerilen öğrenme yaklaşımın ele alınan sistem üzerinde çok iyi bir iyileştirme potansiyeline sahip olduğu ortaya konulmuştur, ancak yazar ayrıca daha kapsamlı ve bilgilendirici verilerle zenginleştirilmesi gereken prototip sistemlerin geliştirilmesi de dahil olmak üzere daha fazla araştırmaya ihtiyaç olduğunu vurgulamıştır (Soheila, 2000).

Moro ve arkadaşları Portekiz’de 2008’den 2013’e kadar yaşanan mali kriz döneminde toplanan banka verilerini inceleyerek ilgili krizin etkileri de dahil olmak üzere ML araştırması yapmıştır (Moro ve ark., 2014). Bu araştırmalarının temel amacı uzun vadeli banka mevduatlarının satışına yönelik telefonla pazarlama çağrılarının başarısını tahmin etmek için bir veri madenciliği yaklaşımı önermektir. Bu bağlamda banka müşterisi, ürün ve sosyo-ekonomik niteliklerle ilgili 150 özellikten oluşan geniş bir veri dizisi analiz edilmiştir. Modelleme aşamasında Temmuz 2012’den önceki verilerle gerçekleştirilen ve 22 özellikten oluşan azaltılmış bir setin seçilmesine izin veren yarı otomatik bir özellik seçimi ele alınmış daha sonra dört farklı veri madenciliği modeli (LR, DTs, NNs ve SVMs) karşılaştırılmıştır.

Düşük maliyeti ve kullanıcı dostu olması nedeniyle tele-pazarlamanın bankaların uzun süredir mevduat elde etmek için tercih ettikleri yaygın bir yol haline geldiği aşikardır. Firmalar tele-pazarlama yöntemlerinin etkinliğini artırmak ve bu yolla mevduat çekmek için tüketici bilgilerini derinlemesine analiz etmek ister. Kim ve arkadaşları belirli bir müşterinin bir bankanın tele-pazarlama kampanyası için uygun olup olmadığını tahmin eden bir derin yapıli sinir ağı mimarisi önermiştir (Kim ve ark., 2015). Bu mekanizma oluşturulurken ayarlanması gereken katman sayısı, öğrenme hızı, düğümlerin başlangıç değeri ve diğer parametreler analiz edilmiş ve uygun mimari nihai sonuç olarak önerilmiştir. Yazarlar önerdikleri öğrenme modelini doğrulamak için 30 ay boyunca toplanan 45.211 telefon görüşmesinin banka pazarlama verilerini kullanmış ve diğer geleneksel sınıflandırıcılardan daha iyi performans (%76,7 doğruluk) elde etmiştir.

Bilindiği üzere dengesiz verilerin sınıflandırılması önemli bir sorun olup bir ML algoritmasının beklenen sonuca ulaşmasını dikkate değer ölçüde etkilemektedir. Dengesiz bir veri setiyle çalışmak çoğu zaman zorluk yaratmaktadır, çünkü her ne kadar azınlık sınıfları bazı durumlarda yararlı olsa da ML algoritmaları genellikle çoğunluk sınıflarına karşı pozitif ayırım yaparak azınlık sınıflarını göz ardı etme eğiliminde olur. Dolayısıyla makinemizden iyi sonuç almak istiyorsak elimizdeki verileri etkin kullanarak onu başarılı biçimde eğitmemiz gerekmektedir, böylece kullandığımız algoritma bilinmeyen bir sonuç için kendi başına doğru karar verebilecek hale gelecektir. Bu konuda Islam ve arkadaşları

*sentetik azınlık örnekleme tekniği*¹⁴² yardımıyla tutarlı sonuçlar üretebilen bir algoritmaya ulaşmışlardır (Islam ve ark., 2019).

Borugadda ve arkadaşları potansiyel banka müşterilerine uzun vadeli banka mevduatlarını teşvik etmek için telefonla pazarlama uygulamalarının benimsenmesine yönelik talebi araştırmak amacıyla farklı bir yaklaşım önermiştir (Borugadda ve ark., 2021).

Bankacılık sektöründeki telefonla pazarlama kampanyalarında potansiyel müşterilerin otomatik olarak başarılı biçimde seçilmesi için mevcut ML tabanlı yaklaşımlar kullanılırken heterojen verilerin işlenmesinde bu modellerin şeffaflık gösteremediklerini, optimum performansa ulaşamadıklarını ve minimum kaynak kullanma özelliğine sahip olmadıklarını gören Tékouabou ve arkadaşları *sınıf üyeliği tabanlı sınıflandırıcı*¹⁴³ adıyla bilinen yeni bir ML yaklaşımı inşa etmiştir (Tékouabou ve ark., 2022). Zira, yukarıda bahsi geçen kukla¹⁴⁴ değişkenler, modellerin nihai performansı üzerindeki etkilerini gerçekten değerlendirmeden çoğu çalışmada genellikle bastırılmakta veya keyfi bir şekilde kodlanmaktadır. Çoğu durumda ilgili kodlamalar ele alınan öğrenme modelinin performansını lehte etkileyerek aşırı uydurma (ya da ezberleme) durumunun yaşanmasına (yani modelin performansın düşmesine) yol açan gerçekliği gizlemektedir. Tékouabou ve arkadaşlarının elde ettiği sonuçlar sınıf üyeliği tabanlı sınıflandırıcı yaklaşımının önceki bakış açılarına oranla daha doğru tahminleri (%97,3 tutarlılık oranıyla) sunabildiğini göstermektedir (Tékouabou ve ark., 2022).

Chen ve arkadaşları gelişmekte olan çevrimiçi etkileşime dayalı ticari bir perspektiften bir firmanın “çevrimiçi davet - mağazada deneyim” sistemini ele alarak sesli sohbet yöntemiyle iletişim kurulan müşterilerin olası yanıtlarını öngörmeyi hedeflemiştir (Chen ve ark., 2023). Yazarlar yaptıkları literatür araştırmalarında benzer konuya yönelen önceki çalışmaların temel olarak müşterilerin beklentisi ile deneyimleri arasındaki tutarsızlıklara odaklanarak ortaya çıkarılabilen memnuniyetin müşterilerin satın alma niyetini çözmek için önemli bir faktör olduğunu gösterdiklerini vurgulamışlardır. Buradan hareketle *kara kutu derin öğrenme*¹⁴⁵ yöntemlerinin ampirik olarak karmaşık ses verileriyle başa çıkmada avantajlı çıktılar (örneğin sesli sohbete dahil edilen metin ve ses bilgileri) vaat ettiğini düşünerek araştırmalarını yapmaya motive olmuşlardır. Sonuç olarak, (i) profil bilgilerinin yokluğunda müşterilerin kişiselleştirilmiş ürün tercihlerinin ve dinamik memnuniyetlerinin öğrenilmesini, (ii) çoklu görüntülü sesli sohbete dayalı olarak müşterilerin gerçek deneyimlerinin modellenmesini ve (iii) kurama dayalı dinamik memnuniyetle sıradan bir kara kutu derin öğrenme modelinin müşteri yanıtının tahmin performansının geliştirilmesini sağlayan bir algoritması (Chen ve ark., 2023) önerdiler.

¹⁴² *Synthetic Minority Over-sampling Technique (SMOTE).*

¹⁴³ *Class membership-based classifier.*

¹⁴⁴ *Dummy.*

¹⁴⁵ *Black box deep learning.*

Xie çalışma arkadaşlarıyla birlikte *Rastgele Alt Uzak*¹⁴⁶, *Çoklu Güçlendirme*¹⁴⁷ ve *Rastgele Alt Uzak-Çoklu Yükseltme*¹⁴⁸ olarak bilinen ML yöntemlerini kullanarak banka telefonla pazarlama sonuçları için tahminler sunabilen bir performans modeli inşa ederek bankalara etkili bir pazarlama planı geliştirmek ve uygulamak için karar bilgileri sağlamak amacıyla bir yorumlanabilme analizi sunmuştur (Xie ve ark., 2023). Ayrıca, seçilen bağımsız değişkenlere dayalı olarak banka telefonla pazarlama tahmin modellerinin yeniden ele alınmasıyla pazarlama kampanyalarının başarısının nasıl etkilendiğini analiz etmek için *Kısmi Bağımlılık Grafikleri* yöntemiyle seçilen bağımsız değişkenlerin tüketicilerin mevcut abonelikleri üzerindeki marjinal etkilerini incelediler (Xie ve ark., 2023). Elde ettikleri ampirik sonuçlar seçilen bağımsız değişkenleri kullanan önerdikleri ML mimarisinin tahmin için en iyi performansı sunduğunu göstermiştir.

Koçoğlu ve Esnaf araştırmalarında sınıf etiketi dağılımının model başarısı üzerindeki etkisini incelemek için SMOTE kullandıktan sonra farklı ML algoritmaları (NBC, C5.0, Ekstrem Öğrenme Makinesi¹⁴⁹ ve DL) ile tele-pazarlamanın başarısını sınıflandırmaya yönelik bir model geliştirilmiştir (Koçoğlu ve Esnaf, 2022). Elde ettikleri önemli sonuçlar şöyledir: (i) DL ve DTs algoritmaları dikkate değer başarı oranı sunmuştur, (ii) veri seti dengeli olmadığında DL algoritması hassasiyet açısından daha iyi performans göstermiştir, (iii) tüm modeller arasında doğruluk, kesinlik ve F1-puanı açısından en iyi performans C5.0 algoritması ile elde edilmiştir.

Literatürden bazı dikkat çelici çalışmalar hakkında önemli bilgiler sunulmuştur. Bu örneklerin sayısı kolayca arttırılabilir, lakin sunulan çalışmalar *tele-pazarlama üzerine ML yaklaşımları* konusunun “gerçek dünya” problemleri arasındaki önemini göstermeye yeterlidir.

¹⁴⁶ *Random Subspace*.

¹⁴⁷ *Multi-Boosting*.

¹⁴⁸ *Random Subspace-Multi-Boosting*.

¹⁴⁹ *Extreme Learning Machine*.

3. MATERYAL VE METOT

Temeli tele-pazarlama tipleri üzerinde ML modellerinin etkisini araştırmak olduğundan bu tez çalışması istatistiksel hesaplamalar ve bilgisayar kodlamaları içermektedir. Tezdeki tüm yönetimin ana çatısını ise kısaca şöyle kodlamak mümkündür:

- Topla,
- Tasarla,
- Ayarla,
- Uygula!

3.1. Kodlama Materyalleri

Çalışmada herhangi bir somut materyal kullanılmamıştır. Fakat bazı hesaplamaları tamamlamak için Python Anaconda 3.7 adıyla bilinen bilgisayar kodlama programı ve bazı önemli kütüphaneleri kullanılmıştır:

- Pandas,
- Seaborn,
- Scikit,
- Numpy,
- Matplotlib.

Yukarıda listelenen kodlama kütüphanelerine ek olarak bazı kodlama paketleri de araştırmamızın istatistiksel analiz aşamasında önemli yer tutmuştur:

- SMOTE,
- DataCleaning,
- LabelEncoder,
- Standardization.

3.2. Öğrenme Metodolojisi

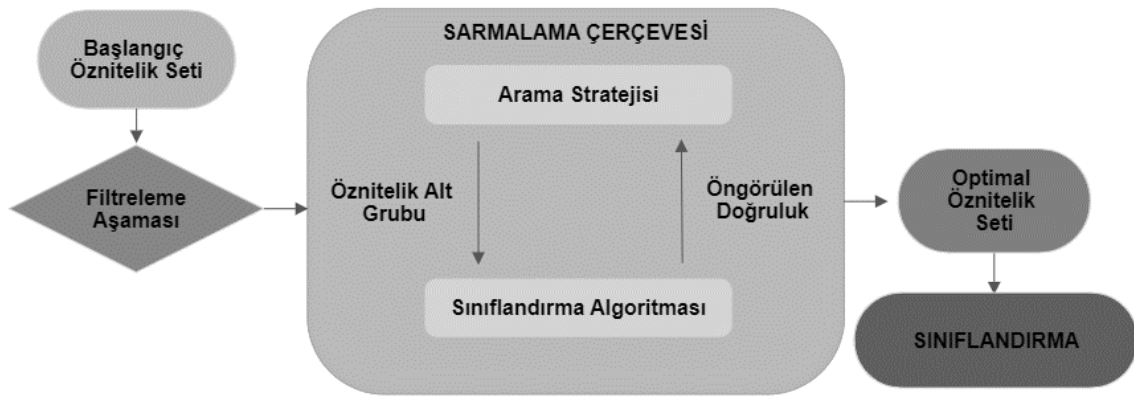
Artan kampanya maliyetleri ve düşük pazarlama yanıt oranları, şirketleri başarılı satış istatistikleri elde etmek için müşteri davranışını modellemeye yönlendirmekte, ancak bu gereklilik onları genellikle karmaşık yöntemlerle çalışmaya zorlamaktadır. Bir müşterinin pazarlamaya yönelik olası tepkisini tahmin etmenin, kampanyadan önce bir şirkete önemli avantaj sağlayabileceği bilinmektedir. Araştırmamız birbirini takip eden altı aşamadan oluşan *Veri Madenciliği için Endüstriler Arası Standart Süreç*¹⁵⁰ olarak bilinen metodolojiye uygun olarak yürütülmüştür:

- i) Sorunu anlamak: İşletmenin ihtiyacı nedir?
- ii) Verilerin deşifre edilmesi: Hangi verilere sahibiz? Neye ihtiyacımız var?

¹⁵⁰ Bu metodoloji literatürde “Cross Industry Standard Process for Data Mining” olarak bilinmektedir ve kısaca “CRISP-DM” biçiminde betimlenir.

- iii) Veri hazırlama: Veri kümesi temiz mi? Modellemeyen önce onu nasıl organize ederiz?
- iv) Uygun tekniklerin geliştirilmesi: Hangi modelleri uygulayabiliriz?
- v) Yaklaşımın performansının değerlendirilmesi: Hangi algoritma talebi en iyi şekilde karşılar?
- vi) Uygulama: Paydaşlar hedeflere nasıl ulaşıyor?

Tez konusu kapsamındaki araştırmaların ML kısmının temelini başarılı sınıflandırma mekanizmalarından olan kNN algoritması oluşturmaktadır. Bu algorithmada denetimli öznelik seçimi yöntemlerinden olan sarıcı¹⁵¹ (ya da sarmalayıcı) yaklaşımı tercih edilmiş olup inşa edilecek kNN mimarisinin iş akışı Şekil 1.30'da şematik olarak gösterilmektedir.



Şekil 3.1. Sarıcı kNN mekanizmasının genel akış şeması.

Sarmalayıcı tipi denetimli öznelik seçimi yaklaşımları ele alınan veri setindeki özelliklerin doğasını genel veri özelliklerine dayalı yöntemler kullanarak değerlendirmektedirler. Şekil 1.3'te önemli sarmalayıcı yaklaşımları vurgulanmakta olup tez çalışmamızda aşağıda listelenen sıralı seçim modelleri¹⁵² kullanılacaktır

- Sıralı Geri-yönde Seçim¹⁵³,
- Sıralı Geri-yönde Kayan Seçim¹⁵⁴,
- Sıralı İleri-yönde Seçim¹⁵⁵,
- Sıralı İleri-yönde Kayan Seçim¹⁵⁶.

¹⁵¹ Wrapper.

¹⁵² Bu modellerin isimleri tez kapsamında kısaltılmış olarak kullanılacaktır ve literatür ile uyumlu olmak adına terminolojide kullanılan kısaltmaları tercih edilecektir.

¹⁵³ Sequential Backward Selection (SBS).

¹⁵⁴ Sequential Backward Floating Selection (SBFS).

¹⁵⁵ Sequential Forward Selection (SFS).

¹⁵⁶ Sequential Forward Floating Selection (SFFS).

Bilindiği üzere sıradan bir kNN yaklaşımının en kritik evrelerinden bir tanesi de iki veri noktası arasındaki mesafenin nasıl ölçüleceğidir. Literatürde en çok kullanılan mesafe ölçüm modelleri birinci bölümde ayrıntılı biçimde tanıtılmıştı. Araştırmamız kapsamında ele alınan veri setine en uygun ve literatürde çok iyi bilinen mesafe ölçüm modelleri Tablo 3.1’de sunulmuştur.

Tablo 3.1. Tez kapsamında kullanılan mesafe ölçüm yöntemleri.

Model	Tanım
CityBlock	$\sum_{i=1}^k x_i - y_i $
Chebyshev	$\max_i x_i - y_i $
Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Correlation	$\frac{1}{2} \left(1 - \frac{\sum_{i=1}^k (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^k (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^k (y_i - \bar{y})^2}} \right)$
Canberra	$\sum_{i=1}^k \frac{ x_i - y_i }{ x_i + y_i }$
Dice	$1 - \frac{2 \sum_{i=1}^k x_i y_i}{\sum_{i=1}^k x_i^2 + \sum_{i=1}^k y_i^2}$
Cosine	$1 - \frac{\sum_{i=1}^k x_i y_i}{\sqrt{\sum_{i=1}^k x_i^2} \sqrt{\sum_{i=1}^k y_i^2}}$

Herhangi bir veri kümesindeki dengesiz dağılımları yeniden düzenleyerek dengeli hale getirmek için algoritmanın veri seti özelliklerine bakarak yeni sentetik veriler üretmesi gerekmektedir. Bu işlem inşa edilen ML algoritmasının azınlık verilerini göz ardı etmesini engelleyerek daha tutarlı ve faydalı tahminler sunmasını sağlamaktadır. Tez kapsamında tercih edilen yöntem SMOTE adıyla bilinen ve literatürde en çok tercih edilen yaklaşımdır.

3.3. Veri Kümesi

Çalışmamızda kullanılan veri seti Google tarafından desteklenen Kaggle platformundan temin edilmiş (Mohamed, 2022) olup 12 öznitelik sütunu ve 100.000 satırlık müşteri bilgileri içermektedir. İlgili veri grubunu betimleyen tablonun ilk 100 satırı örnek olması amacıyla Tablo 3.2, Tablo 3.3 ve Tablo 3.4’te parçalar halinde sunulmaktadır. Bu veri grubu yurt dışı kaynaklı anonim bir firmaya ait olduğundan yabancı dilde ifade edilen öznitelikler bulundurmaktadır: “Call ID (Telefon kimliği)”, “Sales (Satışlar)”, “Agent ID (Acente kimliği)”, “Age (Yaş)”, “Product ID (Ürün kimliği)”, “Time Zone (Zaman bölgesi)”, “Phone Code (Telefon kodu)”, “First Name (İlk Ad)”, “Last Name (Soyadı)”, “Area Code (Alan kodu)”, “Gender (Cinsiyet)” ve “Call Count (Aranma sayısı)”. Tez kapsamında elde edilen sonuçların kullanılan veri setiyle uyumlu olması adına öznitelik adları araştırmamız boyunca değiştirilmeden özüne bağlı kalınarak kullanılmıştır¹⁵⁷. Diğer yandan “Sales (Satışlar)” özneliğinin

¹⁵⁷ Bir diğer önemli neden de tez kapsamında elde edilen sonuçların bilim dünyasıyla paylaşılacak olmasıdır.

“True (Olumlu)” ve “False (Olumsuz)” alt seçeneklerini, “Gender (Cinsiyet) özniteliğinin ise “Male (Erkek)”, “Female (Kadın)” ve “Non-binary (İkili-olmayan)” alt gruplarını içerdiği görülmektedir.

Tablo 3.2. Seçilen telefonla pazarlama veri havuzundan örnekler (1. Kısım).

Call ID	Sales	Agent ID	Age	Product ID	Time Zone	Phone Code	First Name	Last Name	Area Code	Gender	Call Count
9545434	False	5265	42	147	2	37	Jk	Jk	2302	Male	1
9211206	False	5226	74	146	2	37	Em	Sh	1501	Male	10
8873010	False	4452	35	144	2	37	BI	MI	1550	Male	9
9852034	False	5461	40	149	2	37	WT	LI	1401	Male	6
9416548	False	5298	26	147	2	37	LA	LA	125	Female	12
10189322	False	5139	33	150	2	37	Me	Is	4091	Female	1
10277850	False	4828	33	151	2	37	So	Ts	3880	Female	3
9105514	False	5292	70	145	2	37	Dy	Ma	432	Female	9
8663012	False	5044	41	143	2	37	No	Mi	4360	Female	9
10216124	True	4912	72	150	2	37	TA	MS	1983	Male	1
9857654	False	4912	34	149	2	37	Dn	Gi	6330	Female	1
9444378	False	5265	73	149	2	37	Pa	On	81	Female	1
9079430	False	5230	30	145	2	37	Rn	Hs	7580	Female	3
9899362	True	5002	53	149	2	37	Le	Ke	7100	Male	1
9302198	False	5053	67	146	2	37	My	No	1827	Female	7
10271460	False	4452	79	151	2	37	TO	MO	1133	Male	1
9222054	False	5312	63	146	2	37	AM	PO	7100	Male	12
8927484	False	5214	33	144	2	37	Rn	Mh	4092	Male	7
9190940	False	4926	39	145	2	37	NL	NL	60	Male	1
10237446	False	5425	25	150	2	37	Jr	Cm	2021	Male	1
9315364	False	5077	42	146	2	37	NO	HA	81	Male	1
9538208	False	5357	70	147	2	37	Hk	Hk	4068	Female	1
10305716	False	4722	70	151	2	37	PO	AA	1818	Male	1
8966976	False	5053	49	144	2	37	Fm	Jn	7798	Male	8
9385676	False	5298	69	147	2	37	IN	IN	3201	Male	10
9139012	False	5076	55	145	2	37	Fy	Dk	6850	Female	11
10234552	False	5076	73	150	2	37	Pm	Hj	4092	Male	2
9974496	False	4926	74	149	2	37	Nl	Dn	4451	Male	1
9739880	False	4994	56	148	2	37	Ri	Mi	1818	Female	1
8866238	False	5051	59	144	2	37	Cd	Gr	2571	Male	9
9849650	True	5472	58	149	2	37	Ma	Mo	718	Male	2
9204606	False	5296	43	146	2	37	Ua	Ll	1609	Female	7
8822370	False	5207	68	144	2	37	Bi	Mn	7784	Male	9
9750820	False	5386	45	148	2	37	Hn	am	37	Male	3
9283212	False	5209	39	146	2	37	JI	HE	3242	Male	2
8868614	False	5208	56	144	2	37	Ia	Me	8	Female	6
8507636	False	4866	62	142	2	37	Em	Vi	8925	Male	9
9366710	False	5077	29	146	2	37	Gl	As	1	Female	2
9517532	False	-	42	147	2	37	Jz	Jz	1947	Male	1
9391180	False	5297	47	147	2	37	Ja	Ja	4450	Female	1
9226824	False	5208	42	146	2	37	Je	Po	699	Female	3
9233896	False	4995	46	146	2	37	Ls	Ba	7130	Male	1
9327068	False	5053	56	146	2	37	by	li	4309	Female	2
8544466	False	4866	65	142	2	37	Id	Mi	1818	Female	7
9325652	False	5051	38	146	2	37	JS	DI	9701	Male	1
9202788	False	5313	65	146	2	37	Wr	Ne	1389	Male	8
9758378	False	4958	54	148	2	37	Js	ee	7395	Male	2
9114476	False	5073	33	145	2	37	Hk	Th	6520	Male	2
9181376	False	5028	27	145	2	37	Sa	Gs	7380	Male	5
9206096	True	4804	64	146	2	37	MS	ZE	175	Male	2
9521206	False	5341	71	147	2	37	Dr	Dr	1401	Female	7
9744682	False	4876	57	148	2	37	Ms	On	4051	Male	1

Tablo 3.3. Seçilen telefonla pazarlama örneği veri havuzundan örnekler (2. Kısım).

Call ID	Sales	Agent ID	Age	Product ID	Time Zone	Phone Code	First Name	Last Name	Area Code	Gender	Call Count
9272158	False	-	50	146	2	37	KS	LD	181	Male	3
9744686	False	5382	30	148	2	37	Mi	Mg	9701	Female	2
9955872	False	5405	48	149	2	37	AL	MI	1035	Male	4
8826600	False	5146	55	144	2	37	SA	AS	7785	Female	5
9913526	False	5298	78	149	2	37	Or	Ce	2350	Male	6
8764438	True	5181	35	143	2	37	Js	Me	6045	Male	9
9685618	False	4876	74	148	2	37	BN	JN	4037	Male	1
8864940	False	5077	69	144	2	37	BI	MA	4310	Male	9
9894366	False	4702	42	149	2	37	BE	SA	1809	Female	2
9998072	False	5272	54	149	2	37	Js	Gg	1448	Male	1
9536292	False	5312	49	147	2	37	Ni	Ni	4068	Female	1
9218848	False	5209	45	146	2	37	La	Hk	1020	Female	4
9047402	False	5230	63	145	2	37	Ma	Vy	6705	Female	1
10228666	False	5298	62	150	2	37	Co	Oo	9950	Male	1
9970394	False	4894	78	149	2	37	Aa	Vn	185	Female	1
10303180	True	5065	60	151	2	37	PA	MI	2351	Male	3
9797466	False	5403	62	148	2	37	Ia	Sn	1450	Female	8
10018630	False	5459	32	149	2	37	Da	Ni	4001	Female	5
10211246	False	5424	71	150	2	37	Rd	Ls	6530	Male	1
9112510	False	5296	31	145	2	37	GE	RG	8240	Female	3
9452174	False	4971	66	147	2	37	TA	TA	2195	Male	3
9319274	False	5076	35	146	2	37	Dm	My	4001	Female	2
8891970	False	4677	32	144	2	37	JN	BR	2571	Male	6
9973080	False	5265	36	149	2	37	We	Ah	4092	Male	2
9539874	False	4748	70	147	2	37	Jn	Jn	2430	Male	1
8746028	False	4899	57	143	2	37	Ae	Vn	7130	Male	4
9967484	False	5146	76	149	2	37	Ze	Ge	8000	Female	2
9360256	False	5077	34	146	2	37	Oa	Mi	1459	Female	1
9395328	False	5296	38	147	2	37	NA	NA	2001	Female	2
9701364	False	5296	78	148	2	37	An	Na	2774	Male	3
9426978	False	5181	47	147	2	37	ND	BE	300	Male	5
8703544	False	5035	76	143	2	37	Ty	Cy	4159	Female	7
9368260	True	5226	65	146	2	37	No	Ze	7100	Female	5
9715118	False	4926	73	148	2	37	Pe	mr	1449	Male	1
9178868	False	4958	77	145	2	37	NL	NL	1182	Male	2
9161504	False	4926	43	145	2	37	Se	Ka	4068	Female	3
9771870	False	5460	28	148	2	37	Ll	Lz	2196	Male	15
9552914	False	4452	38	147	2	37	Le	Le	2429	Male	1
9693962	False	4968	35	148	2	37	Vt	Ce	157	Male	1
10013608	True	4804	69	149	2	37	BY	FE	1245	Female	1
9829242	False	-	73	148	2	37	Ct	Ma	158	Female	6
8654756	False	4865	67	143	2	37	ME	NI	7784	Male	7
10261690	False	-	49	150	2	37	Km	Le	6070	Female	2
9695408	False	4958	77	148	2	37	Ze	VI	3340	Female	5
9075944	False	5065	58	145	2	37	Ji	Dj	4093	Female	2
8955626	False	4452	46	144	2	37	Nn	Ri	1754	Male	3
9465582	False	5252	75	147	2	37	Ma	Ma	157	Male	1
10200658	False	4876	41	150	2	37	Ee	Mn	1947	Male	2
94697006	False	4748	60	147	2	37	Ro	Ro	4068	Male	2
10213476	False	5399	46	150	2	37	De	Ln	35	Female	1
8901462	False	4677	55	144	2	37	Se	Vi	4001	Female	2
8871322	False	4971	42	144	2	37	na	ya	5454	Female	3
9319528	False	5077	60	146	2	37	Mn	VY	6001	Male	1
9832444	False	4899	66	148	2	37	Aa	Tg	17	Male	2
9107084	False	5077	36	145	2	37	Na	Gr	6070	Female	7
8951730	False	5061	55	144	2	37	To	Ma	152	Male	8

Tablo 3.4. Seçilen telefonla pazarlama örneği veri havuzundan örnekler (3. Kısım).

Call ID	Sales	Agent ID	Age	Product ID	Time Zone	Phone Code	First Name	Last Name	Area Code	Gender	Call Count
8869420	False	4452	36	144	2	37	Bd	Mo	4126	Male	9
9531588	False	5357	25	147	2	37	Me	Me	7370	Female	2
9370532	False	5065	67	146	2	37	WN	VE	3915	Male	1
10215476	False	4706	79	150	2	37	Ga	Pk	186	Female	3
9890938	False	-	63	149	2	37	a	Kk	2940	Female	2
10240088	False	4955	72	150	2	37	JS	JN	1459	Male	1
8732602	False	5220	32	143	2	37	Sl	Ss	7646	Male	9
9948988	False	5399	50	149	2	37	Ey	Re	7646	Male	1
9776284	False	4894	61	148	2	37	gm	bd	6070	Male	3
8692964	False	4971	69	143	2	37	Jn	Pe	7780	Male	6
9119128	False	5077	77	145	2	37	Bd	Kt	1935	Male	1
9910492	False	5341	63	149	2	37	Re	Ma	208	Female	2
9687244	False	4956	53	148	2	37	Ha	No	2091	Female	3
10182424	False	4899	31	150	2	37	Ta	Me	737	Male	2
9149520	False	5216	76	145	2	37	Re	Vk	1003	Female	2
9805536	False	4894	35	148	2	37	Rh	Hd	4092	Male	1
9215842	False	4958	75	146	2	37	FE	MO	1804	Female	1
9816756	False	5032	46	148	2	37	Ha	Bs	3936	Female	1
9936474	False	5386	75	149	2	37	Ma	Ba	2092	Male	4
9512890	False	5146	70	147	2	37	hy	hy	7100	Male	1

3.4. Analiz yöntemi

Araştırmamız kapsamında inşa edilen algoritmanın seçilen veri setine uygulanması beş ana basamaktan oluşmaktadır:

- Verilerin ön işleme,
 - Temizleme,
 - Dengeleme,
- Veri kümesinin görselleştirilerek gizli özelliklerinin ortaya çıkarılması,
- Serbest parametrelerinin ayarlanması (en uygun değerlerinin elde edilmesi),
- Uygun algoritma bileşenlerin belirlenmesi,
- İnşa edilen ML algoritmasının nihai formunun oluşturulması ve çalıştırılması.

Tez konusunda hedeflenen veri setinin analizinin başarılı biçimde yapılabilmesi amacıyla araştırma sürecinin başlarında çeşitli ML modelleri ve farklı alanlardan veri kümeleri çalışılmıştır. Tez araştırmaları kapsamında edinilen bilgiler ışığında tamamlanan bazı ML çalışmaları hedeflenen öğrenme analizinin nasıl yapılması gerektiğine yönelik bir yol haritası oluşmasını sağlamıştır. Bu ön çalışmalar bazı özgün ve önemli sonuçlara ulaşmamıza da olanak sağladığından İngilizce makale formatında hazırlanarak bazı uluslararası endeksli hakemli dergilere gönderilmiştir. Bu bağlamda iki tanesi farklı dergilerde basılmış bir tanesi ise tez yazım aşaması itibarıyla inceleme aşamasında olan üç farklı makale hazırlanmıştır¹⁵⁸. Araştırmamızın yöntemini daha açık ortaya koymamıza olanak sağlayacağına inandığımızdan bu çalışmaların kısa özetlerini sunmak yerinde olacaktır.

¹⁵⁸ Bu makaleler ile ilgili bilgiler tezin son sayfasında özgeçmiş başlığı altında sunulmuştur.

3.4.1. KMA¹⁵⁹ Işınım Sıcaklığına Derin Öğrenme Yaklaşımı

Astrofizik alanında KMA ışıınım sıcaklığına ilişkin yapılan gözlemsel verileri başarılı biçimde açıklayabilen kuramsal bir model henüz ortaya konamamıştır. Literatürde çeşitli kuramsal modellemeler bulunmakta olup bu modellerin başarılı olabilmesi için içerdikleri keyfi parametrelerin gözlemsel modellere göre dikkatlice ayarlanması gerekmektedir. 2022 yılında uluslararası *Annals of Physics* dergisinde basılan makalemizin ilk aşamasında KMA ışıınım sıcaklığı için kuramsal bir model inşa edilmiştir (Salti ve Kangal, 2022):

$$T(z) = T_0 \frac{nm(\Omega-2)+3(1-\Omega)}{nm\Omega(1+z)^{-\frac{3}{m}-n}+(2nm+3)(1-\Omega)} \left[1 - \Omega + \Omega(1+z)^{-\frac{3}{m}-n}\right]^{-(1+m)}. \quad (3.1)$$

Burada yazılan T_0 , n , m ve Ω terimleri ayarlanması gereken keyfi model parametreleridir. Ayrıca, T_0 niceliği KMA ışıınım sıcaklığının günümüzde ölçülen değerini temsil ederken z ise kozmik kızıla kayma parametresidir. Serbest parametrelerinin ayarlanması sonrasında inşa edilen kuramsal bir modelin güncel gözlemleri başarıyla açıklayıp açıklayamadığı ortaya konabilmektedir. Bu bağlamda KMS ışıınım sıcaklığı için güncel ölçüm sonuçları Tablo 3.5 ve Tablo 3.6’da parçalar halinde sunulmaktadır.

Tablo 3.5. KMA ışıınım sıcaklığı için astrofizik gözlem verileri (1. Kısım)

z	$T \pm \sigma$ (Kelvin)	Kaynak
0.000	2.726 ± 0.002	(Mather ve ark., 1999)
0.023	$2.79^{+0.080}_{-0.065}$	(Battistelli ve ark., 2002)
0.037	2.89 ± 0.04	(Hurier ve ark., 2014)
0.053	2.88 ± 0.19	(Lopez-Corredoira ve ark., 2017)
0.072	2.93 ± 0.02	(Hurier ve ark., 2014)
0.115	$3.18^{+2.40}_{-2.05}$	(Lopez-Corredoira ve ark., 2017)
0.119	3.19 ± 0.07	(Lopez-Corredoira ve ark., 2017)
0.125	3.06 ± 0.03	(Hurier ve ark., 2014)
0.152	2.90 ± 0.17	(Luzzi ve ark., 2009)
0.171	3.20 ± 0.03	(Hurier ve ark., 2014)
0.176	3.21 ± 0.14	(Lopez-Corredoira ve ark., 2017)
0.183	2.95 ± 0.27	(Luzzi ve ark., 2009)
0.200	2.74 ± 0.28	(Luzzi ve ark., 2009)
0.202	3.36 ± 0.20	(Luzzi ve ark., 2009)
0.203	3.34 ± 0.10	(Battistelli ve ark., 2002)
0.216	3.85 ± 0.64	(Luzzi ve ark., 2009)
0.220	3.29 ± 0.03	(Hurier ve ark., 2014)
0.232	3.51 ± 0.25	(Luzzi ve ark., 2009)
0.252	3.39 ± 0.26	(Luzzi ve ark., 2009)
0.273	3.42 ± 0.04	(Hurier ve ark., 2014)
0.282	3.22 ± 0.26	(Luzzi ve ark., 2009)
0.291	4.05 ± 0.66	(Luzzi ve ark., 2009)
0.322	3.56 ± 0.05	(Hurier ve ark., 2014)
0.377	3.72 ± 0.06	(Hurier ve ark., 2014)

¹⁵⁹ “Kozmik Mikrodalga Artalan” anlamına gelmekte olup literatürde “Cosmic Microwave Background” olarak bilinmektedir ve “CMB” biçiminde kısaltılmaktadır.

Tablo 3.6. KMA ışıının sıcaklığı için astrofizik gözlem verileri (2. Kısım)

z	$T \pm \sigma$ (Kelvin)	Kaynak
0.399	$3.58^{+0.49}_{-0.39}$	(Lopez-Corredoira ve ark., 2017)
0.428	3.97 ± 0.07	(Hurier ve ark., 2014)
0.451	3.98 ± 0.19	(Luzzi ve ark., 2009)
0.471	3.94 ± 0.11	(Hurier ve ark., 2014)
0.525	4.38 ± 0.12	(Hurier ve ark., 2014)
0.546	3.69 ± 0.37	(Luzzi ve ark., 2009)
0.550	3.97 ± 0.19	(Luzzi ve ark., 2009)
0.565	4.38 ± 0.16	(Hurier ve ark., 2014)
0.572	$4.83^{+3.03}_{-1.62}$	(Lopez-Corredoira ve ark., 2017)
0.579	$4.01^{+3.88}_{-2.42}$	(Lopez-Corredoira ve ark., 2017)
0.583	$3.66^{+0.59}_{-0.43}$	(Lopez-Corredoira ve ark., 2017)
0.618	4.40 ± 0.19	(Hurier ve ark., 2014)
0.676	4.78 ± 0.28	(Hurier ve ark., 2014)
0.718	4.93 ± 0.37	(Hurier ve ark., 2014)
0.777	4.52 ± 0.62	(Hurier ve ark., 2014)
0.870	5.36 ± 0.62	(Hurier ve ark., 2014)
0.890	5.08 ± 0.10	(Muller ve ark., 2013)
0.972	5.81 ± 1.03	(Hurier ve ark., 2014)
1.675	$8.54^{+0.65}_{-0.56}$	(Roth ve Bauer, 1999)
1.729	7.50 ± 1.40	(Noterdaeme ve ark., 2011)
1.774	7.80 ± 0.60	(Noterdaeme ve ark., 2011)
1.776	7.40 ± 0.80	(Songaila ve ark., 1994a)
1.973	7.90 ± 1.00	(Ge ve ark., 1997)
2.038	8.60 ± 1.00	(Noterdaeme ve ark., 2011)
2.337	10.0 ± 4.00	(Srianand ve ark., 2000)
2.418	9.15 ± 0.70	(Srianand ve ark., 2008)
2.689	10.5 ± 0.70	(Noterdaeme ve ark., 2010)
2.811	10.4	(Lu ve ark., 1996a)
2.844	10.5	(Lu ve ark., 1996a)
2.909	10.7	(Songaila ve ark., 1994b)
3.025	$12.1^{+1.70}_{-3.20}$	(Molaro ve ark., 2002)
4.080	13.9	(Lu ve ark., 1996a)
4.383	14.7	(Lu ve ark., 1996b)

Yukarıdaki verilerden bazıları ikili hata payı içermektedir, diğer verilerle uyumlu hale getirmek adına

$\sigma = \sqrt{\frac{\sigma_{alt}^2 + \sigma_{üst}^2}{2}}$ biçiminde ifade edilen yaklaşımla ilgili veri noktaları için ortalama hata payı

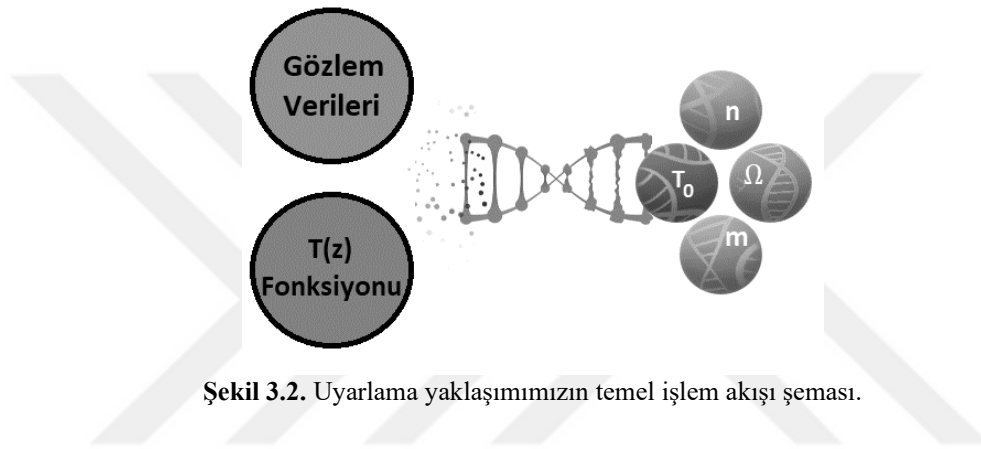
hesaplanmıştır: Bu yaklaşıma göre elde edilen sonuçlar ise şunlardır (Salti ve Kangal, 2022)

$$\sigma = \begin{cases} 0.073 & (z = 0.023 \text{ için}) \\ 2.232 & (z = 0.115 \text{ için}) \\ 0.443 & (z = 0.399 \text{ için}) \\ 2.429 & (z = 0.572 \text{ için}) \\ 3.234 & (z = 0.579 \text{ için}) \\ 0.516 & (z = 0.583 \text{ için}) \\ 0.607 & (z = 1.675 \text{ için}) \\ 2.562 & (z = 3.025 \text{ için}). \end{cases} \quad (3.2)$$

Diğer yandan bazı veri noktaları için de herhangi bir hata payı verilmediği görülmektedir, bu nedenle ilgili veri noktaları ile birlikte kullanılmak üzere diğer 52 veri noktası için hesaplanmış olan hata paylarının ortalaması alınmıştır (Salti ve Kangal, 2022)

$$\bar{\sigma} = 0.598. \quad (3.3)$$

Veri kümesinin düzenlenmesinden sonra modelin serbest parametrelerinin en uygun değerlerini belirleyebilmek adına Genetik Algoritma yaklaşımı kullanılmıştır. Bahsi geçen uyarılama¹⁶⁰ yaklaşımı temsili olarak genel hatlarıyla Şekil 3.2’de resmedilmiştir.



Şekil 3.2. Uyarılama yaklaşımımızın temel işlem akışı şeması.

Genetik algortmada kullanılmak üzere uyarılama fonksiyonu

$$\chi^2(T_0, \Omega, n, n) = \sum_i^{57} \frac{[T_{Gözlem}(z_i) - T_{Kuram}(z_i)]^2}{\sigma_i^2} \quad (3.4)$$

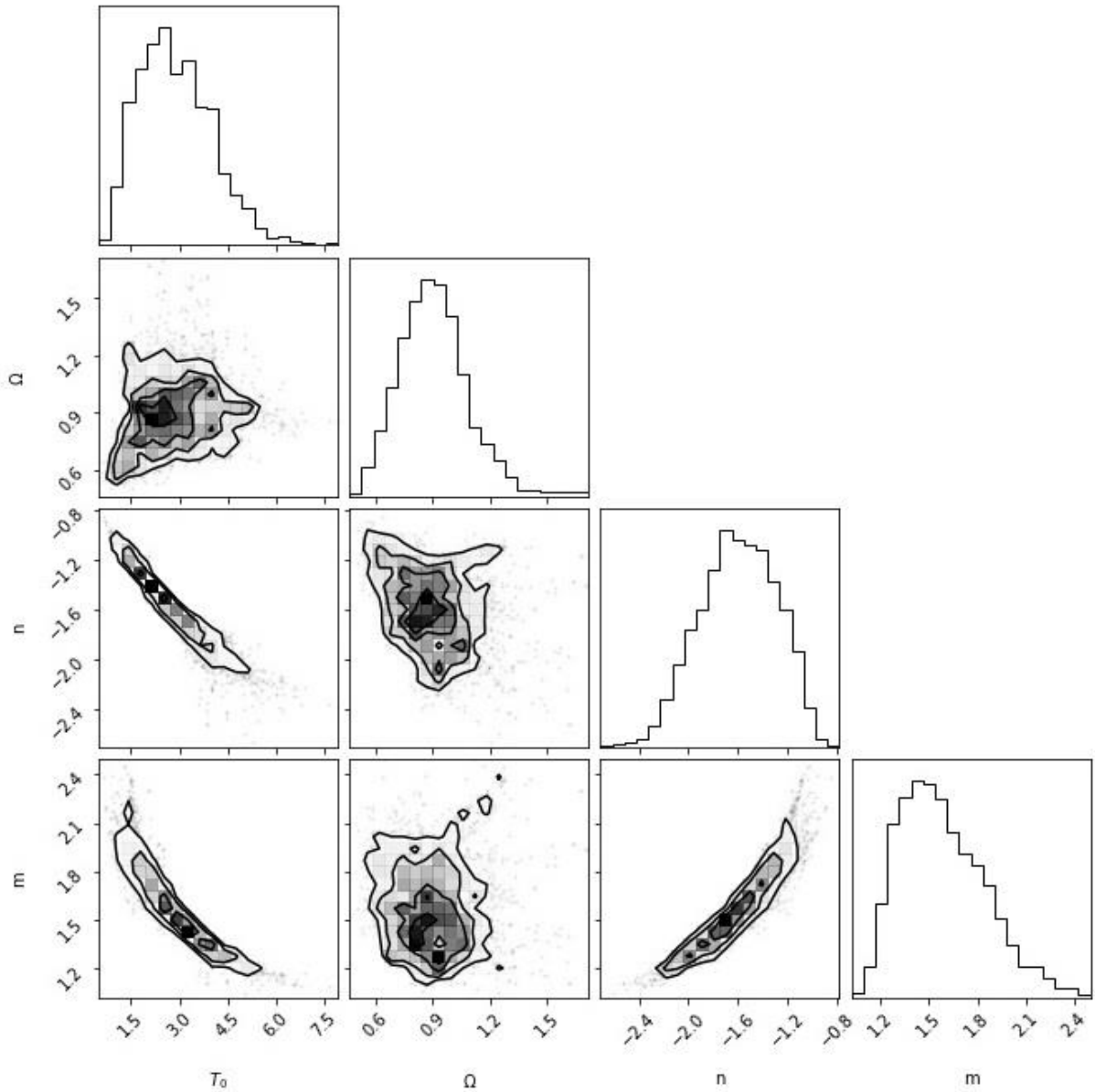
biçiminde seçilirken diğer keyfi model parametreleri için Tablo 3.7’de sunulan değerler kullanılmıştır (Salti ve Kangal, 2022).

Tablo 3.7. Genetik algortmada kullanılan hiper-parametre değerleri.

	Boyut	Popülasyon Büyüküğü	Mutasyon Oranı	Çaprazlama Oranı
Varsayım	4	100	0.05	0.95

Model parametrelerinin uyarlanması analizinde Şekil 3.3’te sunulan Gaussian dağılımlar ve parametre kombinasyonları için 1σ , 2σ ve 3σ güven bölgelerindeki iki boyutlu olasılık dağılımları elde edilmiştir (Salti ve Kangal, 2022).

¹⁶⁰ Fitting.



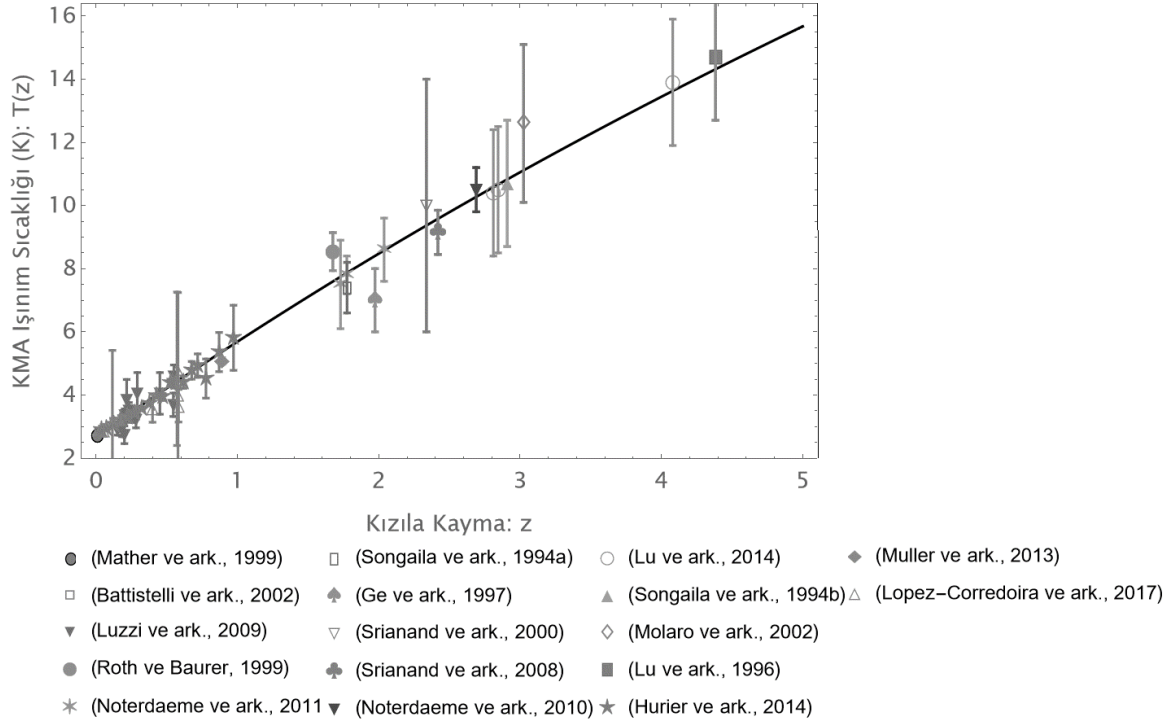
Şekil 3.3. KMA sıcaklık modeli parametrelerinin kombinasyonu için olasılık dağılımları.

Yukarıda paylaşılan grafiklerdeki orta noktalar en uygun parametre değerlerine işaret etmektedir, net değerler Tablo 3.8’de ayrıntılı olarak sunulmuştur (Salti ve Kangal, 2022).

Tablo 3.8. KMA modeli için keyfi parametrelerin en uygun değerleri.

	T_0	n	m	Ω
Değer	2.732	-1.586	1.566	0.892

Gözlemsel verilerle uyumlu hale getirilen kuramsal modelin davranışının deneysel verilerle karşılaştırıldığı analizin sonucu Şekil 3.4’te paylaşılmıştır (Salti ve Kangal, 2022).



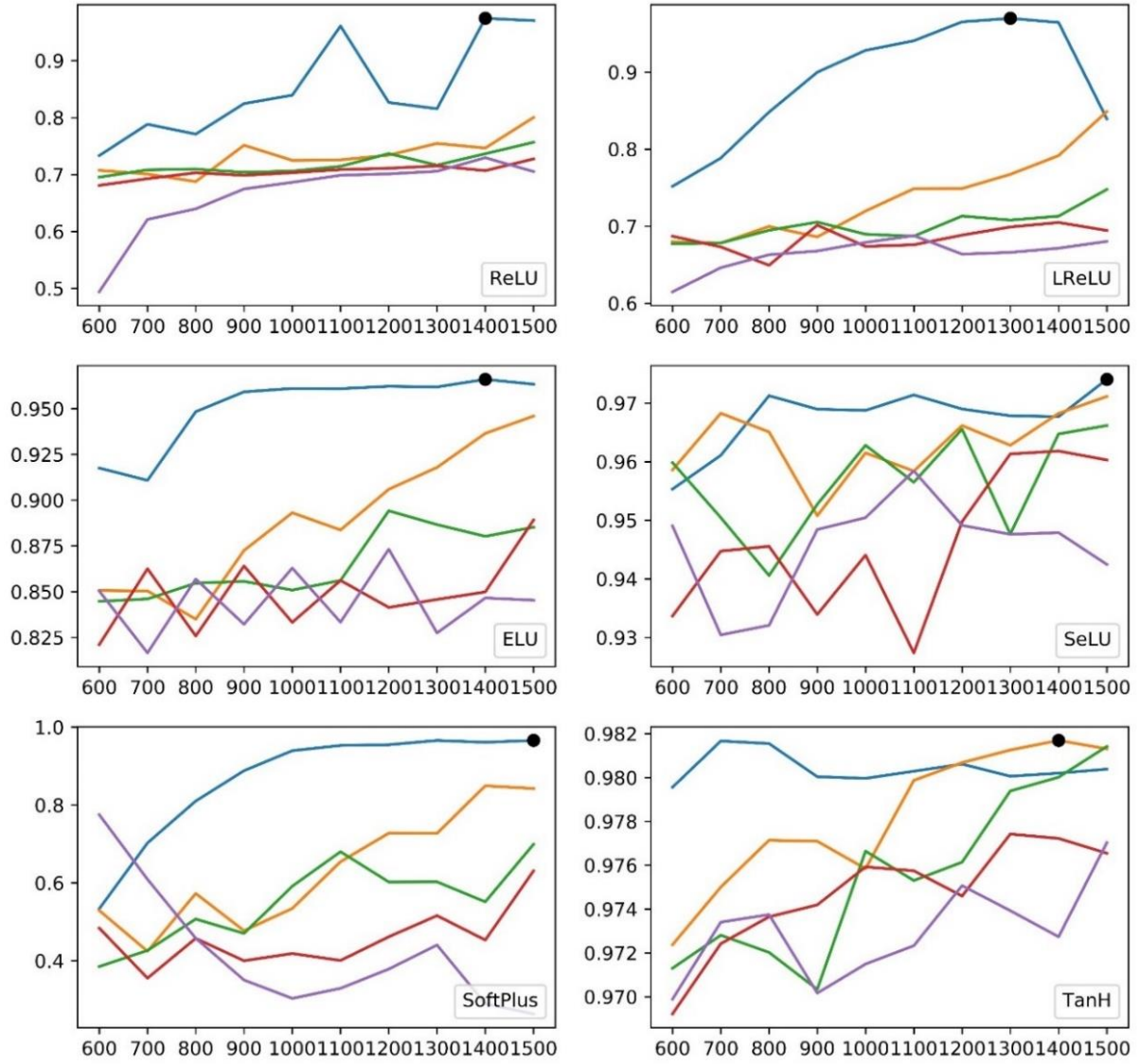
Şekil 3.4. KMA ışınlm sıcaklığının davranışı.

Tablo 3.8’den görüldüğü üzere Genetik ML yaklaşımı serbest model parametrelerinin değerlerini hesaplamamıza olanak sunmuş ancak bu değerler için hata paylarını verememiştir. Bu sorunu çözmek adına FIM yöntemi tercih edilmiş ve bu analizde kullanılmak üzere aşağıdaki bilgi matrisi tanımlanmıştır (Salti ve Kangal, 2022):

$$F_{\mu\nu} = \begin{pmatrix} \frac{\partial^2 \chi^2}{\partial T_0^2} & \frac{\partial^2 \chi^2}{\partial T_0 \partial \Omega} & \frac{\partial^2 \chi^2}{\partial T_0 \partial n} & \frac{\partial^2 \chi^2}{\partial T_0 \partial m} \\ \frac{\partial^2 \chi^2}{\partial \Omega \partial T_0} & \frac{\partial^2 \chi^2}{\partial \Omega^2} & \frac{\partial^2 \chi^2}{\partial \Omega \partial n} & \frac{\partial^2 \chi^2}{\partial \Omega \partial m} \\ \frac{\partial^2 \chi^2}{\partial n \partial T_0} & \frac{\partial^2 \chi^2}{\partial n \partial \Omega} & \frac{\partial^2 \chi^2}{\partial n^2} & \frac{\partial^2 \chi^2}{\partial n \partial m} \\ \frac{\partial^2 \chi^2}{\partial m \partial T_0} & \frac{\partial^2 \chi^2}{\partial m \partial \Omega} & \frac{\partial^2 \chi^2}{\partial m \partial n} & \frac{\partial^2 \chi^2}{\partial m^2} \end{pmatrix}. \quad (3.5)$$

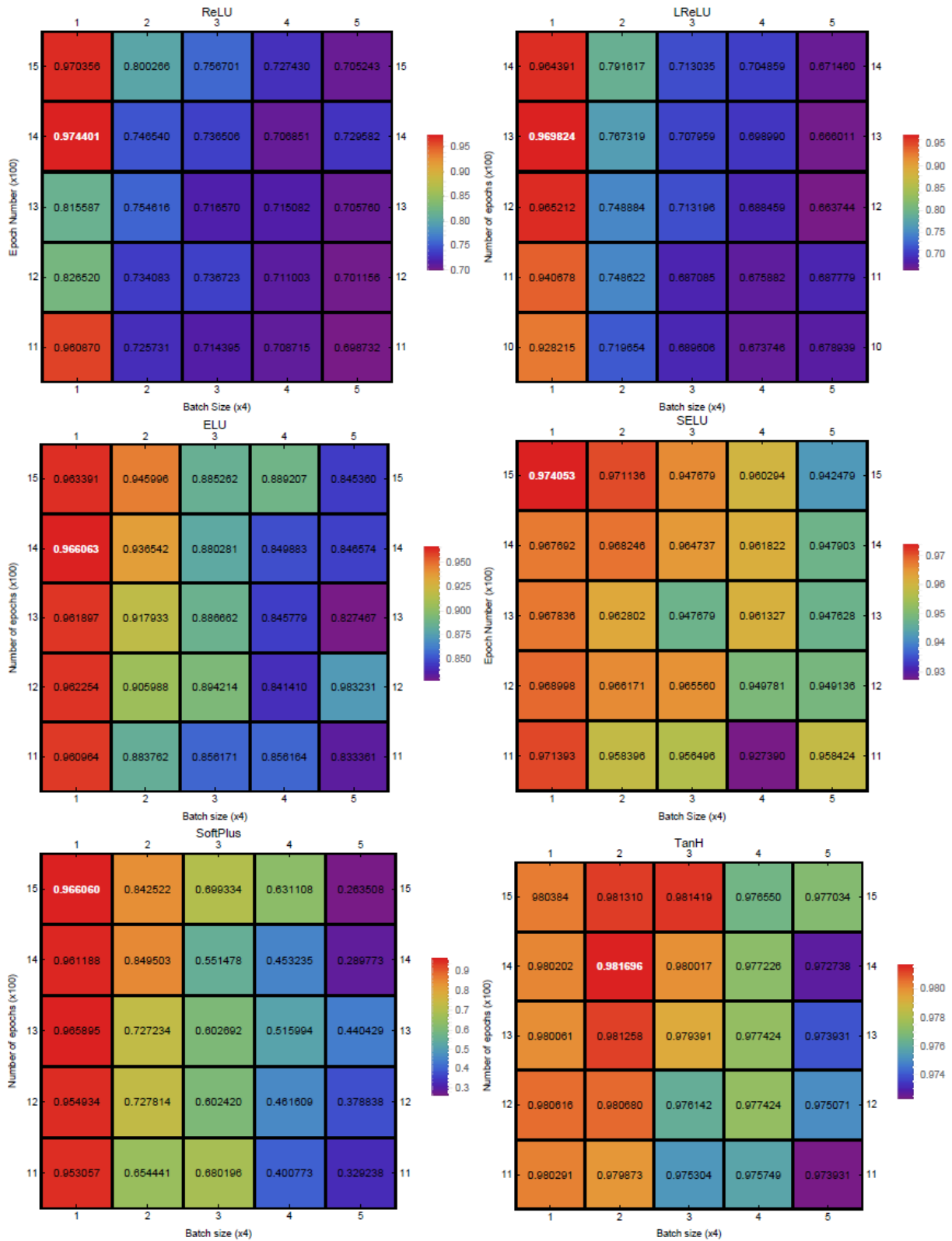
Sonuç olarak FIM analizi aşağıdaki değerlere ulaşmamızı sağlamıştır (Salti ve Kangal, 2022)

$$\theta_\mu \Rightarrow \begin{cases} T_0 = 2.732 \pm 0.015 \\ \Omega = 0.0892 \pm 0.002 \\ n = -1.586 \pm 0.003 \\ m = 1.566 \pm 0.002. \end{cases} \quad (3.6)$$



Şekil 3.5. LSTM-Bırakma algoritması test sürecinin R-kare analizi.

Genetik ML yaklaşımı aracılığıyla gözlemsel verilerle uyumlu hale getirilen kuramsal modelimizin gözlemi yapılmayan kızıla kayma değerleri için vereceği sonuçlara güvenilip güvenilemeyeceğini anlamanın yöntemi mevcut verilere yönelik üreteceği tahminlere bakmaktır. Bu bağlamda derin yapıli LSTM-Bırakma ve FIM yaklaşımları modelin tutarlılığını öğrenme algoritmaları perspektifinden sınamak için göz önüne alınmıştır. Birinci bölümde ifade edildiği üzere LSTM-Bırakma algoritması ayarlanması gereken bazı önemli hiper-parametreler ve seçilmesi gereken aktivasyon fonksiyonları içermektedir. Altı farklı aktivasyon fonksiyonu için LSTM-Bırakma ($p_b=0.5$ seçilmiştir) algoritması çalıştırılarak R-kare analizi yapılmıştır (bakınız Şekil 3.5) (Salti ve Kangal, 2022). Eğriler farklı *parti boyutu* değerlerini (mavi=4, turuncu=8, yeşil=12, kırmızı=16, mor=20) betimlerken yatay eksenler *dönem sayısı* değerlerini temsil etmektedir, ayrıca grafiklerde görülmeyen nümerik değerlere Şekil 3.6'dan ulaşmak mümkündür (Salti ve Kangal, 2022).



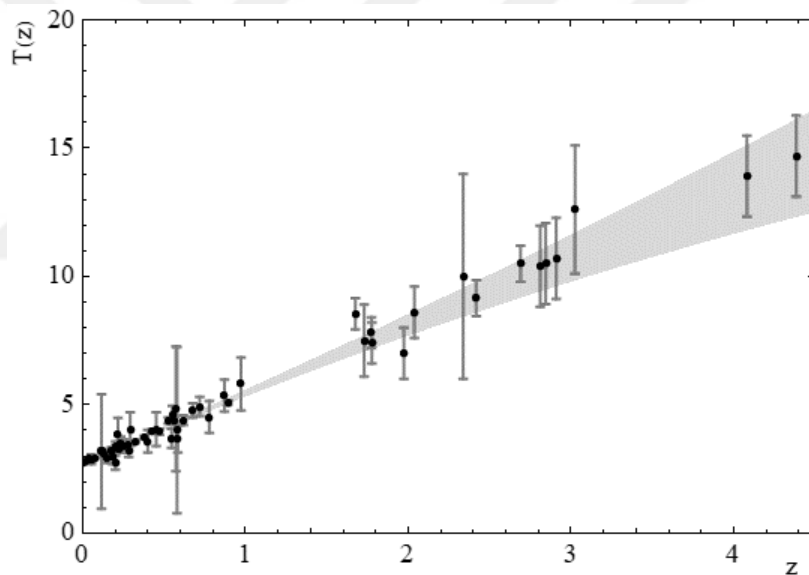
Şekil. 3.6. Belirli aralıklar için R-kare analizimizin sayısal sonuçları.

Şekil 3.5'te siyah noktalarla vurgulanan (ya da Şekil 3.6'da beyaz renkle yazılan) en yüksek R -kare değerleri karşılık gelen aktivasyon fonksiyonu için en uygun hiper-parametre seçimine işaret etmekte olup bu değerler Tablo 3.9'da açık olarak sunulmaktadır (Salti ve Kangal, 2022).

Tablo 3.9. LSTM-Bırakma analizimizde en uygun hiper-parametre değerleri.

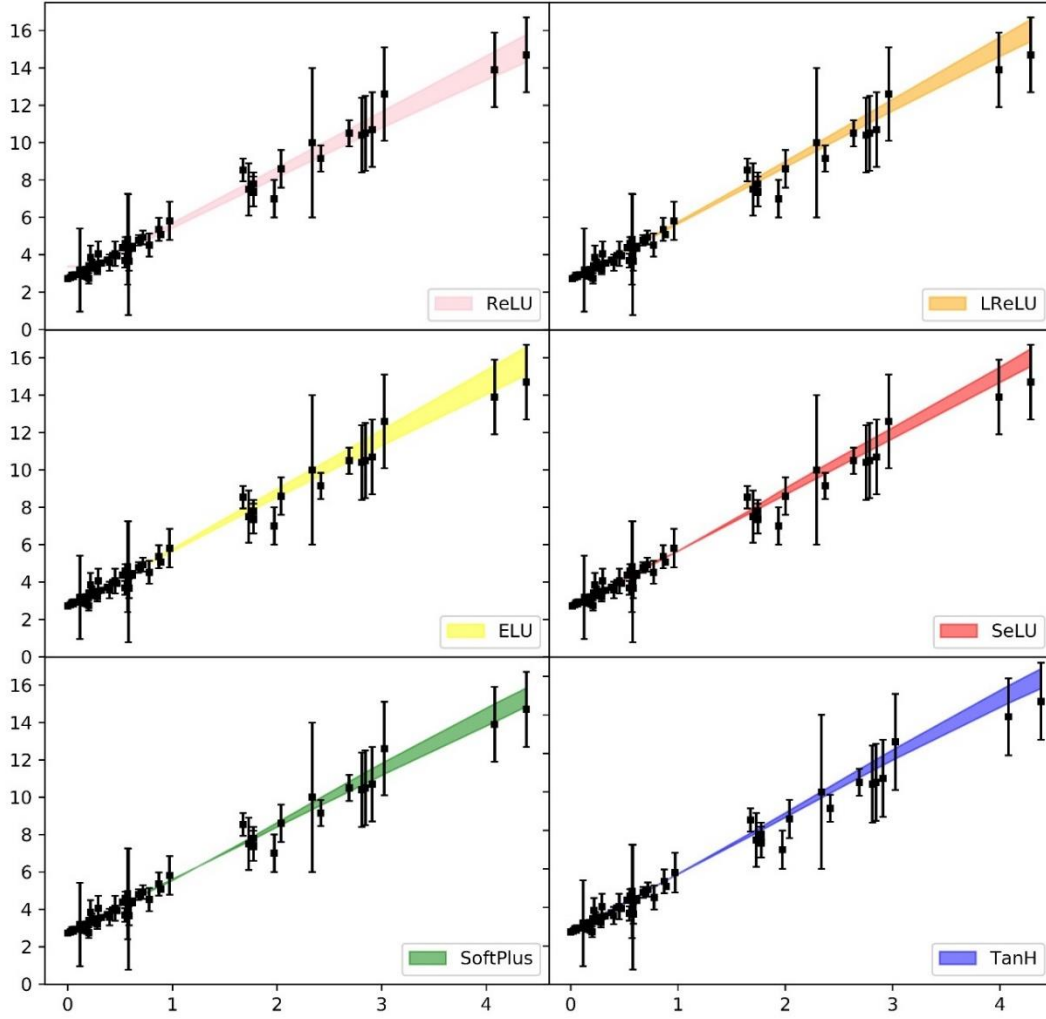
	ReLU	LReLU	ELU	SELU	SoftPlus	TanH
Parti Boyutu	4	4	4	4	4	8
Dönem Sayısı	1400	1300	1400	800	1500	1400
En yüksek R-kare	0.974401	0.969824	0.966063	0.974053	0.966060	0.981696

Denklem 3.6'da sunulan en uygun serbest parametre değerleri FIM yaklaşımında yerine yazılırken Tablo 3.9'da sunulan en uygun hiper-parametre değerleri ise LSTM-Bırakma derin yapılı öğrenme algoritmasında göz önüne alınır. Böylece makine tarafından FIM yaklaşımına göre Şekil 3.7'de sunulan tahmin sonuçları elde edilmiştir (Salti ve Kangal, 2022). Buradaki siyah noktalar deneysel sonuçlara karşılık gelirken çizgiler ise o değerlerin hata payı sınırlarını temsil etmektedir.



Şekil 3.7. FIM yaklaşımına göre KMA ısıtım sıcaklığı tahmin analizi.

Diğer yandan, Şekil 3.8'de ise LSTM-Bırakma algoritmasının sonuçları görülmektedir. Burada gözlemsel veriler ve hata payları siyah nokta ve çizgilerle betimlenmiştir. Ayrıca, Şekil 3.8'de yatay eksen kızılta kayma değerlerine karşılık gelirken düşey eksen ise KMA ısıtım sıcaklığı değerlerini göstermektedir. Hem de Şekil 3.7 hem de Şekil 3.8'de grafiksel temsilleri sunulan iki farklı analizin sonucunda makine tahminlerinin hata payı sınırları içerisinde kaldığı yani gözlemsel verilerle tutarlı oldukları anlaşılmaktadır.



Şekil 3.8. LSTM-Bırakma yaklaşımına göre KMA ısıtım sıcaklığı tahmin analizi.

3.4.2. Hubble Parametresinin Veri-yönlendirmeli Tahmin Temelli Modellemesi

Kozmoloji (ya da evrenbilim) evrenimizin dinamik evrimsel doğasını araştıran bilim dalıdır. 100 yıldan fazla bir süredir evrenimizin genişlediği (yani hacminin arttığı) bilinmekte olup Hubble parametresi bu davranışı betimleyen en önemli niceliktir. Aşağıda, bu parametreyi içeren Hubble Yasası yazılmıştır:

$$\vartheta = Hd. \quad (3.7)$$

Burada ϑ , H ve d nicelikleri sırasıyla gözlenen galaksinin bizden uzaklaşma hızını, Hubble parametresini ve gözlenen galaksinin bize olan uzaklığını göstermektedir. Buradan anlaşılacağı üzere bir galaksi bizden ne kadar uzakta o oranda yüksek bir hızla bizden uzaklaşacaktır ve bu orantıyı betimleyen kozmolojik nicelik ise Hubble parametresidir. Başka bir ifadeyle gözlenen iki galaksiden uzak olan bizden daha hızlı uzaklaşır. Diğer yandan, son birkaç on yıldır evrenimizin kaynağı tam

olarak bilinmeyen bir davranış sergilemeye başladığı ve tahmin edilenden daha hızlı (ivmeli) genişleme sürecine girdiği anlaşılmıştır. Doğal olarak, bu durum Hubble parametresinin önemini daha da arttırmıştır. Bu önemli parametre ile ilgili yapılan gözlemsel veriler Tablo 3.10 ve Tablo 3.11’de kaynaklarıyla birlikte ayrıntılı olarak sunulmuştur.

Tablo 3.10. Güncel Hubble parametresi ölçümleri (1. Kısım).

z	$H \pm \sigma$ (Kelvin)	Kaynak
0.071	69.00±19.7	(Zhang ve ark., 2014)
0.090	69.00±12.0	(Jimenez ve ark., 2003)
0.120	68.60±26.2	(Zhang ve ark., 2014)
0.170	83.00±8.00	(Simon ve ark., 2005)
0.179	75.00±4.00	(Moresco ve ark., 2012)
0.199	75.00±5.00	(Moresco ve ark., 2012)
0.200	72.90±29.6	(Zhang ve ark., 2014)
0.240	79.69±2.65	(Gaztanaga ve ark., 2009)
0.270	77.00±14.0	(Simon ve ark., 2005)
0.280	88.80±36.6	(Zhang ve ark., 2014)
0.352	83.00±13.5	(Moresco ve ark., 2016)
0.380	83.00±17.0	(Moresco ve ark., 2012)
0.400	95.00±17.0	(Simon ve ark., 2005)
0.401	77.00±10.2	(Moresco ve ark., 2016)
0.425	87.10±11.2	(Moresco ve ark., 2016)
0.430	86.50±3.70	(Gaztanaga ve ark., 2009)
0.440	82.60±7.80	(Blake ve ark., 2012)
0.449	92.80±12.9	(Moresco ve ark., 2016)
0.470	89.00±49.6	(Ratsimbazafy ve ark., 2017)
0.478	80.90±9.00	(Moresco ve ark., 2016)
0.480	97.00±62.0	(Stern ve ark., 2010)
0.510	90.40±1.90	(Alam ve ark., 2017)
0.520	94.35±2.64	(Wang ve ark., 2017)
0.560	93.34±2.30	(Blake ve ark., 2012)
0.570	96.80±3.40	(Samushia ve ark., 2013)
0.593	104.0±13.0	(Moresco ve ark., 2012)
0.600	87.90±6.10	(Blake ve ark., 2012)
0.610	97.30±2.10	(Blake ve ark., 2012)
0.640	98.82±2.98	(Wang ve ark., 2017)
0.679	92.00±8.00	(Moresco ve ark., 2012)
0.730	97.30±7.00	(Alam ve ark., 2017)
0.781	105.0±12.0	(Moresco ve ark., 2012)
0.875	125.0±17.0	(Moresco ve ark., 2012)
0.880	90.00±40.0	(Stern ve ark., 2010)
0.900	117.0±23.0	(Simon ve ark., 2005)
1.037	154.0±20.0	(Moresco ve ark., 2012)
1.300	168.0±17.0	(Simon ve ark., 2005)
1.363	160.0±33.6	(Moresco, 2015)
1.430	177.0±18.0	(Simon ve ark., 2005)
1.530	140.0±14.0	(Simon ve ark., 2005)
1.750	202.0±40.0	(Simon ve ark., 2005)
1.965	186.5±50.4	(Moresco, 2015)

Tablo 3.11. Güncel Hubble parametresi ölçümleri (2. Kısım).

z	$H \pm \sigma$ (Kelvin)	Kaynak
2.300	224.0±8.00	(Busca ve ark., 2013)
2.340	222.0±8.50	(Delubac ve ark., 2015)
2.360	226.0±9.30	(Font-Ribera ve ark., 2014)

Hubble parametresi gözlemsel verileri üzerine tamamladığımız ve ML analizi içeren çalışmamız 2022 yılında uluslararası endeksli *Physics Scripta* dergisinde basılmıştır. Bu çalışmamızın ilk aşamasında kozmik Hubble parametresi için kuramsal bir formülasyon önerilmiştir (Salti ve ark., 2022):

$$H(z) = H_0 \left(\Omega_m^0 (1+z)^3 + \Omega_e^0 \left[1 - \varepsilon + \frac{\varepsilon}{(1+z)^B} \right] \right)^{\frac{1}{2}}. \quad (3.8)$$

Burada H_0 , Ω_m^0 , Ω_e^0 , ε ve B nicelikleri modelin keyfi parametreleri olmakla birlikte H_0 , Ω_m^0 ve Ω_e^0 terimlerinin özel anlamları vardır: bu üç nicelik sırasıyla Hubble sabitine (Hubble parametresinin günümüzdeki değeri), günümüzdeki karanlık madde oranına ve günümüzdeki karanlık enerji oranına karşılık gelmektedir. Ayrıca,

$$\Omega_m^0 + \Omega_e^0 = 1 \quad (3.9)$$

biçimindedir. Denklem 3.9 bize denklem 3.8’de esasen dört tane keyfi parametre olduğunu söylemektedir. Bu parametrelerin en uygun değerlerini belirlemek amacıyla genetik ML algoritması tercih edilmiştir. Bu yaklaşımda kullanılmak üzere denklem 3.8’de ifade edilen kuramsal modelleme $C_p(z_i|\theta_\mu)$ biçiminde betimlenmiş olup burada

$$\theta_\mu = (\theta_1, \theta_2, \theta_3, \theta_4) = (H_0, \Omega_e^0, \varepsilon, B) \quad (3.10)$$

tanımı yapılmıştır (Salti ve ark., 2022). Böylece, M_D gözlemsel veri kümesi ve σ_i hata payları olmak üzere, genetik algoritma uyarlama fonksiyonu şöyle tanımlanmıştır (Salti ve ark., 2022):

$$\chi_H^2(\theta) = \sum_i^{45} \frac{[C_p(z_i|\theta) - M_D(z_i)]^2}{\sigma_i^2}. \quad (3.11)$$

Genetik algoritmanın önemli hiper parametreleri için Tablo 3.7’de sunulan varsayımların kullanılmasıyla birlikte kuramsal modelin serbest parametrelerinin gözlemsel verilere yönelik en uygun değerleri Tablo 3.12’deki gibi elde edilmiştir (Salti ve ark., 2022).

Tablo 3.12. Hubble parametresi modelimizdeki keyfi niceliklerin en uygun değerleri.

	H_0	Ω_e^0	ε	B
Değer	69.3226	0.6496	0.4489	1.4324

Ayrıca, denklem 3.9’dan

$$\Omega_m^0 = 1 - \Omega_e^0 = 0.3504 \quad (3.12)$$

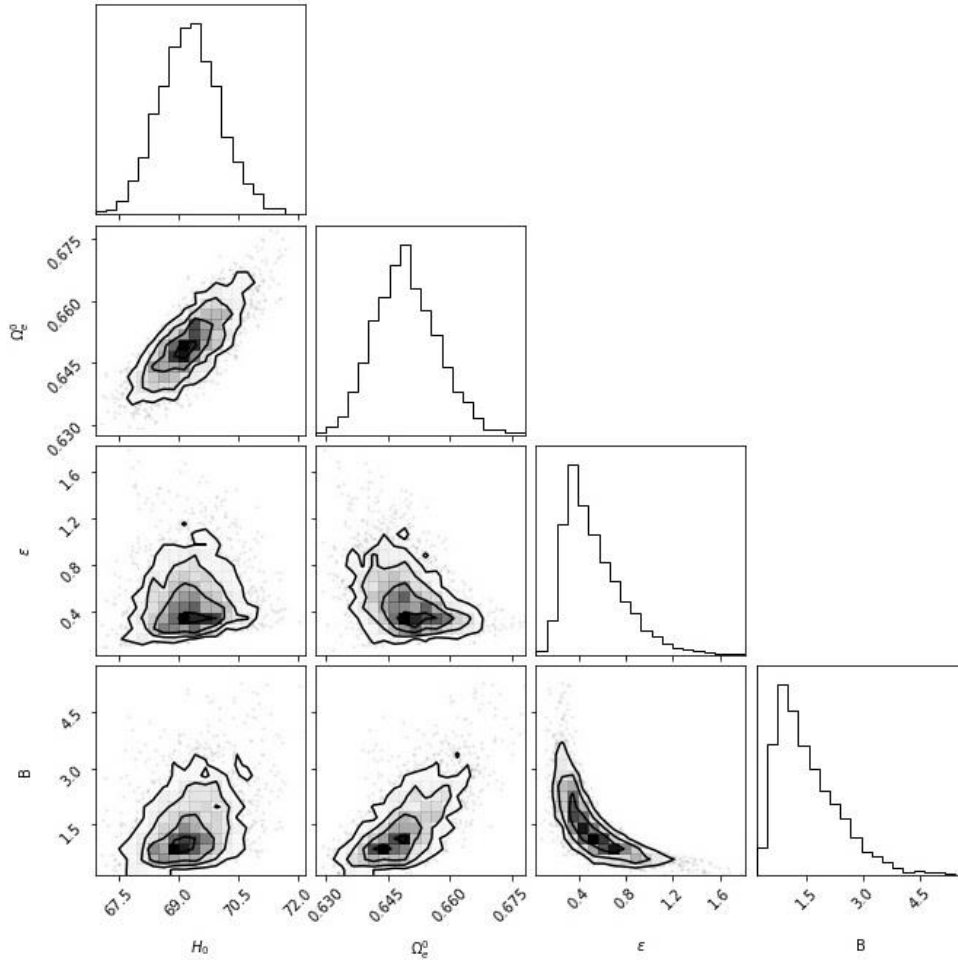
olması gerektiği anlaşılmaktadır. Öte yandan, serbest model parametrelerinin değerlerine yönelik olası hesaplama hataları tanım aralıkları FIM yaklaşımı ile hesaplanmıştır. Buradan yola çıkarak ilgili yaklaşımda kullanılmak üzere bilgi matrisi ise şöyle yazılmıştır (Salti ve ark., 2022):

$$F_{\mu\nu} = \begin{pmatrix} \frac{\partial^2 \chi_H^2}{\partial H_0^2} & \frac{\partial^2 \chi_H^2}{\partial H_0 \partial \Omega_e^0} & \frac{\partial^2 \chi_H^2}{\partial H_0 \partial \varepsilon} & \frac{\partial^2 \chi_H^2}{\partial H_0 \partial B} \\ \frac{\partial^2 \chi_H^2}{\partial \Omega_e^0 \partial H_0} & \frac{\partial^2 \chi_H^2}{\partial \Omega_e^0{}^2} & \frac{\partial^2 \chi_H^2}{\partial \Omega_e^0 \partial \varepsilon} & \frac{\partial^2 \chi_H^2}{\partial \Omega_e^0 \partial B} \\ \frac{\partial^2 \chi_H^2}{\partial \varepsilon \partial H_0} & \frac{\partial^2 \chi_H^2}{\partial \varepsilon \partial \Omega_e^0} & \frac{\partial^2 \chi_H^2}{\partial \varepsilon^2} & \frac{\partial^2 \chi_H^2}{\partial \varepsilon \partial B} \\ \frac{\partial^2 \chi_H^2}{\partial B \partial H_0} & \frac{\partial^2 \chi_H^2}{\partial B \partial \Omega_e^0} & \frac{\partial^2 \chi_H^2}{\partial B \partial \varepsilon} & \frac{\partial^2 \chi_H^2}{\partial B^2} \end{pmatrix}. \quad (3.13)$$

Böylece, keyfi model parametreleri için aşağıdaki nihai sonuçlar elde edilmiştir (Salti ve ark., 2022):

$$\theta_\mu \Rightarrow \begin{cases} H_0 = 69.3226 \mp 7.26 \\ \Omega_e^0 = 0.6496 \mp 0.11 \\ \varepsilon = 0.4489 \mp 0.02 \\ B = 1.4324 \mp 0.30. \end{cases} \quad (3.14)$$

Elde edilen uyarılma değerleri için olasılık dağılımları ise Şekil 3.9'da sunulmuştur (Salti ve ark., 2022).



Şekil 3.9. Hubble parametresi modeli keyfi niceliklerinin olasılık dağılımları.

Serbest model parametreleri arasında olan H_0 , Ω_m^0 ve Ω_e^0 terimlerinin özel anlamları olduğunu ve bunların sırasıyla Hubble sabitine, günümüzdeki karanlık madde oranına ve günümüzdeki karanlık enerji oranına karşılık geldiğini söylemiştik. Bu serbest parametreler için ML yaklaşımı tarafından önerilen sonuçların tutarlılığını kontrol etmek için güncel gözlemsel verilere bakmak gerekmektedir. Bu bağlamda Planck teleskobu ile elde edilen sonuçlardan (Aghanim ve ark., 2020)

$$\Omega_m^0 = 0.315 \pm 0.007, \quad (3.15.)$$

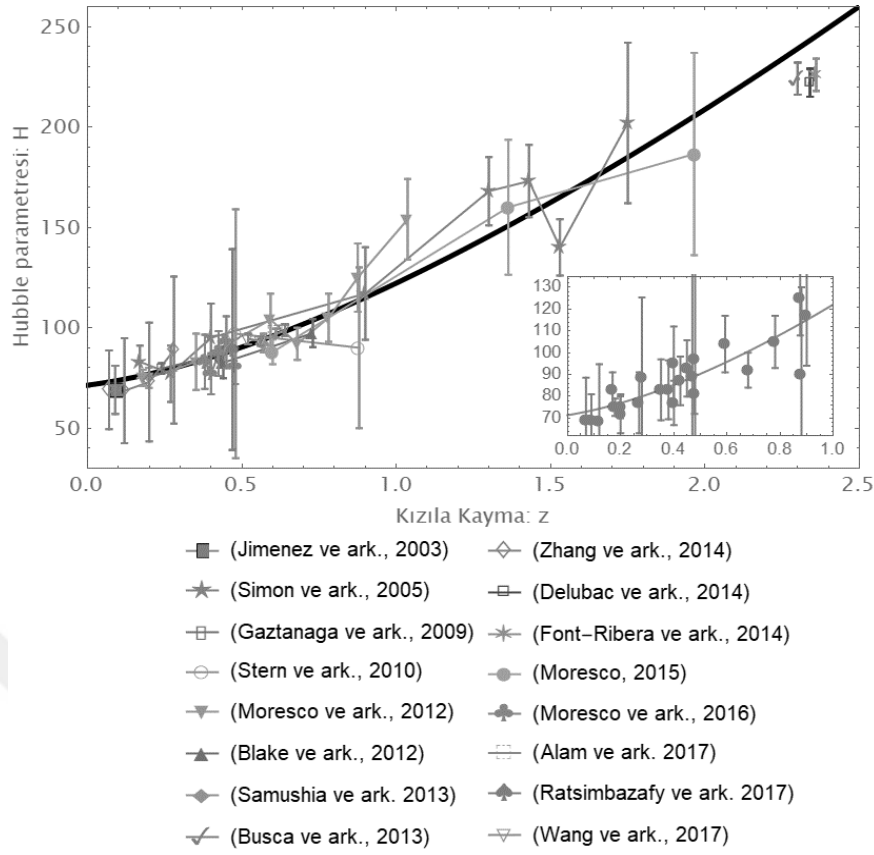
$$\Omega_e^0 = 1 - \Omega_m^0 \approx 0.685 \quad (3.16.)$$

olduğu bilinmektedir. Öte yandan, Hubble sabiti için elde edilen nümerik değer farklı kaynaklara göre şöyledir (bakınız Tablo 3.13):

Tablo 3.13. Güncel Hubble sabiti ölçümleri.

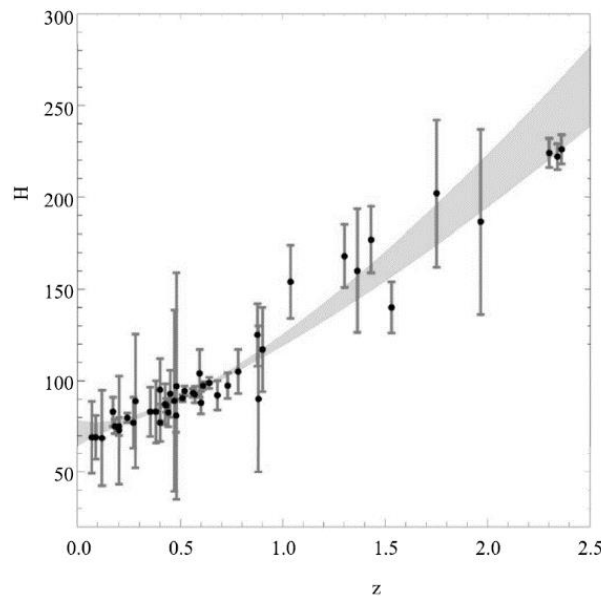
Proje	$H_0 \left[\frac{\text{km}}{\text{Sec Mpc}} \right]$	Kaynak
Hooker Teleskobu	500.30 ± 1.20	(Hubble, 1929)
1996 Öncesi	$\sim(50 - 90)$	(Overbye, 1999)
NASA Hubble Uzay Teleskobu	72.00 ± 8.00	(Freedman ve ark., 2001)
Chandra X-Işını Gözlemevi	$76.90^{+10.7}_{-8.70}$	(Bonamente ve ark., 2006)
WMAP İş Birliği	70.50 ± 1.30	(Hinshaw ve ark., 2009)
WMAP İş Birliği	69.32 ± 0.80	(Bennett ve ark., 2013)
Planck Uzay Aracı	67.80 ± 0.90	(Ade ve ark., 2016)
Spitzer Uzay Teleskobu	$76.20^{+3.40}_{-2.70}$	(Tully ve ark., 2016)
LIGO - VIRGO et al.	$70.00^{+12.0}_{-8.00}$	(Ligo ve diğer iş birlikler, 2017)
BAO Ölçümü	67.40 ± 4.00	(Yu ve ark., 2018)
Planck Uzay Aracı	67.40 ± 0.50	(Aghanim ve ark., 2020)
NASA Hubble Uzay Teleskobu	69.80 ± 2.50	(Freedman ve ark., 2019)
NASA Hubble Uzay Teleskobu	74.30 ± 1.42	(Riess ve ark., 2019)
Megamaser Kozmoloji Projesi	73.90 ± 3.00	(Pesce ve ark., 2020)
Gaia Uzay Aracı	72.10 ± 2.00	(Soltis ve ark. 2021)
Carnegie Süpernova Projesi	69.80 ± 1.60	(Freedman, 2021)
HST, Gaia EDR3, Pantheon+	73.04 ± 1.04	(Riess ve ark., 2022)
Tip II Süpernova Verileri	$75.40^{+3.80}_{-3.70}$	(Jaeger ve ark., 2022)
Hızlı Radyo Veri Patlamaları	62.30 ± 9.10	(Hagstotz ve ark., 2022)
Pantheon+	$73.40^{+0.99}_{-1.22}$	(Brout ve ark., 2022)

Denklem 3.14'te sunulan ve genetik-FIM ML algoritmasıyla elde edilen uyarlama sonuçları denklem 3.15, denklem 3.16 ve Tablo 3.13'te sunulan sonuçlarla karşılaştırıldığında ML yaklaşımıyla elde edilen verilerin deneysel verilerle çok iyi uyduğu görülmektedir. Buna ek olarak Hubble parametresinin doğası grafiksel analiz ile ele alınarak gözlemsel verilerle karşılaştırıldığında modelin başarısı daha açık biçimde ortaya konabilmektedir (bakınız Şekil 3.10). Görüldüğü üzere kuramsal $H(z)$ fonksiyonun davranışı (kalın siyah eğri) gözlemsel verilerle (hata oranları ile birlikte sunulan noktalar) çok iyi uyumaktadır, bu durum bize önerilen $H(z)$ fonksiyonunun Hubble parametresi için başarılı bir kuramsal modelleme olarak kullanılabileceğini göstermektedir.



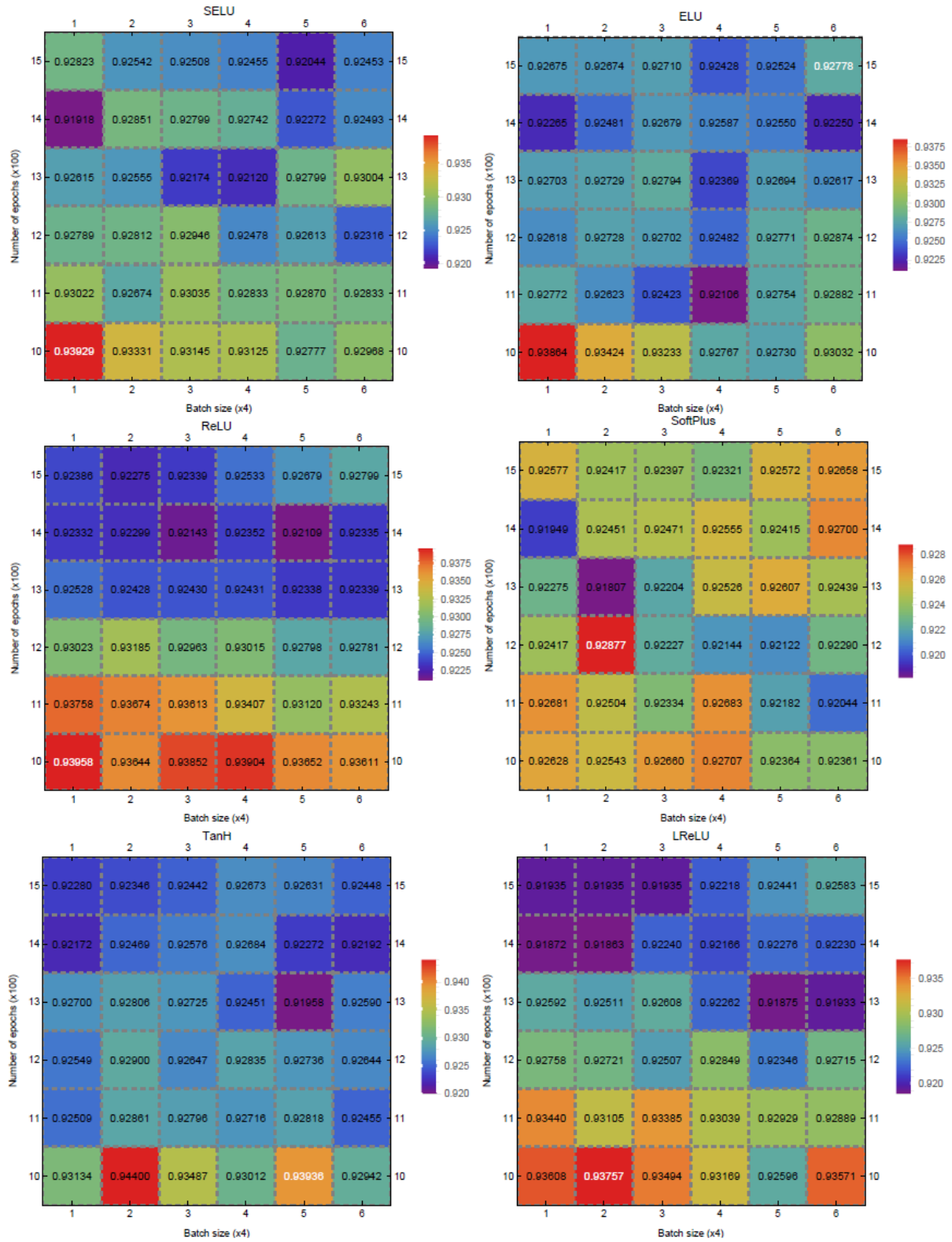
Şekil 3.10. $H(z)$ fonksiyonunu davranışı ve güncel gözlemsel Hubble parametresi verileri.

Son olarak kuramsal model aracılığıyla eğitilen FIM algoritmasının tahmin sonuçları ile gözlemsel verilerin (hata payları ile birlikte gösterilen siyah noktalar) karşılaştırılmasından elde edilen dikkat çekici özellik Şekil 3.11’de açıkça görülmektedir: eğitilen makine oldukça başarılı tahminler yaparak farklı kıızıla kayma değerleri için güvenilir sonuçlar verebileceğini göstermektedir.



Şekil 3.11. FIM yaklaşımı ile elde edilen tahmin $H(z)$ verileri.

3.4.3. Derin Öğrenme Destekli Hubble Parametresi Analizi



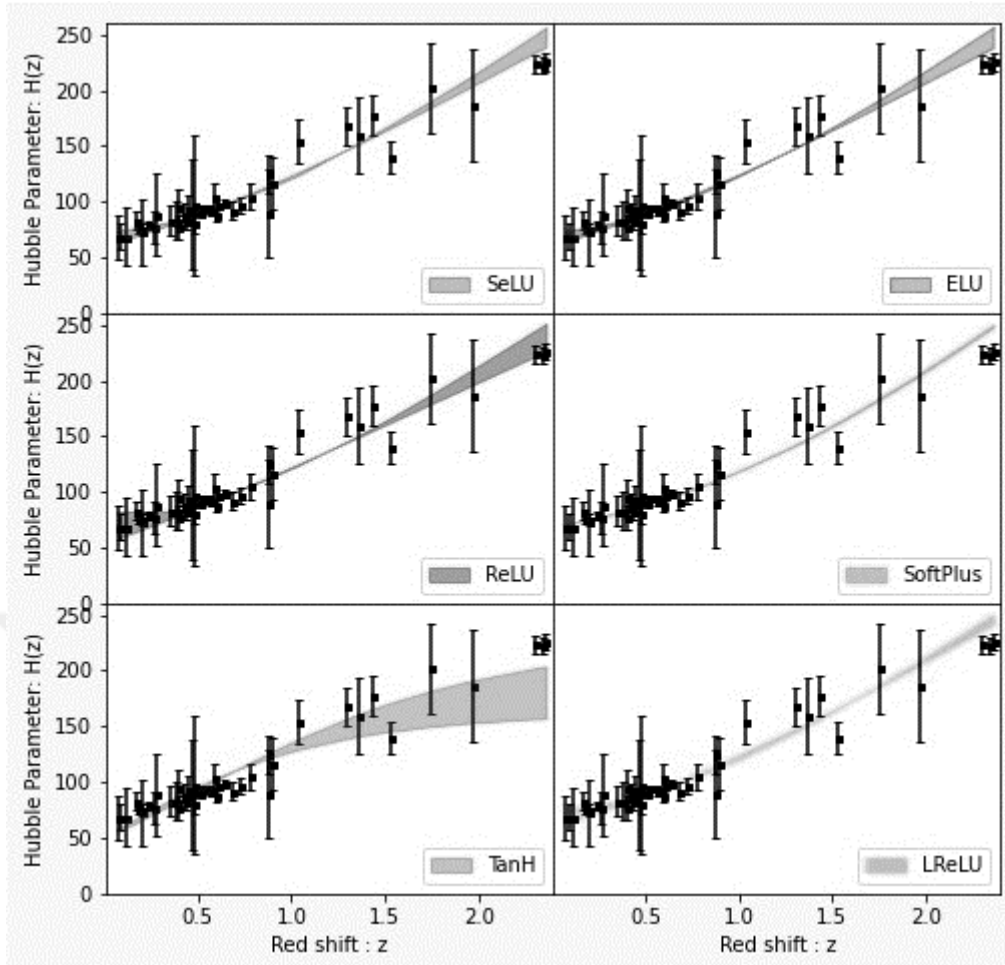
Şekil 3.12. Hiper-parametre ayarlama evresi nümerik R -kare sonuçları.

Bir önceki bölümde kuramsal olarak inşa edilen bir Hubble parametresi modelinin ML yaklaşımları yardımıyla gözlemsel verilere uyumlu hale getirilebilmesine yönelik bir metot paylaşılmıştır. Bu bölümde ise başarılı olduğu ortaya konan $H(z)$ fonksiyonunun derin öğrenme analizi sunulacaktır (Salti ve ark., 2023). Bölüm 3.4.1’de KMA ışıma sıcaklığı için oluşturulan LSTM-Bırakma analizi yöntemi bu defa Hubble parametresi için takip edilecektir. Elde edilen yöntemin farklı bir modelde de başarılı sonuçlar verebileceğini göstermek takip edilen metodun güvenilirliğini ve tutarlılığını ortaya koyacaktır. Bilindiği üzere öncelikle modelin hiper-parametrelerinin ayarlanması yani başarılı tahmin analizi elde edebilmemizi sağlayacak en uygun hiper-parametre değerlerinin belirlenmesi gerekmektedir. Bu bağlamda parametre ayarlama evresinde takip edilen yöntem şu ana basamaklardan oluşmaktadır (Salti ve ark., 2023): (i) öğrenme mimarisinin inşası için MSE kayıp fonksiyonu, ADAM optimize edici ve 100 nöron içeren LSTM hücresi kullanılmıştır, (ii) algoritmanın gizli katmanlarında ELU, SELU, ReLU, LReLU, SoftPlus ve TanH tipi aktivasyon fonksiyonları kullanılmıştır, (iii) Bırakma olasılığı değeri $p_b=0.5$ olarak seçilmiştir, (iv) Tablo 3.10 ve Tablo 3.11 sıralı hale getirilerek düzenlenen gözlemsel veriler karşılaştırma analizinin temelini oluşturmuştur, (v) modülde parti büyüklüğü hiper-parametresi için (4, 8, 12, 16, 20) değerleri ve dönem sayısı için de (500, 600, ..., 1500) sayıları kullanılmıştır. Sonuç olarak elde edilen nümerik değerler Şekil 3.12’de sunulmaktadır (Salti ve ark., 2023). Burada en yüksek R-kare değerleri (beyaz renkli veriler) en uygun hiper-parametre seçimlerine işaret etmektedir ve bu seçimler Tablo 3.14’te özetlenmiştir (Salti ve ark., 2023).

Tablo 3.14. Hubble parametresi modelimiz için uygun hiper-parametre değerleri.

Aktivasyon Fonksiyonu	Parti Boyutu	Dönem Sayısı	En Yüksek R-kare
ReLU	4	1000	0.93958
LReLU	8	1000	0.93757
ELU	24	1500	0.92778
SELU	4	1000	0.93929
TanH	20	1000	0.93936
SoftPlus	8	1200	0.92877

LSTM-Bırakma modülünün en uygun hiper-parametre değerleri belirlendikten sonra bu analize göre çalıştırılan mimari varyantlarında Şekil 3.13’te sunulan analiz sonuçları elde edilmiştir. Görüldüğü üzere derin yapılı LSTM-Bırakma öğrenme algoritması başarılı tahmin verileri önermiştir.



Şekil 3.13. LSTM-Bırakma algoritmasının Hubble parametresi için verdiği tahmin sonuçları.

4. BULGULAR ve TARTIŞMALAR

4.1. Veri Önışleme¹⁶¹

Tez kapsamında ele alınan ve Tablo 3.2, Tablo 3.3 ve Tablo 3.4’te sunulan tele-pazarlama veri setinde görüleceğı üzere nitelikleri farklı veri türleri bulunmaktadır: bu veri tiplerini Tablo 4.1’de sunulduğu gibi gruplandırmak mümkündür.

Tablo 4.1. Veri kümesindeki öznitelik türleri.

Öznitelik	Tip
Call ID	Nümerik
Sales	Kategorik
Agent ID	Nümerik
Age	Nümerik
Product ID	Nümerik
Time Zone	Nümerik
Phone Code	Nümerik
First Name	Kategorik
Last Name	Kategorik
Area Code	Nümerik
Gender	Kategorik
Call Count	Nümerik

İstatistiksel analizimizde "Sales (Satışlar)" özniteliğı *hedef özellik*¹⁶² olarak tanımlanırken diğler tüm özellikler *argümanlar*¹⁶³ olarak dikkate alınmıştır. Tablo 4.1’den ele alınan tele-pazarlama verilerinin aynı türde olmadığı görölmektedir ve bu durum öğrenme algoritmasının tutarlılığına doğrudan etki edecektir. Dolayısıyla, veri önışleme aşamasında bir öznitelik dönüşümü yapılması gerektiğı anlaşılmaktadır.

4.1.1. Etkisiz Özniteliklerin Dışlanması

Kullanılan veri kümesinde bulunan “Time Zone (Zaman bölgesi)”, Phone Code (Telefon kodu)”, “First Name (İlk Ad)”, “Last Name (Soyadı)” ve “Area Code (Alan kodu)” özniteliklerinin hedef parametreyi temsil eden “Sales (Satışlar)” kolonunu dikkate değer ölçüde etkilemeyeceğı aşıkardır. Ayrıca kimi müşterilerin bazı bilgilerinde eksiklikler olduğu görölmektedir. Bu nedenle veri seti adresinin düzgün tanımlanması ve yukarıda bahsi geçen özniteliklerin göz ardı edilmesi için Şekil 4.1’de verilen kodlama satırları kullanılmıştır. Veri setimiz artık 223 acente kimliğı (Agent ID), 10 ürün çeşidi (Product ID) ve 97205 müşteri içermektedir.

¹⁶¹ Literatürde bu aşama “data preprocessing” olarak bilinmektedir.

¹⁶² Target feature.

¹⁶³ Arguments.

```

In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
#veri yolu
veri= pd.read_csv("C:\\...\\dataset.csv")
#veri temizleme
veri=veri.drop(["Call_ID","Timezone","First_Name","Last_Name","Phone_code"],axis=1)
veri.dropna(inplace=True)
veri.reset_index(drop=True,inplace=True)
veri.head()

```

```

Out[1]:
   Sale  Agent_ID  Age  Product_ID  Area_Code  Gender  Call_Count
0  False    5265.0   42         147    2302.0   Male         1
1  False    5226.0   74         146    1501.0   Male        10
2  False    4452.0   35         144    1550.0   Male         9
3  False    5461.0   40         149    1401.0   Male         6
4  False    5298.0   26         147     125.0  Female        12

```

Şekil 4.1. Veri yolu tanımlama ve işlevsiz özniteliklerin temizlenmesi.

4.1.2. Veri Tipi Dönüştürme

Bağımsız değişkenler ve seçilen hedef parametre arasında bir ML modelinin oluşturulup oluşturulamayacağını tartışılması için ele alınan tele-pazarlama veri setindeki kategorik bilgilerin kukla değişkenler yardımıyla ikili değerlere dönüştürülmesi gerekmektedir. Diğer bir deyişle metin tabanlı “Sales (Satışlar)” ve “Gender (Cinsiyet)” kolonları nümerik biçimde ifade edilmelidir. Bu bağlamda ele alınan temel kabullenmeler Tablo 4.2 ve Tablo 4.3’te sunulmaktadır.

Tablo 4.2. Kategorik “Sales (Satışlar)” bilgilerinin sayısal dönüşümü.

Öznitelik	Sales	
	True	False
Nümerik Değer	1	0

Tablo 4.3. Kategorik “Gender (Cinsiyet)” bilgilerinin sayısal dönüşümü.

Öznitelik	Gender		
	Male	Female	Non-binary
Nümerik Değer	1	0	2

Tablo 4.2 ve Tablo 4.3’te sunulan varsayımlar için öncelikle Şekil 4.2.’de gösterilen kodlamalar kullanılmış, daha sonra bu verilerin algoritmaya girilmesi için de Şekil 4.3’teki komut satırlarından faydalanılmıştır.

In [2]: *#Kategorik veri donusumu*

```
for i in veri.keys():
    if i=='Sale' or i=="Gender":
        veri[i]=veri[i].astype('category').cat.codes
veri.head()
```

Out[2]:

	Sale	Agent_ID	Age	Product_ID	Area_Code	Gender	Call_Count
0	0	5265.0	42	147	2302.0	1	1
1	0	5226.0	74	146	1501.0	1	10
2	0	4452.0	35	144	1550.0	1	9
3	0	5461.0	40	149	1401.0	1	6
4	0	5298.0	26	147	125.0	0	12

Şekil 4.2. Kategorik verilerin dönüştürülmesi.

In [3]: *#numerik veri girisi*

```
veri["Agent_ID"] = veri["Agent_ID"].apply(pd.to_numeric, errors='coerce')
veri["Product_ID"] = veri["Product_ID"].apply(pd.to_numeric, errors='coerce')
veri["Age"] = veri["Age"].apply(pd.to_numeric, errors='coerce')
veri["Area_Code"] = veri["Area_Code"].apply(pd.to_numeric, errors='coerce')
veri["Gender"] = veri["Gender"].apply(pd.to_numeric, errors='coerce')
veri["Call_Count"] = veri["Call_Count"].apply(pd.to_numeric, errors='coerce')
veri.head()
```

Out[3]:

	Sale	Agent_ID	Age	Product_ID	Area_Code	Gender	Call_Count
0	0	5265.0	42	147	2302.0	1	1
1	0	5226.0	74	146	1501.0	1	10
2	0	4452.0	35	144	1550.0	1	9
3	0	5461.0	40	149	1401.0	1	6
4	0	5298.0	26	147	125.0	0	12

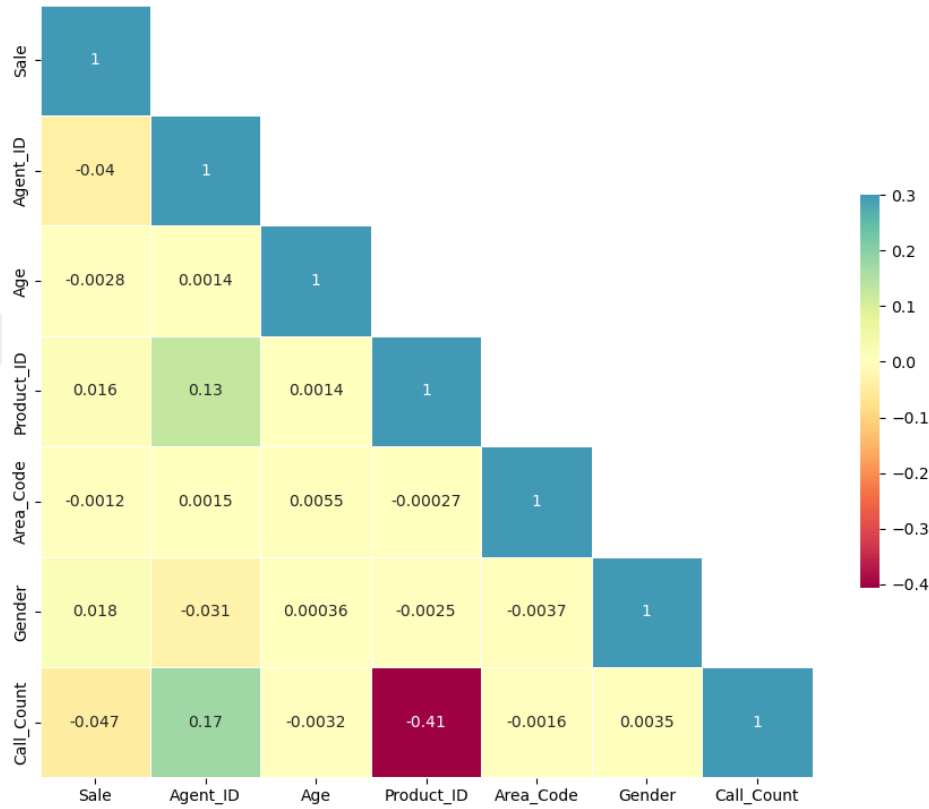
Şekil 4.3. Nümerik verilerin algoritmaya tanıtılması.

4.1.3. Bilgi Görselleştirme

Herhangi bir veri kümesinin başarılı biçimde işlenebilmesi için çok iyi anlaşılması gerekmektedir olup bu bağlamda kullanılan yaygın yöntemlerden bir tanesi de görsel analizdir. Bu alt-başlık kapsamında veri setinde doğrudan gözükmeyen bazı gizli özelliklerin grafiksel analizler yoluyla ortaya çıkarılması hedeflenmektedir.

ML analizlerini olumsuz etkileyebilecek önemli noktalardan bir tanesi de nitelikler arasındaki korelasyon durumlarıdır. Bir korelasyon tablosu tipik olarak kare matris şeklindedir ve seçilen değişkenleri temsil eden sütunlar ile satırları içermektedir (bu matris simetrik olduğundan üçgen biçiminde de gösterilebilmektedir). Bu matristeki her hücre çeşitli nicelikler arasındaki ilişkiyi temsil eden sayılarla temsil edilir. Herhangi iki özellik arasındaki korelasyon değeri $[-1, -0.5]$ aralığında ise onlar arasında negatif yönlü bir ilişki (ya da korelasyon), $[0.5, 1]$ aralığında ise pozitif yönlü bir ilişki

vardır denir. Diğer değerler korelasyon olmayan durumları göstermekte olup 1'leri içeren köşegen elemanları ise her değişkenin kendisiyle daima mükemmel bir şekilde ilişkili olduğunu ima etmektedir. Etkisiz öz niteliklerin temizlenmesi aşamasından geçmiş olan incelenen veri seti öz nitelikleri arasındaki ilişki katsayıları Şekil 4.4'te verilen korelasyon tablosunda yazılmıştır. Bu tablo için gerekli kodlama satırları ise Şekil 4.5'te paylaşılmıştır.



Şekil 4.4. Ele alınan öz niteliklerin korelasyon katsayıları.

```
In [6]: mask = np.triu(np.ones_like(veri.corr(), dtype=bool))

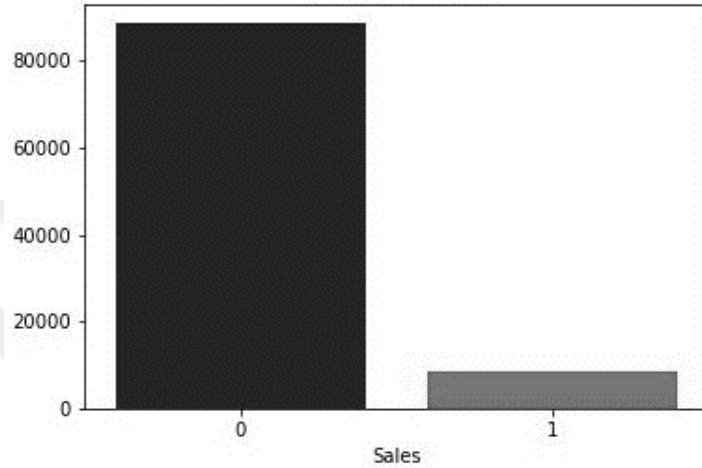
# Set up the matplotlib figure
f, ax = plt.subplots(figsize=(11, 9))
np.fill_diagonal(mask, False)
# Generate a custom diverging colormap
cmap = sns.diverging_palette(230, 20, as_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(veri.corr(), mask=mask, cmap="Spectral", vmax=.3, center=0,
            square=True, linewidths=.5, annot=True, ax=ax, cbar_kws={"shrink": .5})
plt.savefig("cor_table.png")
```

Şekil 4.5. Korelasyon tablosu komut satırları.

Bu analizden öz nitelikler arasında herhangi bir bağlantı sorunu olmadığı yani yapılacak ML analizinin performansını olumsuz etkileyebilecek bir durumun bulunmadığı anlaşılmaktadır.

Daha önce bahsedildiği üzere denetimli öğrenme algoritmalarını kullanırken ele alınan verilerin dengeli olup olmadığı önemlidir, zira veri grubu içerisinde dengesiz bir dağılım varsa algoritmalar azınlık özelliklerini göz ardı ederek taraflı öngörü analizi yapabilmektedir. Faydalı bir analizin öncelikle tutarlı olması gerektiğinden ele aldığımız veri setinde dengesizlik varsa bu sorunu SMOTE veya benzeri bir yaklaşımla ortadan kaldırmamız gerekmektedir. Bu bağlamda veri setindeki hedef öznitelik “Sales (Satışlar)” için pazarlama sonuçlarının dağılımı Şekil 4.6’da sunulmuş ve buradan üzere elimizde dikkatlice düzenlenmesi gereken dengesiz bir veri kümesinin bulunduğu anlaşılmıştır.



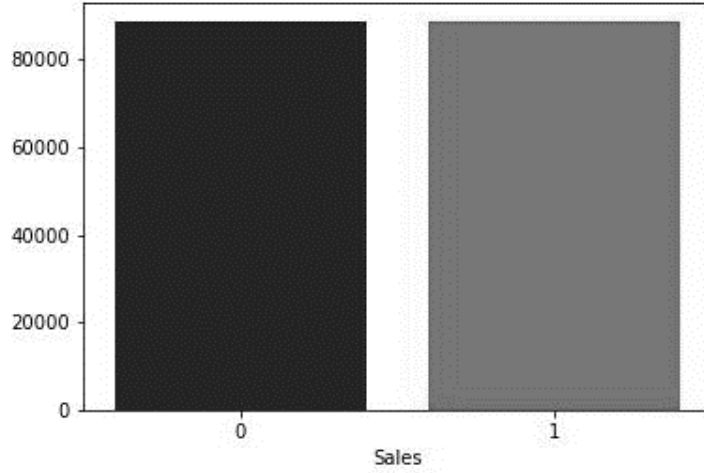
Şekil 4.6. “Sales (Satışlar)” özniteliğinin seçenek dağılımları.

4.1.4. Verilerin Dengeli Biçime Genişletilmesi

Şekil 4.6’dan veri setindeki “olumlu\pozitif satış (True Sales)” sayısının “olumsuz\negatif satış (False Sales)” sayısına oranla az olduğu görülmekte ve bu durum kullanacağımız öğrenme algoritmasını olumsuz yönde etkileyecektir. Bu aşamada SMOTE yardımıyla verilerin dengelenmesi (kodlama satırları için Şekil 4.7’ye bakılabilir) sağlanmıştır (bakınız Şekil 4.8).

```
In [7]: adata.dropna(inplace=True)
adata.reset_index(drop=True,inplace=True)
#karar kolonunu ayıklama
y=adata["Sale"].values
#Karar kolonunun cikarilmasi
adata=adata.drop(["Sale"],axis=1)
#veri turetme noktası
from sklearn.model_selection import train_test_split
from collections import Counter
from sklearn.datasets import make_classification
from imblearn.over_sampling import SMOTE
X = adata[adata.keys()].values
oversample = SMOTE()
X, y = oversample.fit_resample(X, y)
```

Şekil 4.7. SMOTE ile “True (ya da 1)” etiketli yeni veri sentezleme.



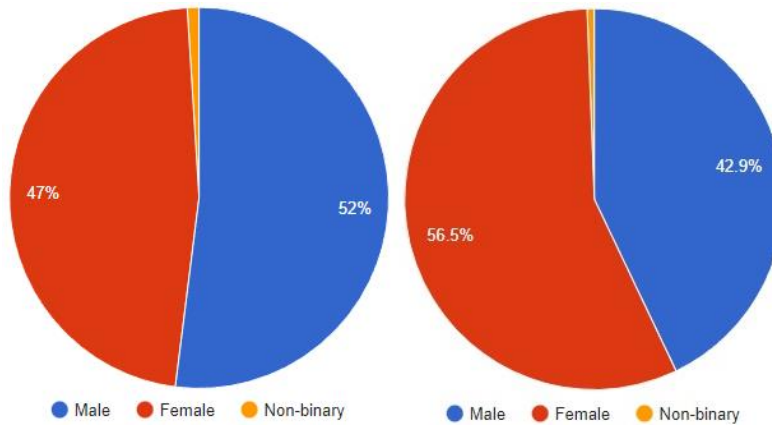
Şekil 4.8. “Sales (Satışlar)” öznitelığının SMOTE sonrası seçenек dağılımları.

SMOTE yaklaşımı veri sentezleme işlemini yaparken kullandığımız tele-pazarlama veri setini çalışmış, böylece diğer özniteliklerde de bir dizi değişimlere neden olmuştur. Araştırmamız kapsamında dengelenmiş yeni veri kümesinin gizli özelliklerini açığa çıkarmak adına çeşitli grafiksel analizler yapılmış ve böylece SMOTE sonrası değişimler de açıkça ortaya konmuştur. Öncelikle Şekil 4.9 ve 4.10’da müşterilerin cinsiyet dağılımları tartışılmış ve SMOTE sonrasında kadın müşterilerin sayısının arttığı açık bir şekilde gösterilmiştir. Böylece, ham verilerden (SMOTE öncesindeki) pazarlamaya olumlu dönüş yapan müşterilerin büyük oranda kadın olduğu anlaşılmaktadır, zira yeni pozitif veriler mevcut pozitif verilere bakılarak sentezlenmiştir.

```
In [14]: print(kadin.shape[0]/bdata.shape[0], " ", erkek.shape[0]/bdata.shape[0], " ", nb.shape[0]/bdata.shape[0])
0.4696556619787862  0.5207765352208311  0.009567802800382712

In [18]: print(akadin.shape[0]/naveri.shape[0], " ", a erkek.shape[0]/naveri.shape[0], " ", anb.shape[0]/naveri.shape[0])
0.565014833782671  0.4295310719805074  0.005454094236821623
```

Şekil 4.9. SMOTE öncesi (üst) ve sonrası (alt) müşteri cinsiyetleri dağılımı kod satırları.



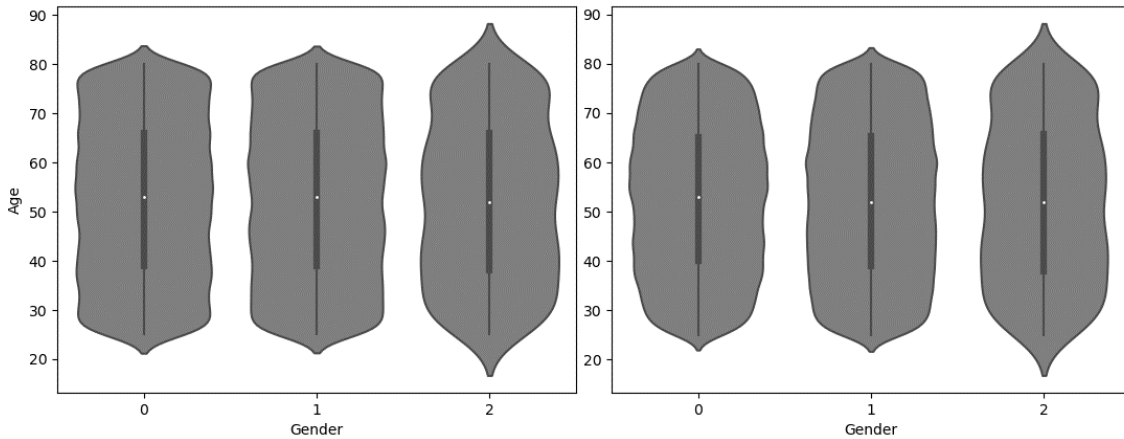
Şekil 4.10. SMOTE öncesi (sol) ve sonrası (sağ) müşteri cinsiyetleri dağılımı.

Müşteri profilini incelemeye yönelik araştırmalarda *keman grafikleri*¹⁶⁴ yaklaşımını uygulamak duruma farklı bir bakış açısı kazandıracaktır. Basit bir anlatımla, bir keman grafiği çizimi verilerdeki tepe noktalarını gösteren bir kutu grafiği ile bir çekirdek yoğunluğu grafiğinin karışımıdır ve sayısal verilerin dağılımını görselleştirmek için sıklıkla kullanılmaktadır. Yalnızca özet istatistikleri gösterebilen bir kutu çizimi yönteminden farklı olarak keman grafikleri özet istatistikleri ile birlikte her bir değişkenin yoğunluğunu da göstermektedir. Peki bu grafikler nasıl okunur? Grafiklerdeki beyaz nokta orta noktayı temsil ederken kalın gri çubuk ise betimsel istatistikteki *çeyrekler açıklığı*¹⁶⁵ olarak bilinmektedir. Diğer yandan, ince gri çizgi ise "aykırı" olarak belirlenen noktalar hariç dağılımın geri kalanını temsil etmektedir. Şekil 4.11 ve Şekil 4.12’de paylaşılan grafiklerden görüleceği üzere müşterilerin cinsiyet-yaş dağılımı keman grafiği yöntemiyle ele alınmış ve müşterilerin yoğun olarak 35-65 yaş aralığında olduğu anlaşılmıştır.

```
In [7]: sns.violinplot(data=bdata,y="Age",x="Gender",palette=['red','blue','Orange'])
        #plt.ylim(0,60)
        plt.ylabel("Age")
        plt.title("Before the SMOTE")
        plt.savefig("age_gen.eps")
        plt.savefig("age_gen.png")

In [11]: sns.violinplot(data=naveri,y="Age",x="Gender",palette=['red','blue','Orange'])
         #plt.ylim(0,60)
         plt.ylabel("Age")
         plt.title("After the SMOTE")
         plt.savefig("age_gen.eps")
         plt.savefig("age_gen.png")
```

Şekil 4.11. SMOTE öncesi (üst) ve sonrası (alt) müşterilerin yaş-cinsiyet keman dağılımı kodları.



Şekil 4.12. SMOTE öncesi (sol) ve sonrası (sağ) müşterilerin yaş ve cinsiyet keman dağılımları.

¹⁶⁴ Violin plots.

¹⁶⁵ Inter Quantile Range (IQR): Çeyrekler açıklığı sıralanmış bir veri dizisinin orta yarısını (%50'sini) domine eden ve üçüncü dörtte-birlik (Q3) sıralanmış veri dizisi ile birinci dörtte-birlik (Q1) sıralanmış veri dizisi aralığını (veya farkını, yani Q3 - Q1) gösteren bir nevi yayılma ölçüsüdür.

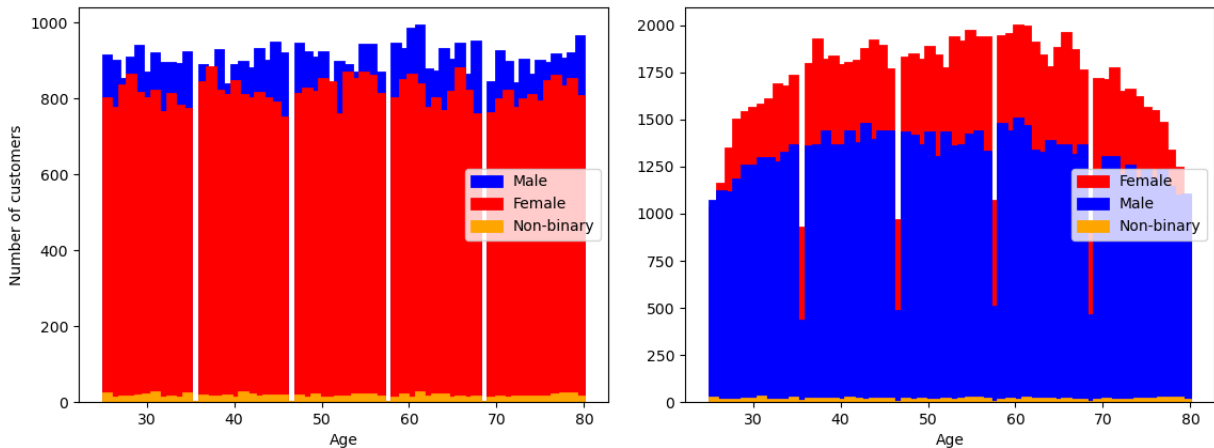
Buraya kadar ele alınan müşteri profili analizini bir adım öteye götürerek için şimdi de müşteri sayısı ile yaşları arasındaki dağılıma odaklanalım, zira bu araştırma bize farklı bilgiler sunacaktır. Bu bağlamda, Şekil 4.13'te müşteri sayılarının yaşlarına göre tartışılması amacıyla yazılan Python komut satırları paylaşılmış, bu analizin sonuçları ise grafiksel olarak Şekil 4.14'te sunulmuştur. Burada SMOTE işlemi sonrasında "Female (Kadın)" müşterilerin sayısının arttığı görülmektedir dolayısıyla ham verilerde pazarlamaya olumlu dönüş yapan müşteriler büyük oranda 35-65 yaş aralığındaki kadınlar olduğundan algoritma yeni olumlu satışlar sentezlerken bu durumu kullanmıştır.

```
In [13]: plt.hist(erkek["Age"],bins=60,label="Male", histtype='step',linewidth=2,fill=True,color="Blue")
plt.hist(kadin["Age"],bins=60,label="Female",histtype='step',linewidth=2,fill=True,color="Red")
plt.hist(nb["Age"],bins=60,label="Non-binary",histtype='step',linewidth=2,fill=True,color="Orange")
plt.xlabel("Age")
plt.ylabel("Number of customers")
#plt.yscale("log")
plt.legend(loc="center right")
plt.title("Before the SMOTE")
plt.savefig("mus_say_age.eps")
plt.savefig("mus_say_age.png")
```

```
In [17]: plt.hist(akadin["Age"],bins=60,label="Female",histtype='step',linewidth=2,fill=True,color="Red")
plt.hist(aerkek["Age"],bins=60,label="Male", histtype='step',linewidth=2,fill=True,color="Blue")

plt.hist(anb["Age"],bins=60,label="Non-binary",histtype='step',linewidth=2,fill=True,color="Orange")
plt.xlabel("Age")
plt.ylabel("Number of customers")
#plt.yscale("log")
plt.legend(loc="center right")
plt.title("After the SMOTE")
plt.savefig("mus_say_age.eps")
plt.savefig("mus_say_age.png")
```

Şekil 4.13. SMOTE öncesi (üst) ve sonrası (alt) müşteri sayısının yaşa göre dağılımı kod satırları.



Şekil 4.14. SMOTE öncesi (soldaki) ve sonrası (sağdaki) müşteri sayısının yaşa göre dağılımı.

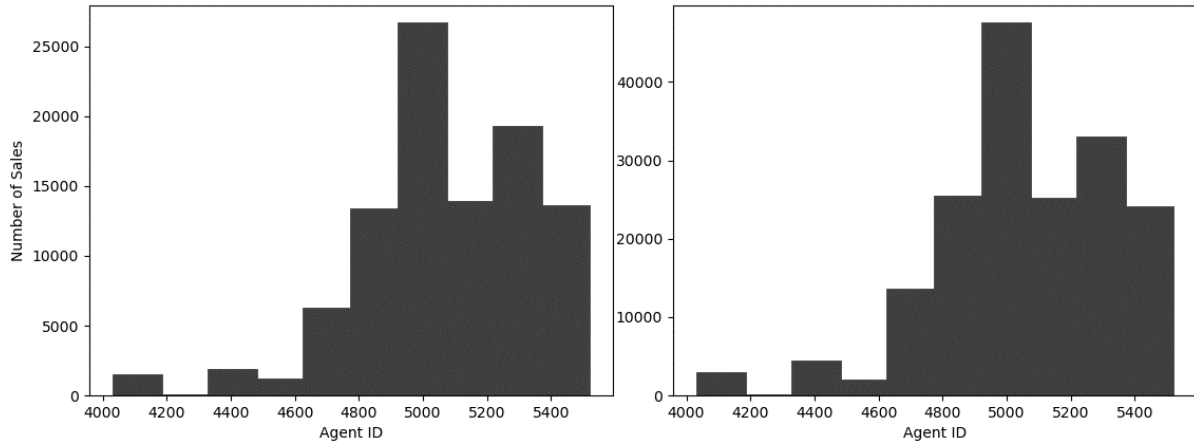
Bu aşamada önceki analizlerden görülen sonuçları hangi acentelerin daha başarılı satışlar yaptığı ve hangi ürünlerin daha çok satıldığı bilgileriyle birleştirmemiz karar destek aşamasında önemli katkılar sunacaktır. Pazarlama sayıları ile acenteler arasındaki ilişkiyi kontrol etmek amacıyla yazılan

kodlamalara Şekil 4.15'ten ulaşmak mümkündür. Bu kodlama satırlarından elde edilen nümerik sonuçların grafiksel gösterimleri ise Şekil 4.16'da sunulmuştur. Bu analiz sonuçlarından görüldüğü üzere bazı acenteler nispeten daha çok müşteriye ulaşmış yani aktif çalışmalar yürütmüştür.

```
In [20]: plt.hist(bdata["Agent_ID"], histtype='step',linewidth=2,fill=True,color="Green")
plt.title("Before the SMOTE")
plt.xlabel("Agent ID")
plt.ylabel("Number of Sales")
plt.savefig("Agent_ID.eps")
plt.savefig("Agent_ID.png")
```

```
In [25]: plt.hist(naveri["Agent_ID"], histtype='step',linewidth=2,fill=True,color="Green")
plt.title("After the SMOTE")
plt.xlabel("Agent ID")
plt.ylabel("Number of Sales")
plt.savefig("Agent_ID.eps")
plt.savefig("Agent_ID.png")
```

Şekil 4.15. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-acente dağılımı kod satırları.



Şekil 4.16. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-acente dağılımı.

```
In [26]: plt.hist(erkek["Agent_ID"],label="Male",histtype='step',linewidth=2,fill=True,color="Blue")
plt.hist(kadin["Agent_ID"],label="Female",histtype='step',linewidth=2,fill=True,color="Red")
plt.hist(nb["Agent_ID"],label="Non-binary",histtype='step',linewidth=2,fill=True,color="Orange")

plt.xlabel("Agent ID")
plt.ylabel("Number of Sales")
plt.title("Before the SMOTE")
plt.legend(loc="upper right")
plt.savefig("Agent_ID_Gen.eps")
plt.savefig("Agent_ID_Gen.png")
```

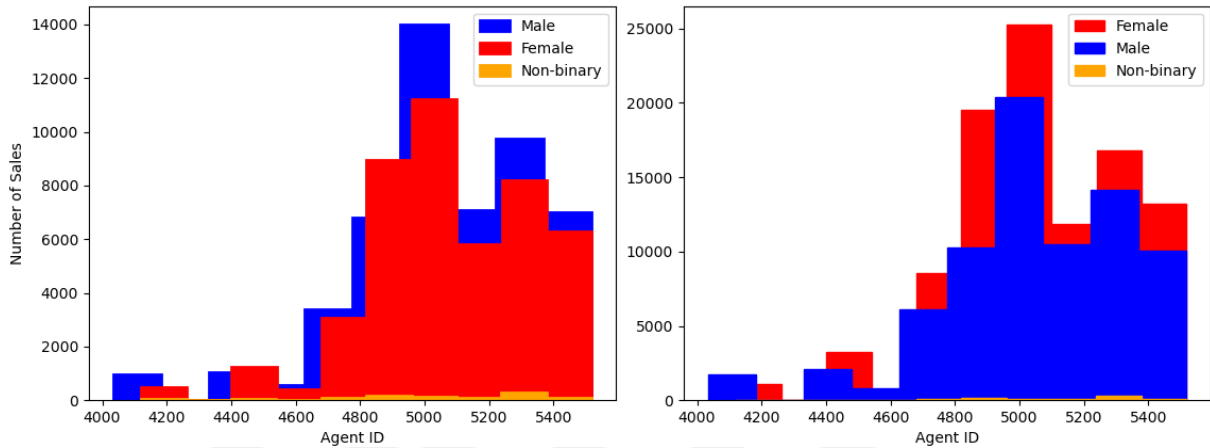
```
In [20]: #Agent Id
plt.hist(akadin["Agent_ID"].astype(int), histtype='step',linewidth=1,fill=True,label="Female",color="Red")
plt.hist(aerkek["Agent_ID"].astype(int), histtype='step',linewidth=1,fill=True,label="Male",color="Blue")
plt.hist(anb["Agent_ID"].astype(int), histtype='step',linewidth=1,fill=True,label="Non-binary",color="Orange")

plt.xlabel("Agent ID")
plt.ylabel("Number of Sales")

plt.legend(loc="upper right")
plt.title("After the SMOTE")
plt.savefig("sales_agent S.eps")
plt.savefig("sales_agent S.png")
```

Şekil 4.17. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-acente-cinsiyet dağılımı kod satırları.

Pazarlama sayısı acentelere göre ele alındığında bazı acentelerin diğerlerine göre daha yoğun çalıştığı anlaşılmıştır. Şimdi bu analizi daha derinleştirmek adına acentelerin toplam pazarlama sayılarını ilgili müşterilerin cinsiyetlerine ve pazarlamaların sonuçlarına göre tartışalım. Şekil 4.17 ve 4.18’de pazarlama sayılarının acentelere göre dağılışı ele alınırken durum müşterilerin cinsiyet dağılımlarına göre tartışılmıştır.



Şekil 4.18. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-acente-cinsiyet dağılımı.

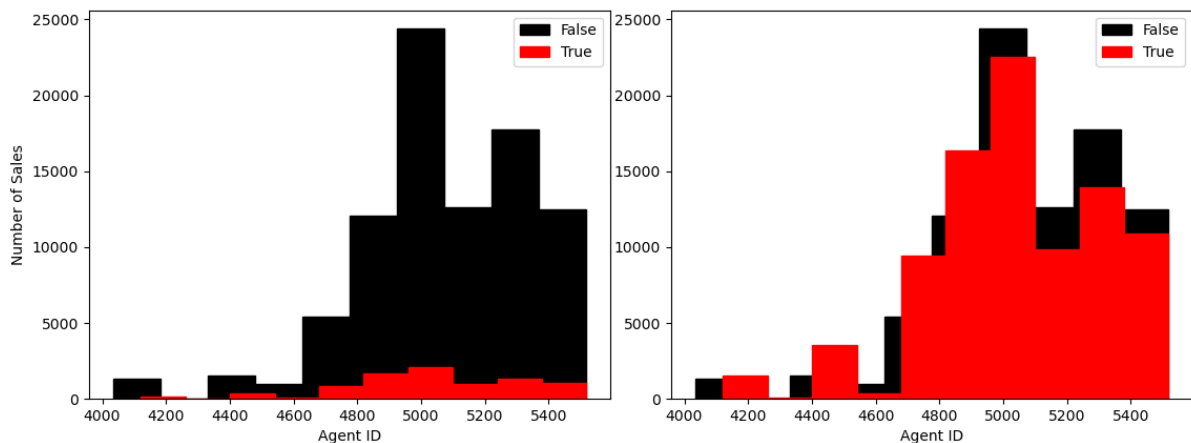
```
In [21]: plt.hist(ns["Agent_ID"],label="False",stacked=True, histtype='step',linewidth=1,fill=True,color="Black")
plt.hist(ps["Agent_ID"],label="True",stacked=True, histtype='step',linewidth=1,fill=True,color="Red")

plt.xlabel("Agent ID")
plt.ylabel("Number of Sales")
plt.title("Before the SMOTE")
plt.legend(loc="upper right")
plt.savefig("Agent_Sales_FT.eps")
plt.savefig("Agent_Sales_FT.png")
```

```
In [26]: plt.hist(ans["Agent_ID"],label="False",stacked=True, histtype='step',linewidth=1,fill=True,color="Black")
plt.hist(aps["Agent_ID"],label="True",stacked=True, histtype='step',linewidth=1,fill=True,color="Red")

plt.xlabel("Agent ID")
plt.ylabel("Number of Sales")
plt.title("After the SMOTE")
plt.legend(loc="upper right")
plt.savefig("Agent_Sales_FT.eps")
plt.savefig("Agent_Sales_FT.png")
```

Şekil 4.19. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-acente-sonuç dağılımı kod satırları.



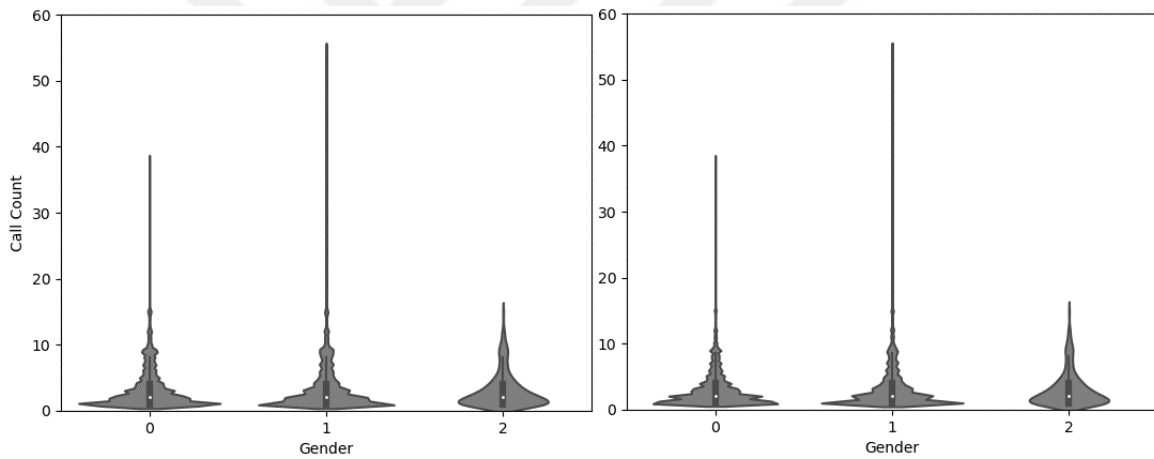
Şekil 4.20. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-acente-sonuç dağılımı.

Şekil 4.19 ve 4.20’de ise acentelerin pazarlama performansları olumlu ve olumsuz geri dönüşlere göre incelenmiştir. Bu analizlerden bazı acentelerin yoğun bir şekilde pazarlama performansı sergilemelerine rağmen olumsuz pazarlama geri dönüşlerinin de yine bu acentelerde daha sıklıkla görüldüğü anlaşılmıştır.

```
In [9]: sns.violinplot(data=bdata,y="Call_Count",x="Gender",palette=['red','blue','Orange'])
plt.ylim(0,60)
plt.ylabel("Call Count")
plt.title("Before the SMOTE")
plt.savefig("call_gen.eps")
plt.savefig("call_gen.png")
```

```
In [13]: sns.violinplot(data=naveri,y="Call_Count",x="Gender",palette=['red','blue','Orange'])
plt.ylim(0,60)
plt.ylabel("Call Count")
plt.title("After the SMOTE")
plt.savefig("call_gen.eps")
plt.savefig("call_gen.png")
```

Şekil 4.21. SMOTE öncesi (üst) ve sonrası (alt) arama-cinsiyet keman dağılımı komut satırları.

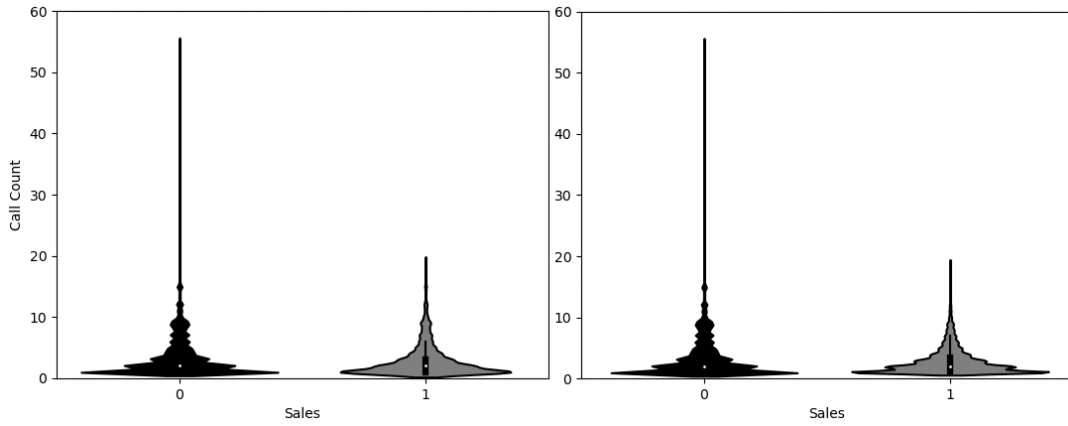


Şekil 4.22. SMOTE öncesi (sol) ve sonrası (sağ) arama-cinsiyet keman dağılımı.

```
In [10]: sns.violinplot(data=bdata,y="Call_Count",x="Sale",palette=['black','red'])
plt.ylim(0,60)
plt.ylabel("Call Count")
plt.xlabel("Sales")
plt.title("Before the SMOTE")
plt.savefig("call_sale.eps")
plt.savefig("call_sale.png")
```

```
In [14]: sns.violinplot(data=naveri,y="Call_Count",x="Sale",palette=['black','red'])
plt.ylim(0,60)
plt.ylabel("Call Count")
plt.xlabel("Sales")
plt.title("After the SMOTE")
plt.savefig("call_sale.eps")
plt.savefig("call_sale.png")
```

Şekil 4.23. SMOTE öncesi (üst) ve sonrası (alt) arama-sonuç keman dağılımı komut satırları.



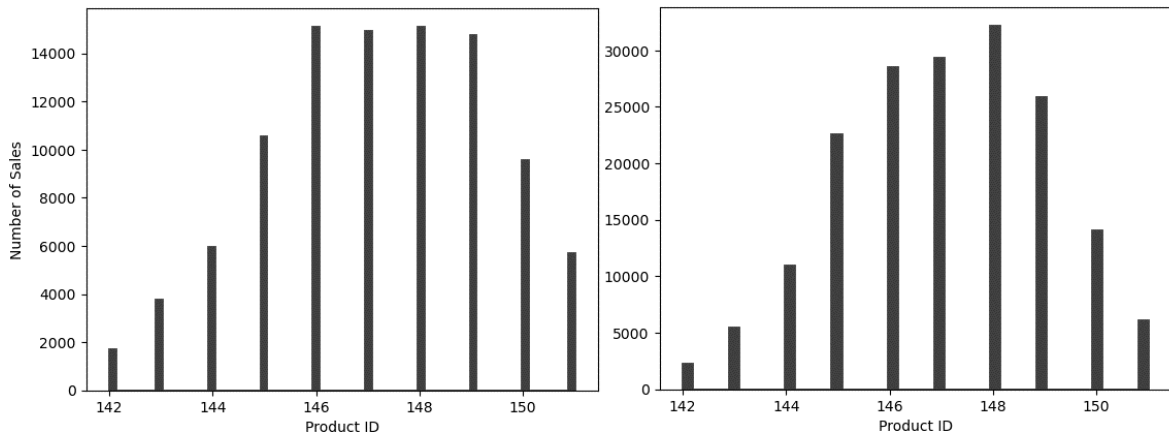
Şekil 4.24. SMOTE öncesi (sol) ve sonrası (sağ) arama-sonuç kerman dağılımı.

Kerman grafiği yöntemi kullanılarak Şekil 4.21 ve 4.22’de müşterilerin aranma sayılarının cinsiyetlerine göre dağılımının incelenmesine yönelik analizler görselleştirilirken Şekil 4.23 ve 4.24’te ise müşterilerin aranma sayılarının pazarlama geri dönüşlerine göre analizin basamakları betimlenmiştir. Görüldüğü üzere müşterilerin aranma sayıları çoğunlukla [1-10] aralığında olup aranma sayısındaki artış olumlu pazarlama dönüşü olarak sonuçlanmamaktadır.

```
In [16]: plt.hist(bdata["Product_ID"],bins=80, histtype='step',linewidth=2,fill=True,color="Green")
plt.title("Before the SMOTE")
plt.xlabel("Product ID")
plt.ylabel("Number of Sales")
plt.savefig("Product_ID.eps")
plt.savefig("Product_ID.png")

In [21]: plt.hist(naveri["Product_ID"].astype(int),bins=50, histtype='step',linewidth=2,fill=True,color="Green")
plt.title("After the SMOTE")
plt.xlabel("Product ID")
plt.ylabel("Number of Sales")
plt.savefig("Product_ID.eps")
plt.savefig("Product_ID.png")
```

Şekil 4.25. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-ürün dağılımı kod satırları.



Şekil 4.26. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-ürün dağılımı.

Ele alınan tele-pazarlama verilerini görselleştirerek onları daha başarılı biçimde anlamaya çalıştığımız araştırmamızın son basamağında pazarlama sayıları ile ürünler arasındaki ilişkiye

odaklanılmıştır. Şekil 4.25 ve 4.26’da bu kısmın ilk basamağı olarak katalogdaki ürünlerin pazarlanma sayıları ile olan ilişkileri dikkate alınmıştır. Görüldüğü üzere 146, 147, 148 ve 149 kod numaralı ürünler en fazla pazarlanan örneklem olarak ön plana çıkmaktadır. Kod numarası 146, 147, 148 ve 149 olan ürünlerin diğerlerine göre daha fazla sayıda pazarlandığı bilgisi değerlidir, ancak yeterli olmadığı aşıkardır. Bu analizin derinleştirilmesi yani ilgili ürünlerin pazarlanma durumlarının müşteri cinsiyetleri ve pazarlama sonuçlarına göre daha detaylı ele alınması faydalı olacaktır. Bu bağlamda ilgili analizler için okuyucunun dikkatini Şekil 4.27, 4.28, 4.29 ve 4.30’a sunulan araştırma sonuçlarımıza çekmek isteriz. Bunlardan ilk ikisi satılan ürünlerin pazarlanma sayılarını cinsiyet bağlamında ele alırken, sonraki ikisi ise satılan ürünlerin pazarlanma sayılarını işlem sonucuna göre değerlendirmektedir. Öncelikle bir noktaya dikkat çekmek yerinde olacaktır: yapılan SMOTE ile veri sentezleme işlemi sonrasında en fazla pazarlanan ürün çeşitleri değişmemiştir. Dolayısıyla araştırmamızın sonraki basamaklarında hatalı tahminler yürütülmesine neden olabilecek bir olumsuzluk yaşanmamış ve algoritma ham veri karakteristiği üzerinde hatalı kabul edilebilecek değişiklikler yapmamıştır. Bir diğer önemli çıkarım ise özellikle 147, 148 ve 149 kod numaralı ürünlerin kadın müşteriler tarafından satın alındığı görülmektedir, diğer bir deyişle olumlu (ya da true veya pozitif) pazarlama sonucu bağlamında bu ürünler dikkat çekmektedirler.

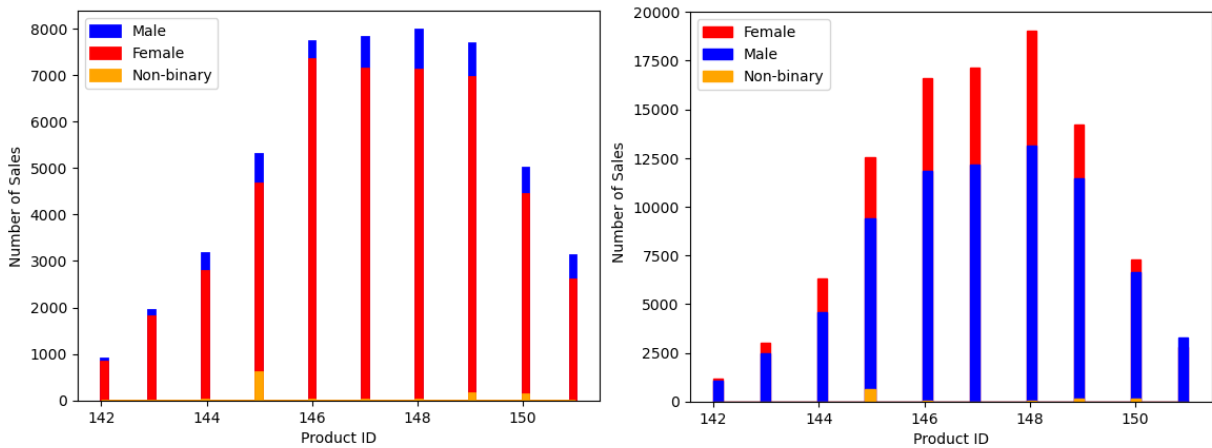
```

In [17]: plt.hist(erkek["Product_ID"],bins=80,label="Male", histtype='step',linewidth=2,fill=True,color="Blue")
plt.hist(kadin["Product_ID"],bins=80,label="Female", histtype='step',linewidth=2,fill=True,color="Red")
plt.hist(nb["Product_ID"],bins=80,label="Non-binary", histtype='step',linewidth=2,fill=True,color="Orange")
plt.title("Before the SMOTE")
plt.xlabel("Product ID")
plt.legend(loc="upper left")
plt.ylabel("Number of Sales")
plt.savefig("Product_ID_Gen.eps")
plt.savefig("Product_ID_Gen.png")

In [22]: plt.hist(akadin["Product_ID"].astype(int),bins=50, label="Female",stacked=True, histtype='step',linewidth=1,fill=True,color="Red")
plt.hist(aerkek["Product_ID"].astype(int),bins=50,label="Male",stacked=True, histtype='step',linewidth=1,fill=True,color="Blue")
plt.hist(anb["Product_ID"].astype(int),bins=50,label="Non-binary", stacked=True, histtype='step',linewidth=1,fill=True,color="Orange")
plt.title("After the SMOTE")
plt.xlabel("Product ID")
plt.yscale("Log")
plt.legend(loc="upper left")
plt.ylabel("Number of Sales")
plt.savefig("Product_ID_Gen.eps")
plt.savefig("Product_ID_Gen.png")

```

Şekil 4.27. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-ürün-cinsiyet dağılımı kodlar.



Şekil 4.28. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-ürün-cinsiyet dağılımı.

```

In [19]: plt.hist(ns["Product_ID"],bins=80,label="False", histtype='step',linewidth=2,fill=True,color="Black")
plt.hist(ps["Product_ID"],bins=80,label="True", histtype='step',linewidth=2,fill=True,color="Red")

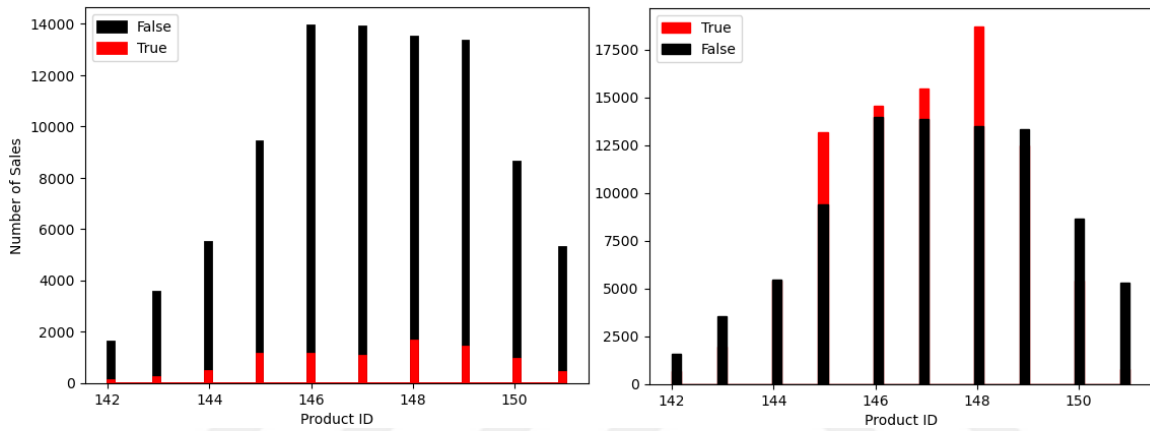
plt.title("Before the SMOTE")
plt.xlabel("Product ID")
plt.ylabel("Number of Sales")
plt.legend(loc="upper left")
plt.savefig("Product_ID_FT.eps")
plt.savefig("Product_ID_FT.png")

In [24]: plt.hist(aps["Product_ID"].astype(int),bins=50,label="True", stacked=True,histtype='step',linewidth=1,fill=True,color="Red")
plt.hist(ans["Product_ID"].astype(int),bins=50,label="False",stacked=True, histtype='step',linewidth=1,fill=True,color="Black")

plt.title("After the SMOTE")
plt.xlabel("Product ID")
plt.ylabel("Number of Sales")
plt.legend(loc="upper left")
plt.savefig("Product_ID_FT.eps")
plt.savefig("Product_ID_FT.jpg")

```

Şekil 4.29. SMOTE öncesi (üst) ve sonrası (alt) pazarlama-ürün-sonuç dağılımı için kodlar.



Şekil 4.30. SMOTE öncesi (sol) ve sonrası (sağ) pazarlama-ürün-sonuç dağılımı.

4.2. Algoritmayı Ayarla, Eğit ve Test Et!

Başarılı bir ML analizi için algoritmadaki hiper-parametreyi çok iyi ayarlanmamız ve uygun mimari bileşenlerinin dikkatlice seçmemiz gerekmektedir. Daha önce de vurgulandığı üzere bu tez araştırması kapsamında ele alınan tele-pazarlama veri kümesinde esasen “Sales (Satışlar)” örnekleme göre bir sınıflandırma yapılması hedeflenmiştir. Birinci bölümde kısa bilgilerde tanıttığımız ML algoritmalarından bazıları bu amaca uygundur, ancak hedeflenen analizde kullanılmak üzere sarmalayıcı yapılı bir kNN algoritması tercih edilmiştir. Bu ML yaklaşımında k hiper-parametresi öğrenme sürecini kontrol etmek için kullanıcı tarafından açıkça tanımlanması gereken hayati bir öğrenim etkenidir¹⁶⁶. Diğer yandan, algoritma parçalarının en doğru kombinasyonunu bularak en uygun öğrenme mimarisinin seçilmesi de gerekmektedir. Genel yöntem mevcut verilerin bir kısmının farklı algoritma seçimlerinde ayar grubu olarak kullanılmasıdır, dolayısıyla veri setinde eğitim-test bölünme oranı 70-30 olarak belirlenmiştir. Burada, [1,5] aralığındaki tüm k tamsayı değerleri, yedi farklı uzaklık fonksiyonu ve dört farklı sarmalayıcı tipi için mimariler çalıştırılmıştır. İlk olarak CityBlock (Manhattan) ölçüm fonksiyonu SBS, SBFS, SFFS ve SFS sarmalayıcıları ile birlikte ele alınmıştır (kod satırları için bakınız Şekil 4.31).

¹⁶⁶ Böylesi parametreler için kullanılan "hiper" ön eki aslında onun öğrenme yaklaşımının kontrolünde dikkate alınan en üst düzey parametre olduğuna işaret etmektedir.

```

In [ ]: #optimizasyon alanı
mm=[[False,False],[False,True],[True,True],[True,False]]

dataf={"SBS":[],"SBFS":[],"SFFS":[],"SFS":[]}
xx=["SBS","SBFS","SFFS","SFS"]
for j in range(4):
    for i in range(1,6):
        print(mm[j][0]," ",mm[j][1])
        # Build KNN classifier to use in feature selection
        knn_clf = KNeighborsClassifier(n_neighbors=i,metric="manhattan")
        # Build step feature selection
        k_sfs = sfs(knn_clf,
                    k_features=4,
                    forward= mm[j][0],
                    floating=mm[j][1],
                    verbose=2,
                    scoring='accuracy',
                    cv=5,
                    n_jobs=-1)

        # Perform SFFS
        k_sfs = k_sfs.fit(X_train, y_train)
        X_train_efs = k_sfs.transform(X_train)
        X_test_efs = k_sfs.transform(X_test)

        # Fit the estimator using the new feature subset
        # and make a prediction on the test data
        knn_clf.fit(X_train_efs, y_train)
        y_pred = knn_clf.predict(X_test_efs)

        # Compute the accuracy of the prediction
        acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
        # print('Test set accuracy: %.2f %%' % (acc*100))
        dataf[xx[j]].append(acc)
        feat_cols = list(k_sfs.k_feature_idx_)
        print(feat_cols)

        cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
        print(cnf_matrix)

```

Şekil 4.31. CityBlock ölçüm modelinin performans analizi kod satırları.

```

In [17]: #tune alanı
fig ,axs=plt.subplots(2,2)
fig.set_figwidth(9)
axs[0,0].scatter(range(1,6),dataf["SBS"])
axs[0,0].plot(range(1,6),dataf["SBS"],label="SBS",linestyle='dashed',color="Orange")
axs[0,0].legend(loc="upper right")
axs[0,0].set_ylabel("Accuracy")

axs[0,1].scatter(range(1,6),dataf["SBFS"])
axs[0,1].plot(range(1,6),dataf["SBFS"],label="SBFS",linestyle='dashed',color="Green")
axs[0,1].legend(loc="upper right")

axs[1,0].scatter(range(1,6),dataf["SFS"])
axs[1,0].plot(range(1,6),dataf["SFS"],label="SFS",linestyle='dashed',color="Purple")
axs[1,0].set_ylabel("Accuracy")
axs[1,0].set_xlabel("k")
axs[1,0].legend(loc="upper right")

axs[1,1].scatter(range(1,6),dataf["SFFS"])
axs[1,1].plot(range(1,6),dataf["SFFS"],label="SFFS",linestyle='dashed',color="Gray")
axs[1,1].set_xlabel("k")
axs[1,1].legend(loc="upper right")

plt.savefig("cityblock_Tune.eps")
plt.savefig("cityblock_Tune.png")

```

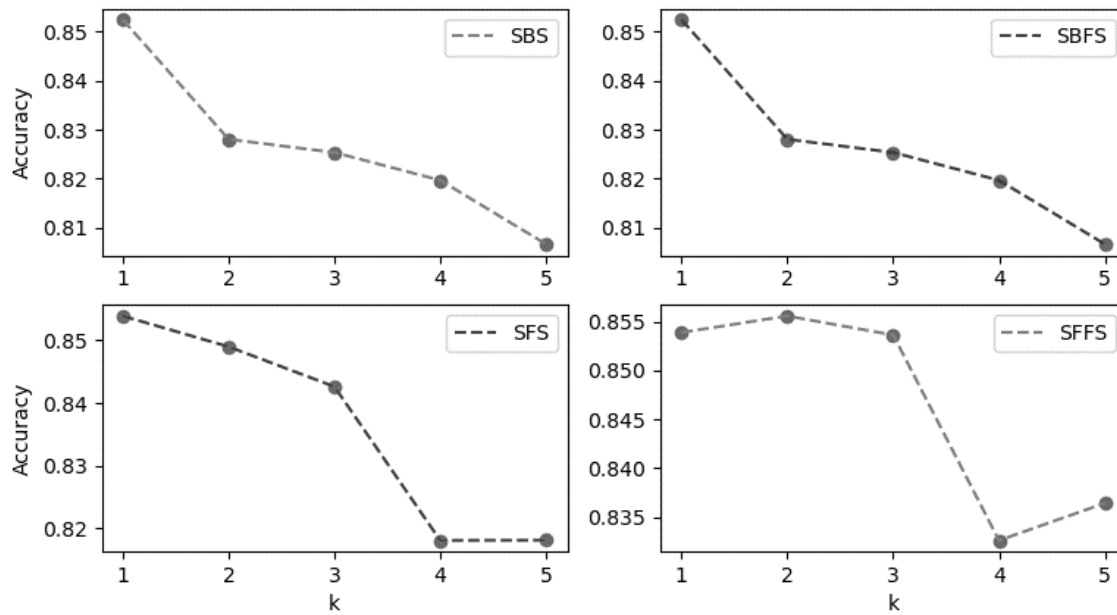
Şekil 4.32. CityBlock modeli performans analizi grafik kodları.

Yaklaşık 13 dakika süren bu analizin sonuçlarını görebilmek için bir dizi yeni komut satırlarına (bakınız Şekil 4.32 ve 4.33) ihtiyaç duyulmaktadır. İlgili kodların kullanılmasıyla Şekil 4.34 ve Tablo 4.4'te verilen çıktılar elde edilmiştir.

```
In [18]: #ACC sayısal sonuçlar
sonuc=pd.DataFrame(dataf)

sonuc.head(10)
```

Şekil 4.33. CityBlock modeli performans analizi nümerik çıktı kodları.



Şekil 4.34. CityBlock ölçüm modelinin performans analizi.

Tablo 4.4. CityBlock modeli analizinde elde edilen tutarlılık oranları.

k	SBS	SBFS	SFS	SFFS
1	0.852453	0.852453	0.853901	0.853901
2	0.828050	0.828050	0.848994	0.855593
3	0.825362	0.825362	0.842621	0.853638
4	0.819684	0.819684	0.818067	0.832581
5	0.806655	0.806655	0.818105	0.836398

En yüksek aküresi değeri kalın puntolu yazıyla Tablo 4.4'te vurgulanmıştır. Görüldüğü üzere SFFS tipi bir sarmalayıcı içeren kNN algoritmasında en uygun hiper-parametre değeri en yüksek aküresi olan 0.855593 değerini veren $k=2$ seçimidir. Benzer komut satırları ve aynı analiz basamakları daha önce kullanılacağı vurgulanan diğer mesafe ölçüm fonksiyonları Chebyshev, Euclidian, Correlation (ya da Korelasyon), Canberra, Dice ve Cosine (ya da Kosinüs) için de tekrarlanmalı ve nihai sonuçlar karşılaştırılmalıdır.

```

In [ ]: #optimizasyon alanı
mm=[[False,False],[False,True],[True,True],[True,False]]

dataf={"SBS":[],"SBFS":[],"SFFS":[],"SFS":[]}
xx=["SBS","SBFS","SFFS","SFS"]
for j in range(4):
    for i in range(1,6):
        print(mm[j][0]," ",mm[j][1])
        # Build KNN classifier to use in feature selection
        knn_clf = KNeighborsClassifier(n_neighbors=i,metric="chebyshev")
        # Build step feature selection
        k_sfs = sfs(knn_clf,
                    k_features=4,
                    forward= mm[j][0],
                    floating=mm[j][1],
                    verbose=2,
                    scoring='accuracy',
                    cv=5,
                    n_jobs=-1)

        # Perform SFFS
        k_sfs = k_sfs.fit(X_train, y_train)
        X_train_efs = k_sfs.transform(X_train)
        X_test_efs = k_sfs.transform(X_test)

        # Fit the estimator using the new feature subset
        # and make a prediction on the test data
        knn_clf.fit(X_train_efs, y_train)
        y_pred = knn_clf.predict(X_test_efs)

        # Compute the accuracy of the prediction
        acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
        # print('Test set accuracy: %.2f %%' % (acc*100))
        dataf[xx[j]].append(acc)
        feat_cols = list(k_sfs.k_feature_idx_)
        print(feat_cols)

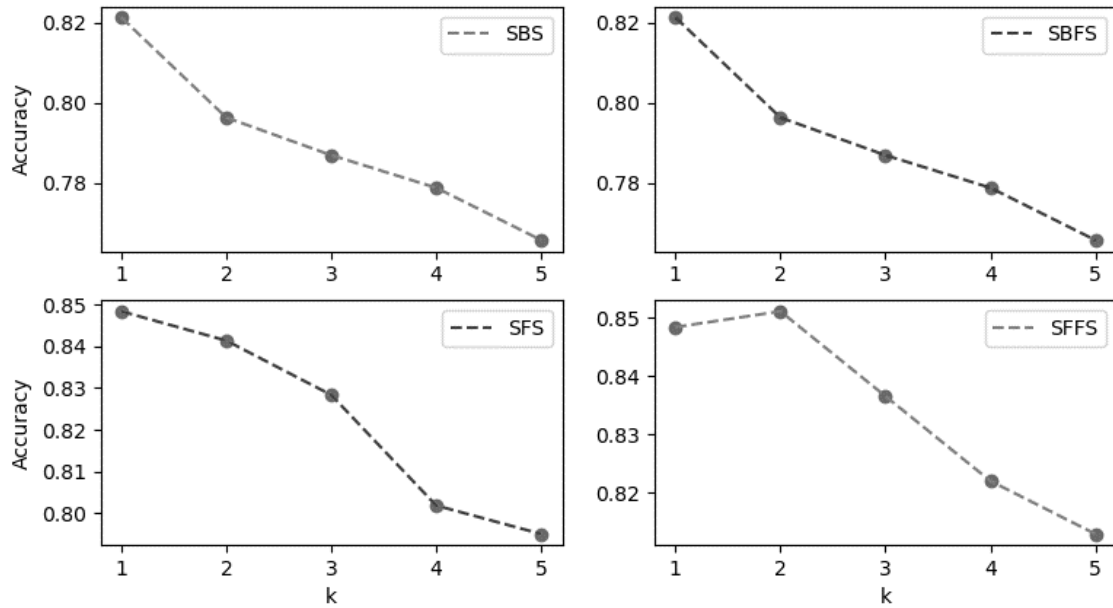
        cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
        print(cnf_matrix)

```

Şekil 4.35. Chebyshev ölçüm modelinin performans analizi kod satırları.

Chebyshev mesafe ölçüm modeliyle birlikte SBS, SBFS, SFFS ve SFS sarmalayılarının kullanıldığı algoritma ayarlama işlemi (bakınız Şekil 4.35) yaklaşık 60 dakika sürmüştür. Şekil 4.32 ve 4.33'te verilen algoritma kodları Chebyshev ölçüm yaklaşımı için de aynen kullanılabilir¹⁶⁷, böylece Şekil 4.36 ve Tablo 4.5'te sunulan çıktıları ulaşılmıştır. Analiz sonucunda en yüksek aküresi değeri 0.851100 (Tablo 4.5'te kalın punto ile yazılan veri) olarak SFFS tipi sarmalayıcıda $k=2$ hiperparametre değerinde görülmüştür.

¹⁶⁷ Sadece Şekil 4.32'de son iki komut satırında bilgisayardan kaydedilmek üzere çıktıların "eps" ve "png" olarak verilmesi istenmiştir. Burada dosyaya modeli hatırlatan bir isim verilebilir.

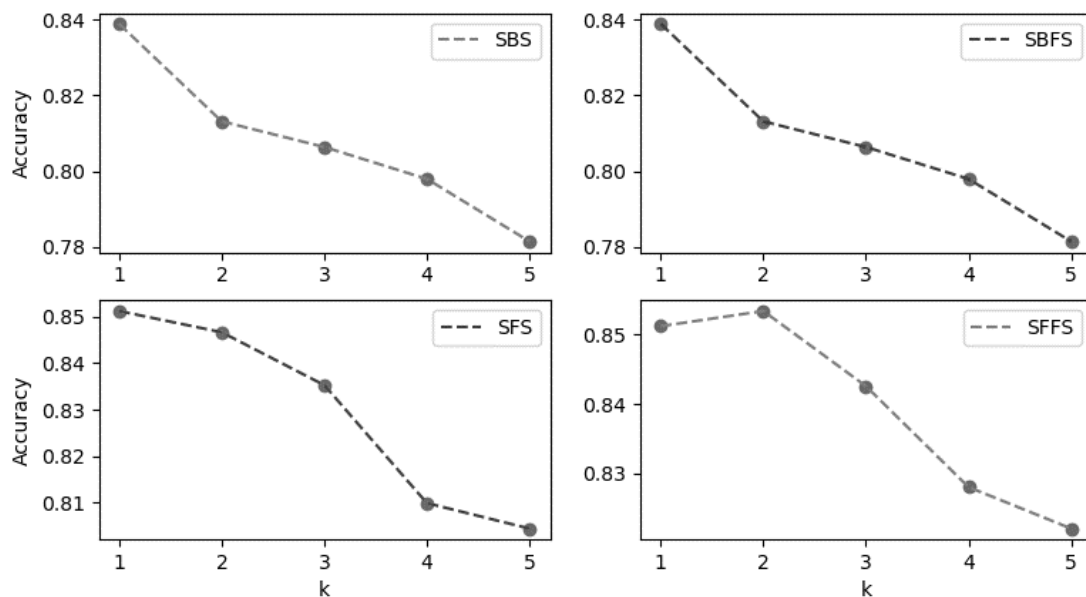


Şekil 4.36. Chebyshev ölçüm modelinin performans analizi.

Tablo 4.5. Chebyshev yaklaşımı analizinde elde edilen tutarlılık oranları.

k	SBS	SBFS	SFS	SFFS
1	0.821357	0.821357	0.848393	0.848393
2	0.796353	0.796353	0.841380	0.851100
3	0.786952	0.786952	0.828351	0.836529
4	0.778812	0.778812	0.801842	0.821978
5	0.765745	0.765745	0.795018	0.812878

Euclidian ölçüm yaklaşımı için algoritma ayarlama kodları tekrarlandığında analiz süresi 45 dakika olmuştur ve en yüksek aküresi değeri olan 0.855593 sayısı SFFS tipi sarmalayıcıyla $k=2$ değerinde elde edilmiştir (Analizin grafiksel sonuçları, nümerik verileri ve kodlama satırları için bakınız Şekil 4.37, Tablo 4.6 ve Şekil 4.38).



Şekil 4.37. Euclidean ölçüm modelinin performans analizi.

Tablo 4.6. Euclidean ölçüm modeli analizinde elde edilen tutarlılık oranları.

k	SBS	SBFS	SFS	SFFS
1	0.838785	0.838785	0.851213	0.851213
2	0.813066	0.813066	0.846644	0.853393
3	0.806317	0.806317	0.835270	0.842564
4	0.797894	0.797894	0.809889	0.828013
5	0.781425	0.781425	0.804418	0.822034

```

In [ ]: #optimizasyon alanı
mm=[[False,False],[False,True],[True,True],[True,False]]

dataf={"SBS":[],"SBFS":[],"SFFS":[],"SFS":[]}
xx=["SBS","SBFS","SFFS","SFS"]
for j in range(4):
    for i in range(1,6):
        print(mm[j][0]," ",mm[j][1])
        # Build KNN classifier to use in feature selection
        knn_clf = KNeighborsClassifier(n_neighbors=i,metric="euclidean")
        # Build step feature selection
        k_sfs = sfs(knn_clf,
                    k_features=4,
                    forward= mm[j][0],
                    floating=mm[j][1],
                    verbose=2,
                    scoring='accuracy',
                    cv=5,
                    n_jobs=-1)

        # Perform SFFS
        k_sfs = k_sfs.fit(X_train, y_train)
        X_train_efs = k_sfs.transform(X_train)
        X_test_efs = k_sfs.transform(X_test)

        # Fit the estimator using the new feature subset
        # and make a prediction on the test data
        knn_clf.fit(X_train_efs, y_train)
        y_pred = knn_clf.predict(X_test_efs)

        # Compute the accuracy of the prediction
        acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
        # print('Test set accuracy: %.2f %%' % (acc*100))
        dataf[xx[j]].append(acc)
        feat_cols = list(k_sfs.k_feature_idx_)
        print(feat_cols)

        cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
        print(cnf_matrix)

```

Şekil 4.38. Euclidean ölçüm modelinin performans analizi kod satırları.

Correlation mesafe ölçümüne göre analiz yapıldığında (bakınız Şekil 4.39) diğer modellere göre daha uzun süren bir durum oluşmuş ve 7 saat sonunda SFFS tipi sarmalayıcıyla 0.875108 olan en yüksek aküresi değeri $k=1$ hiper-parametre değerinde görülmüştür (bakınız Şekil 4.40 ve Tablo 4.7).

```

In [ ]: #optimizasyon alanı
mm=[[False,False],[False,True],[True,True],[True,False]]

dataf={"SBS":[],"SBFS":[],"SFFS":[],"SFS":[]}
xx=["SBS","SBFS","SFFS","SFS"]
for j in range(4):
    for i in range(1,6):
        print(mm[j][0]," ",mm[j][1])
        # Build KNN classifier to use in feature selection
        knn_clf = KNeighborsClassifier(n_neighbors=i,metric="correlation")
        # Build step feature selection
        k_sfs = sfs(knn_clf,
                    k_features=4,
                    forward= mm[j][0],
                    floating=mm[j][1],
                    verbose=2,
                    scoring='accuracy',
                    cv=5,
                    n_jobs=-1)

        # Perform SFFS
        k_sfs = k_sfs.fit(X_train, y_train)
        X_train_efs = k_sfs.transform(X_train)
        X_test_efs = k_sfs.transform(X_test)

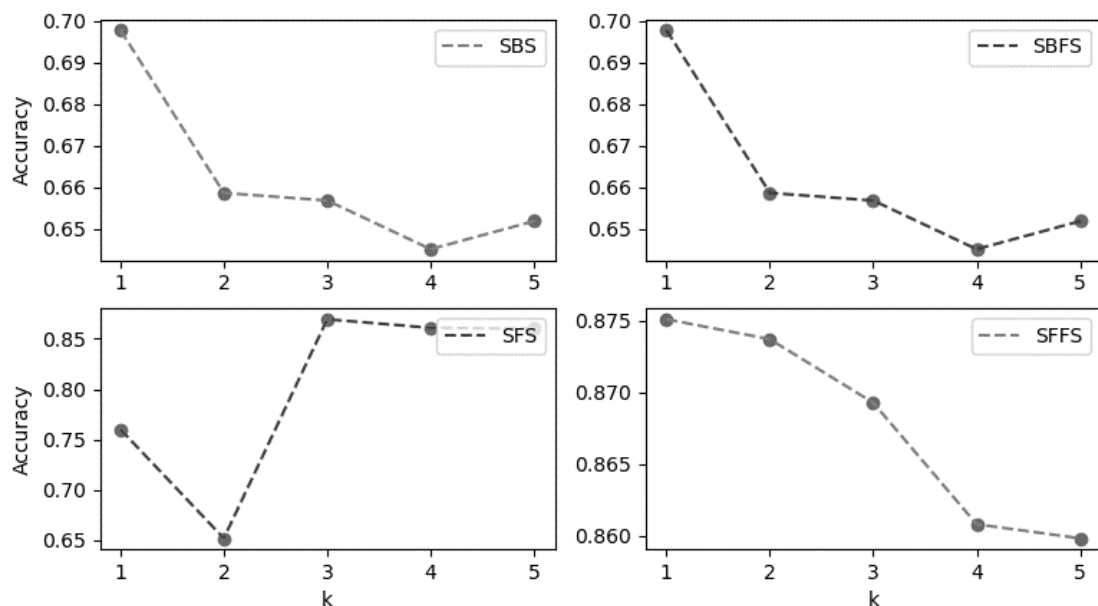
        # Fit the estimator using the new feature subset
        # and make a prediction on the test data
        knn_clf.fit(X_train_efs, y_train)
        y_pred = knn_clf.predict(X_test_efs)

        # Compute the accuracy of the prediction
        acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
        # print('Test set accuracy: %.2f %%' % (acc*100))
        dataf[xx[j]].append(acc)
        feat_cols = list(k_sfs.k_feature_idx_)
        print(feat_cols)

        cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
        print(cnf_matrix)

```

Şekil 4.39. Correlation ölçüm modelinin performans analizi kod satırları.



Şekil 4.40. Correlation ölçüm modelinin performans analizi.

Tablo 4.7. Correlation modeli analizinde elde edilen tutarlılık oranları.

<i>k</i>	SBS	SBFS	SFS	SFFS
1	0.697763	0.697763	0.760068	0.875108
2	0.658564	0.658564	0.652002	0.873717
3	0.656721	0.656721	0.869280	0.869280
4	0.645008	0.645008	0.860801	0.860801
5	0.651758	0.651758	0.859823	0.859823

Canberra modeliyle çalışıldığında (Python kodlarına Şekil 4.41'den ulaşmak mümkündür) yaklaşık 9 saat süren araştırma SBS, SBFS ve SFFS tipi sarmalayıcıların en yüksek akürezı değeri olan 0.915304 sayısına ulaştığını göstermiştir (bakınız Şekil 4.42 ve Tablo 4.8). Görüldüğü üzere üç sarmalayıcı tipi $k=2$ değerini en uygun hiper-parametre seçimi olarak önermektedir.

```
In [ ]: #optimizasyon alanı
mm=[[False,False],[False,True],[True,True],[True,False]]

dataf={"SBS":[],"SBFS":[],"SFFS":[],"SFS":[]}
xx=["SBS","SBFS","SFFS","SFS"]
for j in range(4):
    for i in range(1,6):
        print(mm[j][0], " ",mm[j][1])
        # Build KNN classifier to use in feature selection
        knn_clf = KNeighborsClassifier(n_neighbors=i,metric="canberra")
        # Build step feature selection
        k_sfs = sfs(knn_clf,
                    k_features=4,
                    forward= mm[j][0],
                    floating=mm[j][1],
                    verbose=2,
                    scoring='accuracy',
                    cv=5,
                    n_jobs=-1)

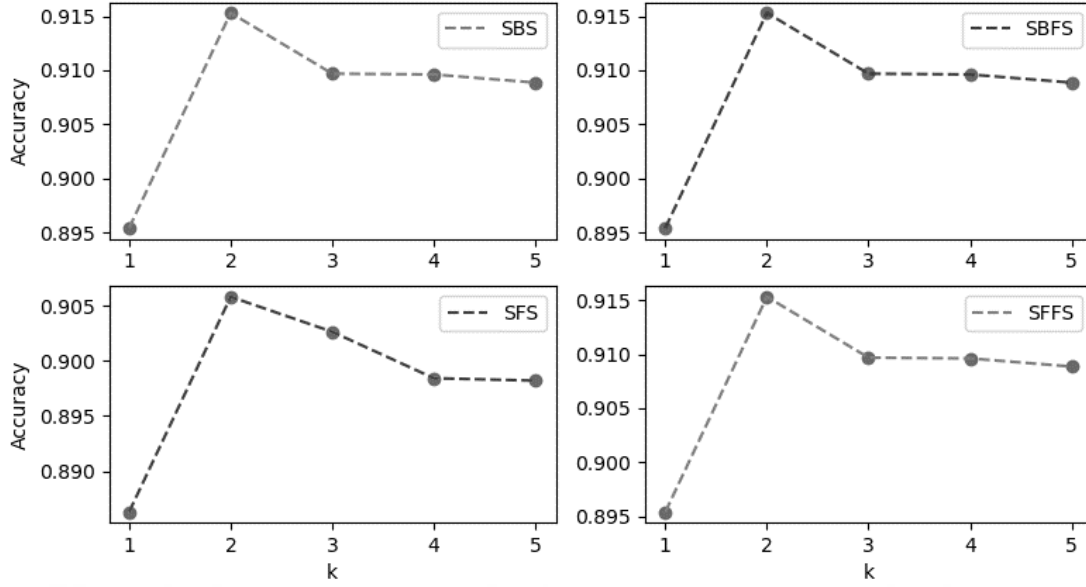
        # Perform SFFS
        k_sfs = k_sfs.fit(X_train, y_train)
        X_train_efs = k_sfs.transform(X_train)
        X_test_efs = k_sfs.transform(X_test)

        # Fit the estimator using the new feature subset
        # and make a prediction on the test data
        knn_clf.fit(X_train_efs, y_train)
        y_pred = knn_clf.predict(X_test_efs)

        # Compute the accuracy of the prediction
        acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
        # print('Test set accuracy: %.2f %%' % (acc*100))
        dataf[xx[j]].append(acc)
        feat_cols = list(k_sfs.k_feature_idx_)
        print(feat_cols)

    cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
    print(cnf_matrix)
```

Şekil 4.41. Canberra ölçümünün performans analizi için Python kodları.



Şekil 4.42. Canberra ölçüm modelinin performans analizi.

Tablo 4.8. Canberra ölçümünün analizinde elde edilen tutarlılık oranları.

k	SBS	SBFS	SFS	SFFS
1	0.895413	0.895413	0.886294	0.895413
2	0.915304	0.915304	0.905828	0.915304
3	0.909682	0.909682	0.902651	0.909682
4	0.909607	0.909607	0.898440	0.909607
5	0.908855	0.908855	0.898214	0.908855

Dice mesafe ölçüm yöntemi ayarlama evresinde (algoritma ayar kodları için bakınız Şekil 4.43) dikkat çeken farklı bir davranış sergilemiş ve yaklaşık olarak 11 saatlik analizin sonucunda iki farklı mimari durumunu önermiştir. En yüksek aküresi değeri 0.625738 olarak ölçülmüş, ancak bu değer farklı durumları öne çıkarmıştır (bakınız Şekil 4.44 ve Tablo 4.9). Görüldüğü üzere Dice yaklaşımı ile çalışılırken $k=1$ değeri seçilecekse SFS ya da SFFS tipi sarmalayıcıların kullanılması gerekirken $k=5$ değeri kullanıldığında dört sarmalayıcı arasında fark olmadığı (yani seçim serbestliği durumunun söz konusu olduğu) anlaşılmaktadır. Çalışmamızın bundan sonraki kısımlarında Dice modelinin yukarıda sıralan farklı davranış durumlarına dikkat çekecek biçimde Dice_vk1¹⁶⁸ ve Dice_vk5¹⁶⁹ model tanımları kullanılacaktır. Diğer yandan, Dice yaklaşımın tutarlılığı diğer modellere göre oldukça düşük oranda kalmış ve yaklaşım çok uzun analiz süresi gerektirmiştir. Analiz süresinin uzaması bu modelin güçlü bilgisayar donanımı gerektirdiğine de işaret etmektedir. Yani elimizde yeterince güçlü bilgisayar altyapısı ya da araştırmaya ayıracak uzun analiz süresi yoksa bu mesafe ölçümünü kullanmak külfet yaratmaktadır. Bu modelin ayarlama analizi bize her mesafe ölçüm yönteminin araştırmalar için aynı (ya da benzer) karakter sergileyemeyeceği gerçeğini açıkça göstermiş ve mimari ayarlama evresinin öğrenme analizinde ne kadar önemli olduğunu ortaya koymuştur.

¹⁶⁸ $k=1$ için Dice yaklaşımı.¹⁶⁹ $k=5$ için Dice modeli.

```

In [ ]: #optimizasyon alanı
mm=[[False,False],[False,True],[True,True],[True,False]]

dataf={"SBS":[],"SBFS":[],"SFFS":[],"SFS":[]}
xx=["SBS","SBFS","SFFS","SFS"]
for j in range(4):
    for i in range(1,6):
        print(mm[j][0]," ",mm[j][1])
        # Build KNN classifier to use in feature selection
        knn_clf = KNeighborsClassifier(n_neighbors=i,metric="dice")
        # Build step feature selection
        k_sfs = sfs(knn_clf,
                    k_features=4,
                    forward= mm[j][0],
                    floating=mm[j][1],
                    verbose=2,
                    scoring='accuracy',
                    cv=5,
                    n_jobs=-1)

        # Perform SFFS
        k_sfs = k_sfs.fit(X_train, y_train)
        X_train_efs = k_sfs.transform(X_train)
        X_test_efs = k_sfs.transform(X_test)

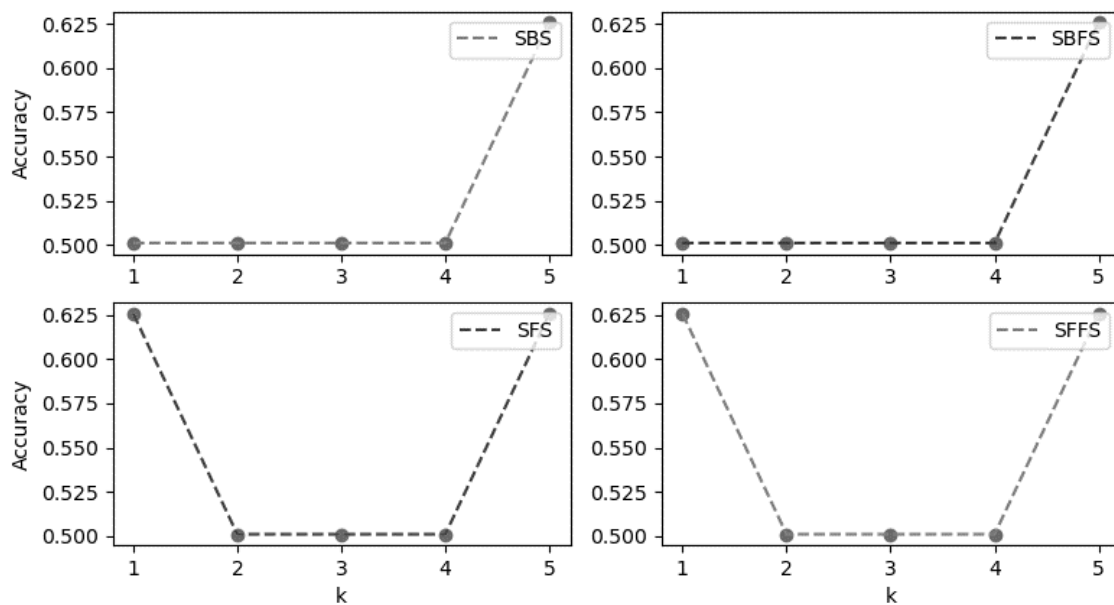
        # Fit the estimator using the new feature subset
        # and make a prediction on the test data
        knn_clf.fit(X_train_efs, y_train)
        y_pred = knn_clf.predict(X_test_efs)

        # Compute the accuracy of the prediction
        acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
        # print('Test set accuracy: %.2f %%' % (acc*100))
        dataf[xx[j]].append(acc)
        feat_cols = list(k_sfs.k_feature_idx_)
        print(feat_cols)

    cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
    print(cnf_matrix)

```

Şekil 4.43. Dice ölçüm modelinin performans analizi Python kodları.



Şekil 4.44. Dice ölçüm modelinin performans analizi.

Tablo 4.9. Dice ölçümü için elde edilen tutarlılık oranları.

<i>k</i>	SBS	SBFS	SFS	SFFS
1	0.501072	0.501072	0.625738	0.625738
2	0.501072	0.501072	0.501072	0.501072
3	0.501072	0.501072	0.501072	0.501072
4	0.501072	0.501072	0.501072	0.501072
5	0.625738	0.625738	0.625738	0.625738

```

In [ ]: #optimizasyon alanı
mm=[[False,False],[False,True],[True,True],[True,False]]

dataf={"SBS":[],"SBFS":[],"SFFS":[],"SFS":[]}
xx=["SBS","SBFS","SFFS","SFS"]
for j in range(4):
    for i in range(1,6):
        print(mm[j][0]," ",mm[j][1])
        # Build KNN classifier to use in feature selection
        knn_clf = KNeighborsClassifier(n_neighbors=i,metric="cosine")
        # Build step feature selection
        k_sfs = sfs(knn_clf,
                    k_features=4,
                    forward= mm[j][0],
                    floating=mm[j][1],
                    verbose=2,
                    scoring='accuracy',
                    cv=5,
                    n_jobs=-1)

        # Perform SFFS
        k_sfs = k_sfs.fit(X_train, y_train)
        X_train_efs = k_sfs.transform(X_train)
        X_test_efs = k_sfs.transform(X_test)

        # Fit the estimator using the new feature subset
        # and make a prediction on the test data
        knn_clf.fit(X_train_efs, y_train)
        y_pred = knn_clf.predict(X_test_efs)

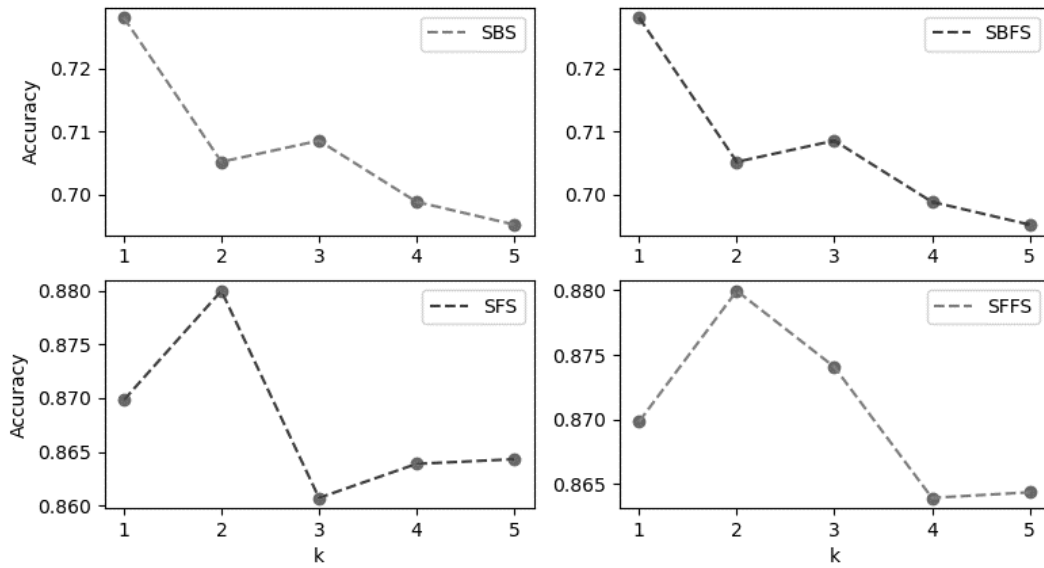
        # Compute the accuracy of the prediction
        acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
        # print('Test set accuracy: %.2f %%' % (acc*100))
        dataf[xx[j]].append(acc)
        feat_cols = list(k_sfs.k_feature_idx_)
        print(feat_cols)

        cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
        print(cnf_matrix)

```

Şekil 4.45. Cosine yaklaşımın performans analizi komut satırları.

Son olarak, Cosine mesafe ölçüm modelini baz alan ve yaklaşık olarak 10 saat süren analiz (kodlama satırları Şekil 4.45'te verilmiştir) SFS ve SFFS sarmalayıcı tiplerini en yüksek akürezî değeri olan 0.879959 ile ön plana çıkarmış ve en uygun hiper-parametre değerinin bu durumda $k=2$ olması gerektiğini göstermiştir. Bu analizin grafiksel sonuçları Şekil 4.46'da nümerik verileri ise Tablo 4.10'da sunulmuştur.



Şekil 4.46. Cosine ölçüm modelinin performans analizi.

Tablo 4.10. Cosine modelinde elde edilen tutarlılık oranları.

k	SBS	SBFS	SFS	SFFS
1	0.728069	0.728069	0.869825	0.869825
2	0.705170	0.705170	0.879959	0.879959
3	0.708535	0.708535	0.860745	0.874055
4	0.698816	0.698816	0.863903	0.863903
5	0.695206	0.695206	0.864335	0.864335

Araştırmamızın bu noktasında ayarlama evresinin sonuçlarını özetleyecek bir tablo sunmak resmin tamamının görülmesinde kolaylık sağlayacaktır. Bu bağlamda, Tablo 4.11'den ayarlama analizlerinin önemli sonuçlarına ulaşmak mümkündür.

Tablo 4.11: kNN algoritmasında ayarlama evresinin önemli sonuçları

Mesafe Tipi	k	Sarmalayıcı	Akürezi (%)	Süre (≈)
CityBlock	2	SFFS	85.6	13 Dakika
Chebyshev	2	SFFS	85.1	60 Dakika
Euclidean	2	SFFS	85.3	45 Dakika
Correlation	1	SFFS	87.5	7 saat
Canberra	2	SBS, SBFS, SFFS	91.6	9 Saat
Dice	1	SFS, SFFS	62.6	11 Saat
Dice	5	SBS, SBFS, SFS, SFFS	62.6	11 Saat
Cosine	2	SFS, SFFS	87.9	10 Saat

Algoritma bileşenlerinin en uygun kombinasyonu ve k hiper-parametresinin en iyi değeri her bir mimari kombinasyonu için artık açıkça ortaya çıkmıştır. Buradaki verilerden çeşitli mesafe ölçüm ifadelerinin performansları arasındaki en önemli farkların maksimum tutarlılık oranı ve gereken işlem süresi olduğu görülmektedir. Correlation, Canberra ve Cosine formüllerinin nispeten daha yüksek tahmin doğruluğu sunsa da oldukça uzun bir analiz süresi ve güçlü bilgisayar donanımı gerektirdikleri anlaşılmaktadır. Diğer yandan, bütün algoritma kombinasyonları bir noktada aynı duruma işaret etmektedir: *SFFS tipi sarmalayıcı her durumda en başarılı tutarlılığı sağlayacaktır!* Bu önemli bir çıkarımdır, fakat

yetersizdir. Öğrenme analizi için artık yorucu ve uzun hesaplama kısımları bitmiş, son adım kalmıştır. Ayarlama evresi ana sonuçlarının yardımıyla tasarlanan sarmalayıcı kNN kombinasyonları (Tablo 4.11’de paylaşılan 9 farklı öğrenme mimarisi) kullanılarak ele alınan tele-pazarlama veri seti için en başarılı algoritma bir öğrenme simülasyonu ile belirlenebilecektir.

4.3. Nihai Algoritmayı Tasarla ve Çalıştır!

Bir önceki bölümde en uygun mimari bileşenleri ve en faydalı hiper-parametre değerine odaklanılmış ve analizler sonrasında ortaya 9 farklı tutarlı sarmalayıcı tipi kNN mimarisi çıkmıştır. Araştırmamızın bu kısmı öğrenme simülasyonlarına ayrılmış ve algoritmadan dengelenen tele-pazarlama verilerini taradıktan sonra “Sales (Satışlar)” özneteliğini etkileyen en önemli faktörleri belirlemesi beklenmiştir. Seçilen veri setinde öğrenme-öngörme bölünme oranı algoritma ayarlama evresinde olduğu gibi yine 70-30’dur ve kNN mimarisinin öznetelik seçim yöntemi olarak SFFS tipi sarmalayıcı kullanılmasına karar verilmiştir. Sarmalayıcı tipi öznetelik seçimi içeren kNN algoritmasında en yakın komşuluğun CityBlock mesafe ölçüm modeline göre belirlenmesi istendiğinde en uygun hiper-parametre değeri Tablo 4.11’den görüldüğü üzere $k=2$ olmalıdır¹⁷⁰. Bu bağlamda inşa edilen algoritmadan öğrenme ve sonrasında tahmin etme analizlerini tamamlayıp hesapladığı karmaşıklık matrisini ve belirlediği etken faktörleri vermesi istenmiştir (yaklaşık 1 dakika süren analizin kodlama satırları için bakınız Şekil 4.47).

```
In [17]: #similasyon alanı

knn_clf = KNeighborsClassifier(n_neighbors=2,metric="manhattan")
# Build step feature selection
k_sfs = sfs(knn_clf,
            k_features=(1,6),
            forward= True,
            floating=True,
            verbose=2,
            scoring='accuracy',
            cv=5,
            n_jobs=-1)

# Perform SFFS
k_sfs = k_sfs.fit(X_train, y_train)
X_train_efs = k_sfs.transform(X_train)
X_test_efs = k_sfs.transform(X_test)

# Fit the estimator using the new feature subset
# and make a prediction on the test data
knn_clf.fit(X_train_efs, y_train)
y_pred = knn_clf.predict(X_test_efs)

# Compute the accuracy of the prediction
acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
# print('Test set accuracy: %.2f %%' % (acc*100))

feat_cols = list(k_sfs.k_feature_idx_)
print(feat_cols)

cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(cnf_matrix)
print(metrics.classification_report(y_test, y_pred))
```

Şekil 4.47. CityBlock-SFFS-kNN_vk2 mimarisinin simülasyon kodları.

¹⁷⁰ Bu algoritma kombinasyonuna CityBlock-SFFS-kNN_vk2 adı verilmiştir.

Şekil 4.47’de verilen analiz sonrasında elde edilen karışıklık matrisi Denklem 4.1’de yazılmış, bu matris bileşenleri kullanılarak hazırlanan performans raporu ise Tablo 4.12’de sunulmuştur.

$$C_{CityBlock} = \begin{pmatrix} DN & YN \\ DP & YP \end{pmatrix} = \begin{pmatrix} 26383 & 269 \\ 3249 & 23289 \end{pmatrix}, \quad (4.1)$$

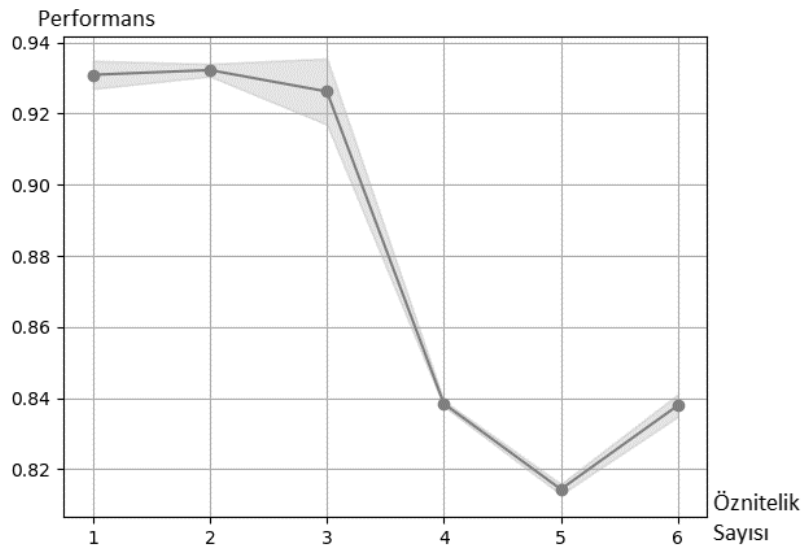
Tablo 4.12. CityBlock-SFFS-kNN_vk2 algoritmasının performans raporu.

Ağırlıklı Mod (0,1)	Kesinlik	Hatırlama	F1-puanı	Destek
0	0.89	0.99	0.94	26652
1	0.99	0.88	0.93	26538
Toplam				
Akürezi			0.93	53190
Makro Ortalama	0.94	0.93	0.93	53190
Ağırlıklı Ortalama	0.94	0.93	0.93	53190

Buradaki önemli ihtiyaç (yani asıl hedefimiz) algoritmanın sınıflandırma analizinin raporuna ulaşmaktır, zira ML modelinden veri tabanındaki “Sales (Satışlar)” özneteliğini etkileyen en önemli faktörleri belirlemesi beklenmiştir. Bu amaca yönelik algorithmadan ulaştığı sonuçları grafiksel olarak almak için Şekil 4.48’deki kodlama satırları kullanılmış Şekil 4.49’daki grafik elde edilmiştir.

```
In [15]: from mlxtend.plotting import plot_sequential_feature_selection as plot_sfs
import matplotlib.pyplot as plt
fig1 = plot_sfs(k_sfs.get_metric_dict(), kind='std_dev', confidence_interval=0.95)
plt.title('cityblock')
plt.grid()
plt.show()
```

Şekil 4.48. CityBlock-SFFS-kNN_vk2 mimarisinin performans grafiği kodları.



Şekil 4.49. CityBlock-SFFS-kNN_vk2 mimarisinin performans grafiği.

Şekil 4.49’dan görüldüğü üzere algoritma en tutarlı sonuçlarını “Sales (Satışlar)” özneteliğini etkileyen en önemli 2 faktör arandığında vermekte ve karar destek mekanizmasına etki eden özellik sayısı arttıkça

algoritmanın başarı oranının düştüğü görülmektedir. Bu durumu netleştirmek adına ayrıntılı rapor için Şekil 4.50'deki komut kullanıldığında Tablo 4.13'te verilen çıktılarına ulaşılmaktadır. Buradan "Sales (Satışlar)" öz niteliği için etken iki faktörün "Age (Yaş)" ve "Product ID (Ürün Kodu)" olduğu görülmektedir.

In [16]: k_sfs.subsets_

Şekil 4.50. CityBlock-SFFS-kNN_vk2 mimarisinin sınıflandırma raporu.

Tablo 4.13. CityBlock-SFFS-kNN_vk2 mimarisinin detaylı sınıflandırma raporu.

Öznitelik Sayısı	1	2	3	4	5	6
Akürezi	93.1	93.2	92.6	83.8	81.4	83.8
Agent ID				X	X	X
Age	X	X				X
Product ID		X	X	X	X	X
Area Code					X	X
Gender			X	X	X	X
Call Count			X	X	X	X

```
In [17]: #similasyon alanı

knn_clf = KNeighborsClassifier(n_neighbors=2,metric="euclidean")
# Build step feature selection
k_sfs = sfs(knn_clf,
            k_features=(1,6),
            forward=True,
            floating=True,
            verbose=2,
            scoring='accuracy',
            cv=5,
            n_jobs=-1)

# Perform SFFS
k_sfs = k_sfs.fit(X_train, y_train)
X_train_efs = k_sfs.transform(X_train)
X_test_efs = k_sfs.transform(X_test)

# Fit the estimator using the new feature subset
# and make a prediction on the test data
knn_clf.fit(X_train_efs, y_train)
y_pred = knn_clf.predict(X_test_efs)

# Compute the accuracy of the prediction
acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
# print('Test set accuracy: %.2f %%' % (acc*100))

feat_cols = list(k_sfs.k_feature_idx_)
print(feat_cols)

cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(cnf_matrix)
print(metrics.classification_report(y_test, y_pred))
```

Şekil 4.51. Euclidean-SFFS-kNN_vk2 mimarisinin simülasyon komutları.

Chebyshev ve Euclidean mesafe ölçüm modelleri üzerine inşa edilen SFFS-kNN algoritmaları için en uygun komşuluk sayısının Tablo 4.11’de $k=2$ olduğu belirtilmişti¹⁷¹. CityBlock-SFFS-kNN-vk2 mimarisinde olduğu gibi Chebyshev-SFFS-kNN_vk2 ve Euclidean-SFFS-kNN_vk2 yaklaşımları için de sınıflandırma simülasyonu yapılmış ve karışıklık matrisleri, performans raporları ve sınıflandırma çıkarımları için elde edilen sonuçlar derlenmiştir. Gerekli kodlama satırları Şekil 4.47’de sunulan komutlar dizini ile büyük oranda aynıdır, ancak dikkat edilmesi gereken kritik bir değişiklik bulunmaktadır: ilgili şekilde sunulan ve “knn_clf=...” ile başlayan satırın sonundaki “manhattan” ibaresi yerine sırasıyla “chebyshev” ve “euclidean” ibareleri kullanılmalıdır (örnek olması adına Şekil 4.51’de Euclidean-SFFS-kNN_vk2 yaklaşımının kaynak kodları sunulmuştur). Bu analizler CityBlock-SFFS-kNN-vk2 algoritmasında olduğu gibi yaklaşık 1’er dakika sürmüş ve bazı küçük farklılıklar dışında neredeyse aynı sonuçlar elde edilmiştir. Karışıklık matrisi Denklem 4.2’de, performans raporu ise Tablo 4.14’te paylaşılmıştır.

$$C_{Chebyshev} = C_{Euclidean} = C_{CityBlock} = \begin{pmatrix} DN & YN \\ DP & YP \end{pmatrix} = \begin{pmatrix} 26383 & 269 \\ 3249 & 23289 \end{pmatrix}. \quad (4.2)$$

Tablo 4.14. Chebyshev-SFFS-kNN_vk2 ve Euclidean-SFFS-kNN_vk2 raporları.

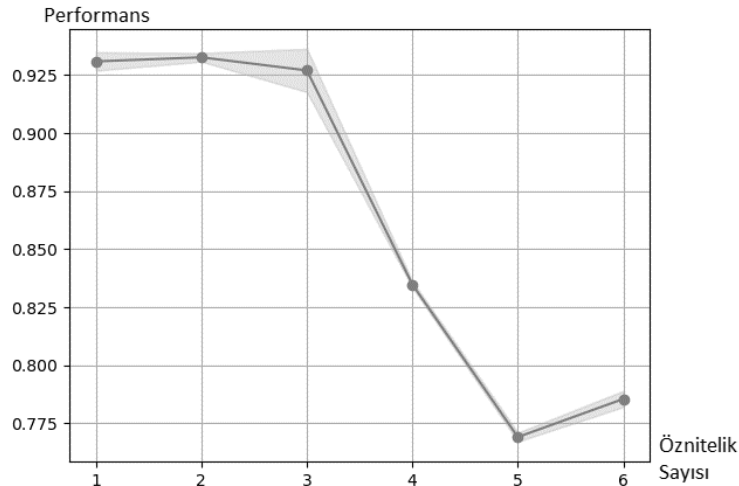
Ağırlıklı Mod (0,1)	Kesinlik	Hatırlama	F1-puanı	Destek
0	0.89	0.99	0.94	26652
1	0.99	0.88	0.93	26538
Toplam				
Akürezi			0.93	53190
Makro Ortalama	0.94	0.93	0.93	53190
Ağırlıklı Ortalama	0.94	0.93	0.93	53190

Sınıflandırma analizinin detaylı raporları için Şekil 4.48 ve Şekil 4.50’de yazılan kodlar kullanıldığında Tablo 4.15, Şekil 4.52, Tablo 4.16 ve Şekil 4.53’te paylaşılan sonuçlar elde edilmektedir.

Tablo 4.15: Chebyshev-SFFS-kNN vk2 mimarisinin detaylı sınıflandırma raporu.

Öznitelik Sayısı	1	2	3	4	5	6
Akürezi	93.1	93.3	92.7	83.5	79.9	78.5
Agent ID				X	X	X
Age	X	X				X
Product ID		X	X	X	X	X
Area Code					X	X
Gender			X	X	X	X
Call Count			X	X	X	X

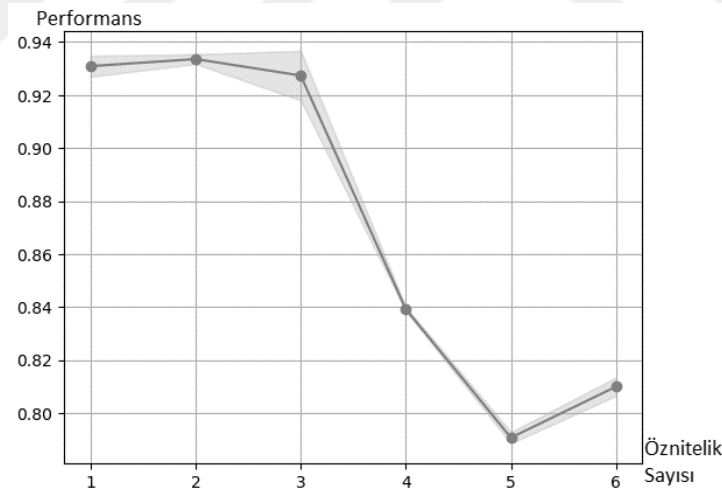
¹⁷¹ Bu mimari kombinasyonları “Chebyshev-SFFS-kNN_vk2” ve “Euclidean-SFFS-kNN_vk2” şeklinde kodlanmıştır.



Şekil 4.52. Chebyshev-SFFS-kNN_vk2 mimarisinin performans grafiği.

Tablo 4.16. Euclidean-SFFS-kNN_vk2 mimarisinin detaylı sınıflandırma raporu.

Öznitelik Sayısı	1	2	3	4	5	6
Akürezi	93.1	93.4	92.7	83.9	76.1	80.9
Agent ID				X	X	X
Age	X	X				X
Product ID		X	X	X	X	X
Area Code					X	X
Gender			X	X	X	X
Call Count			X	X	X	X



Şekil 4.53. Euclidean-SFFS-kNN_vk2 mimarisinin performans grafiği.

Tablo 4.15, Şekil 4.52, Tablo 4.16 ve Şekil 4.53'te sunulan verilerden Chebyshev-SFFS-kNN_vk2 ve Euclidean-SFFS-kNN_vk2 yaklaşımlarının CityBlock-SFFS-kNN-vk2 algoritmasıyla aynı nihai sonucu ürettiği görülmektedir, diğer bir deyişle veri setini incelediğimiz firmanın satışlarını etkileyen en önemli iki faktör müşterilerinin yaşları ve satılan ürün tipleridir.

Şimdi, dikkatimizi Correlation mesafe ölçüm modeli üzerine inşa edilen SFFS-kNN yaklaşımına odaklanıyoruz. Bu yaklaşımda en uygun hiper-parametre değeri Tablo 4.11'de vurgulandığı

üzere $k=1$ 'dir¹⁷². Bu yöntem bu bölümde araştırılan ilk üç algoritmaya göre daha uzun zaman almış ve yaklaşık 47 dakika sürmüştür. Sonuç olarak elde edilen karışıklık matrisine Denklem 4.3'ten ve bu matris yardımıyla elde edilen performans raporuna ise Tablo 4.17'den ulaşmak mümkündür.

$$C_{Correlation} = \begin{pmatrix} DN & YN \\ DP & YP \end{pmatrix} = \begin{pmatrix} 24212 & 2440 \\ 2639 & 23899 \end{pmatrix}. \quad (4.3)$$

Tablo 4.17. Correlation-SFFS-kNN_vk1 yaklaşımı performans raporu.

Ağırlıklı Mod (0,1)	Kesinlik	Hatırlama	F1-puanı	Destek
0	0.90	0.91	0.91	26652
1	0.91	0.90	0.90	26538
Toplam				
Akürezi			0.90	53190
Makro Ortalama	0.90	0.90	0.90	53190
Ağırlıklı Ortalama	0.90	0.90	0.90	53190

```
In [17]: #similasyon alanı

knn_clf = KNeighborsClassifier(n_neighbors=1,metric="correlation")
# Build step feature selection
k_sfs = sfs(knn_clf,
            k_features=(1,6),
            forward= True,
            floating=True,
            verbose=2,
            scoring='accuracy',
            cv=5,
            n_jobs=-1)

# Perform SFFS
k_sfs = k_sfs.fit(X_train, y_train)
X_train_efs = k_sfs.transform(X_train)
X_test_efs = k_sfs.transform(X_test)

# Fit the estimator using the new feature subset
# and make a prediction on the test data
knn_clf.fit(X_train_efs, y_train)
y_pred = knn_clf.predict(X_test_efs)

# Compute the accuracy of the prediction
acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
# print('Test set accuracy: %.2f %%' % (acc*100))

feat_cols = list(k_sfs.k_feature_idx_)
print(feat_cols)

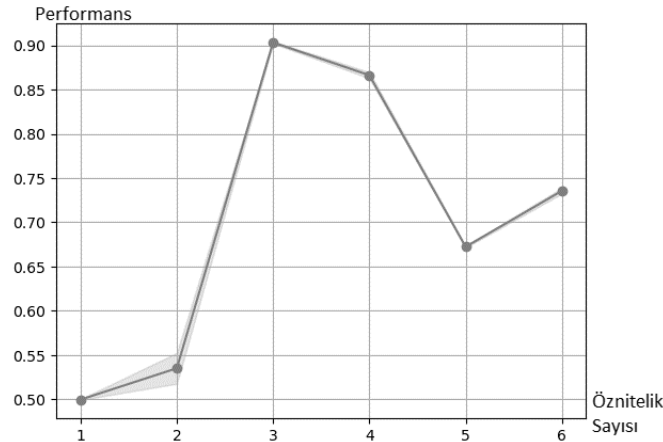
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(cnf_matrix)
print(metrics.classification_report(y_test, y_pred))
```

Şekil 4.54. Correlation-SFFS-kNN_vk1 mimarisinin simülasyon komutları.

Correlation modelinin sınıflandırma çıktıları için Şekil 4.48 ve 4.50'de paylaşılan kodlama satırlarının uygun biçimde düzenlenerek (Şekil 4.54'te paylaşılan kodlama satırları) kullanılmasıyla Şekil 4.55'te sunulan grafik raporu ve Tablo 4.18'de vurgulanan öznetelik durumları elde edilmektedir. Görüldüğü

¹⁷² Bu algoritma kombinasyonuna Correlation-SFFS-kNN_vk1 adı verilmiştir.

üzere, Correlation-SFFS-kNN_vk1 modeli en başarılı tahmin tutarlılığını ilgili firmanın satışlarını etkileyen üç öz nitelik arandığında sergilemektedir.



Şekil 4.55. Correlation-SFFS-kNN_vk1 mimarisinin performans grafiği.

Tablo 4.18. Correlation-SFFS-kNN_vk1 mimarisinin detaylı sınıflandırma raporu.

Öz nitelik Sayısı	1	2	3	4	5	6
Akürezi	49.9	53.5	90.3	86.6	67.2	73.5
Agent ID	X	X			X	X
Age		X	X	X	X	X
Product ID			X	X	X	X
Area Code						X
Gender			X	X	X	X
Call Count				X	X	X

```
In [17]: #similasyon alanı

knn_clf = KNeighborsClassifier(n_neighbors=2,metric="canberra")
# Build step feature selection
k_sfs = sfs(knn_clf,
            k_features=(1,6),
            forward= True,
            floating=True,
            verbose=2,
            scoring='accuracy',
            cv=5,
            n_jobs=-1)

# Perform SFFS
k_sfs = k_sfs.fit(X_train, y_train)
X_train_efs = k_sfs.transform(X_train)
X_test_efs = k_sfs.transform(X_test)

# Fit the estimator using the new feature subset
# and make a prediction on the test data
knn_clf.fit(X_train_efs, y_train)
y_pred = knn_clf.predict(X_test_efs)

# Compute the accuracy of the prediction
acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
# print('Test set accuracy: %.2f %%' % (acc*100))

feat_cols = list(k_sfs.k_feature_idx_)
print(feat_cols)

cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(cnf_matrix)
print(metrics.classification_report(y_test, y_pred))
```

Şekil 4.56. Canberra-SFFS-kNN_vk2 algoritmasının simülasyon komutları.

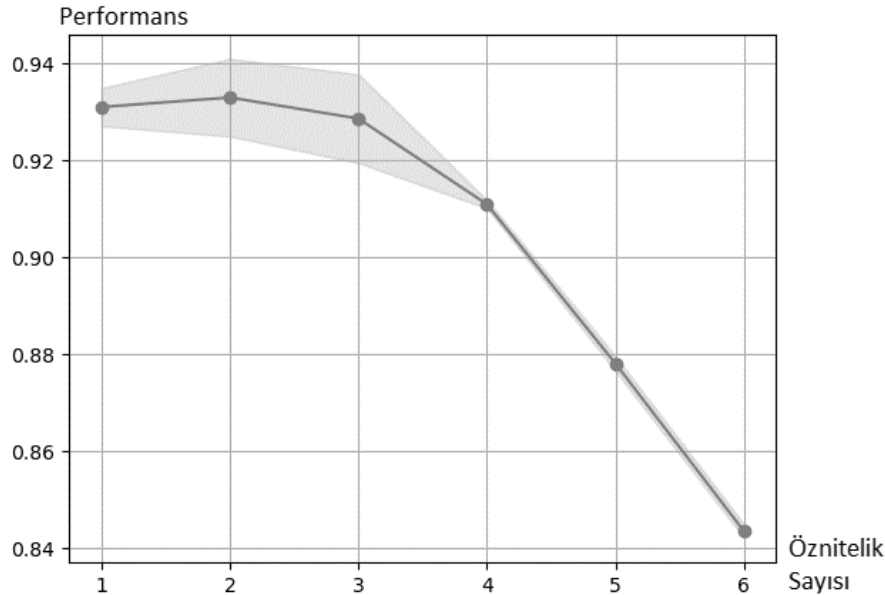
Tablo 4.11'e göre Canberra mesafe ölçümünü baz alan SFFS-kNN algoritmasında en tutarlı tahminler $k=2$ hiper-parametre değerinde elde edilmektedir¹⁷³. Canberra-SFFS-kNN_vk2 yaklaşımıyla yapılan analiz yaklaşık olarak 40 dakika sürmüştür (algoritma kodlamaları için bakınız Şekil 4.56) ve şimdiye kadar tartışılan CityBlock-SFFS-kNN_vk2, Chebyshev-SFFS-kNN_vk2, Euclidean-SFFS-kNN_vk2 ve Correlation-SFFS-kNN_vk1 tipi sınıflandırma algoritmalarına kıyasla nispeten daha başarılı performans sergilemiştir. Bu modelle yapılan araştırmadan elde edilen karışıklık matrisi Denklem 4.4'te ve ilgili öğrenme algoritmasının performans raporu ise Tablo 4.19'da verilmiştir.

$$C_{Canberra} = \begin{pmatrix} DN & YN \\ DP & YP \end{pmatrix} = \begin{pmatrix} 26412 & 240 \\ 3028 & 23510 \end{pmatrix}. \quad (4.4.)$$

Tablo 4.19. Canberra-SFFS-kNN vk2 yaklaşımı performans raporu.

Ağırlıklı Mod (0,1)	Kesinlik	Hatırlama	F1-puanı	Destek
0	0.90	0.99	0.94	26652
1	0.99	0.89	0.94	26538
Toplam				
Akürezi			0.94	53190
Makro Ortalama	0.94	0.94	0.94	53190
Ağırlıklı Ortalama	0.94	0.94	0.94	53190

Şekil 4.48 ve Şekil 4.50'de paylaşılan komut satırlarının gereken şekilde değiştirilerek kullanılmasıyla Şekil 4.57'te paylaşılan grafik raporu ve Tablo 4.20'de vurgulanan öznetelik durumları elde edilmektedir.



Şekil 4.57. Canberra-SFFS-kNN_vk2 mimarisinin performans grafiği.

¹⁷³ Bu algoritma kombinasyonuna Canberra-SFFS-kNN_vk2 adı verilmiştir.

Tablo 4.20. Canberra-SFFS-kNN_vk2 mimarisinin detaylı sınıflandırma raporu.

Öznitelik Sayısı	1	2	3	4	5	6
Akürezi	93.1	93.3	92.7	91.1	87.8	84.3
Agent ID				X	X	X
Age	X	X				X
Product ID			X	X	X	X
Area Code					X	X
Gender		X	X	X	X	X
Call Count			X	X	X	X

Seçilen veri havuzundaki satışları betimleyen "Sales (Satışlar)" öz niteliğini etkileyen ana faktörlerin Canberra-SFFS-kNN_vk2 algoritmasına göre "Age (Yaş)" ve "Gender (Cinsiyet)" olduğu görülmektedir. Canberra-SFFS-kNN_vk2 yaklaşımına göre bu iki öz niteliğe odaklanıldığında "Olumsuz Pazarlama" ve "Olumlu Pazarlama" durumları için faydalı raporlar hazırlamak mümkündür.

Sıradaki adımda yeni bir SFFS-kNN mimarisi oluşturmak için Tablo 4.11'den faydalanılarak Dice_vk1 ve Dice_vk5 tipi ölçümlere odaklanılmıştır. Bu algoritmaların kullanıldığı Python kodlamaları Şekil 4.58 ve Şekil 4.59'da verilmiştir¹⁷⁴.

```
In [17]: #similasyon alanı

knn_clf = KNeighborsClassifier(n_neighbors=1,metric="dice")
# Build step feature selection
k_sfs = sfs(knn_clf,
            k_features=(1,6),
            forward= True,
            floating=True,
            verbose=2,
            scoring='accuracy',
            cv=5,
            n_jobs=-1)

# Perform SFFS
k_sfs = k_sfs.fit(X_train, y_train)
X_train_efs = k_sfs.transform(X_train)
X_test_efs = k_sfs.transform(X_test)

# Fit the estimator using the new feature subset
# and make a prediction on the test data
knn_clf.fit(X_train_efs, y_train)
y_pred = knn_clf.predict(X_test_efs)

# Compute the accuracy of the prediction
acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
# print('Test set accuracy: %.2f %%' % (acc*100))

feat_cols = list(k_sfs.k_feature_idx_)
print(feat_cols)

cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(cnf_matrix)
print(metrics.classification_report(y_test, y_pred))
```

Şekil 4.58. Dice-SFFS-kNN_vk1 algoritmasının simülasyon komutları.

¹⁷⁴ Bu mimari kombinasyonları kısaca "Dice-SFFS-kNN_vk1" ve "Dice-SFFS-kNN_vk5" şeklinde kodlanmıştır.

```

In [17]: #similasyon alanı

knn_clf = KNeighborsClassifier(n_neighbors=5,metric="dice")
# Build step feature selection
k_sfs = sfs(knn_clf,
            k_features=(1,6),
            forward= True,
            floating=True,
            verbose=2,
            scoring='accuracy',
            cv=5,
            n_jobs=-1)

# Perform SFFS
k_sfs = k_sfs.fit(X_train, y_train)
X_train_efs = k_sfs.transform(X_train)
X_test_efs = k_sfs.transform(X_test)

# Fit the estimator using the new feature subset
# and make a prediction on the test data
knn_clf.fit(X_train_efs, y_train)
y_pred = knn_clf.predict(X_test_efs)

# Compute the accuracy of the prediction
acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
# print('Test set accuracy: %.2f %%' % (acc*100))

feat_cols = list(k_sfs.k_feature_idx_)
print(feat_cols)

cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(cnf_matrix)
print(metrics.classification_report(y_test, y_pred))

```

Şekil 4.59. Dice-SFFS-kNN_vk5 algoritmasının simülasyon komutları.

Dice-SFFS-kNN_vk1 (analiz 65 dakika sürmüştür) ve Dice-SFFS-kNN_vk5 (analiz 75 dakika sürmüştür) mimarileri için karışıklık matrisleri aynı forma sahiptir ve şu şekilde yazılmaktadır:

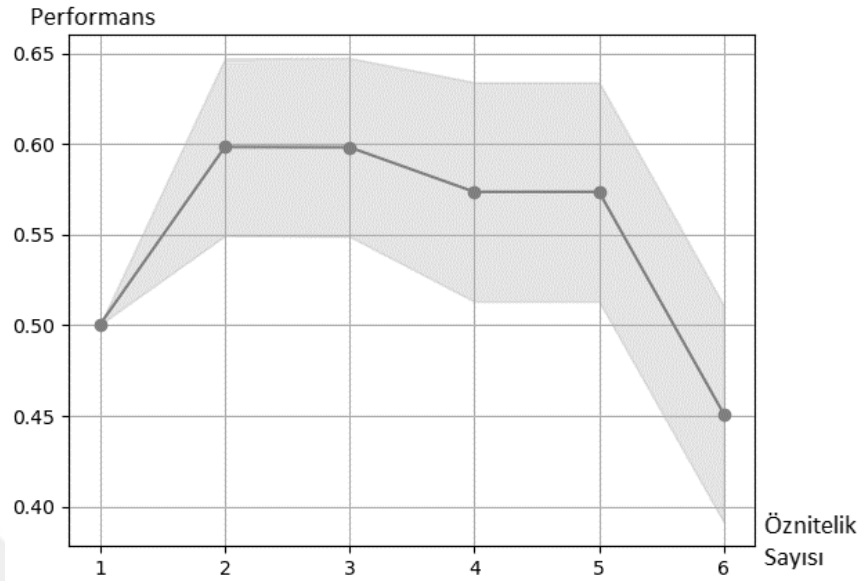
$$C_{Dice_vk1} = C_{Dice_vk5} = \begin{pmatrix} DN & YN \\ DP & YP \end{pmatrix} = \begin{pmatrix} 26412 & 240 \\ 3028 & 23510 \end{pmatrix}. \quad (4.5)$$

Böylece, yukarıdaki matris bileşenlerine göre hesaplanan performans değerlerinin Dice-SFFS-kNN_vk1 ve Dice-SFFS-kNN_vk5 algoritmalarında aynı olacağı aşikardır. İlgili performans verileri Tablo 4.21’de sunulmuştur.

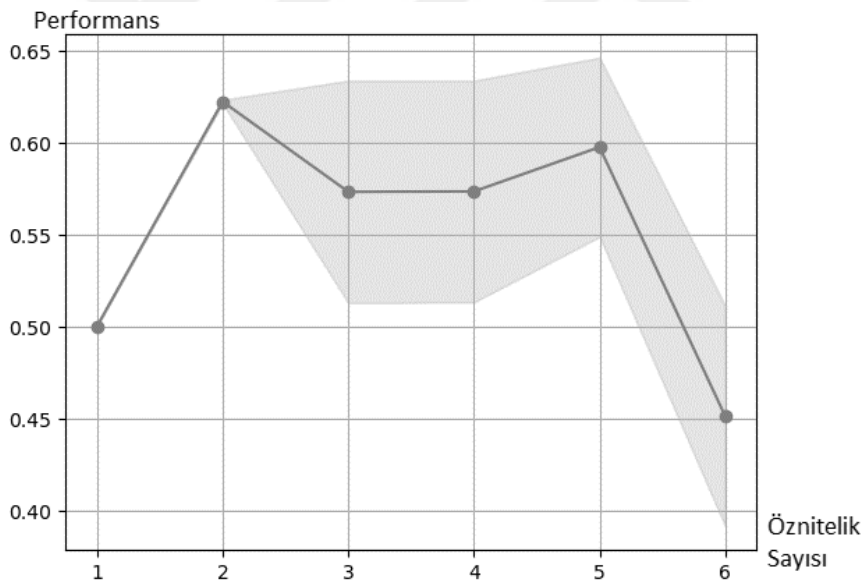
Tablo 4.21. Dice-SFFS-kNN_vk1 ve Dice-SFFS-kNN_vk5 için performans raporu.

Ağırlıklı Mod (0,1)	Kesinlik	Hatırlama	F1-puanı	Destek
0	0.68	0.48	0.56	26652
1	0.60	0.77	0.67	26538
Toplam				
Akürezi			0.63	53190
Makro Ortalama	0.64	0.63	0.62	53190
Ağırlıklı Ortalama	0.64	0.63	0.62	53190

Ek olarak Dice-SFFS-kNN_vk1 ve Dice-SFFS-kNN_vk5 öğrenme algoritmalarının sınıflandırma sonuçlarına Şekil 4.60, Şekil 4.61, Tablo 4.22 ve Tablo 4.23'ten ulaşmak mümkündür.



Şekil 4.60. Dice-SFFS-kNN_vk1 mimarisinin performans grafiği.



Şekil 4.61. Dice-SFFS-kNN_vk5 mimarisinin performans grafiği.

Tablo 4.22. Dice-SFFS-kNN vk1 mimarisinin detaylı sınıflandırma raporu.

Öz nitelik Sayısı	1	2	3	4	5	6
Akürezi	50.1	59.83	59.81	57.3	57.3	45.1
Agent ID				X	X	X
Age	X	X				X
Product ID			X	X	X	X
Area Code					X	X
Gender		X	X	X	X	X
Call Count			X	X	X	X

Tablo 4.23. Dice-SFFS-kNN_vk5 mimarisinin detaylı sınıflandırma raporu.

Öznitelik Sayısı	1	2	3	4	5	6
Akürezi	50.1	62.25	57.35	57.37	59.8	45.2
Agent ID				X	X	X
Age	X	X				X
Product ID			X	X	X	X
Area Code					X	X
Gender		X	X	X	X	X
Call Count			X	X	X	X

İki kritik özniteliği aradığımız zaman her iki formülasyonun da en iyi performansını sergilediği görülmektedir, ancak buraya kadar inşa ettiğimiz diğer 6 farklı kNN mimarisi ile karşılaştırıldığında en başarısız doğruluk oranlarının bu modellerde elde edildiği sonucuna ulaşılmıştır.

```
In [17]: #similasyon alanı

knn_clf = KNeighborsClassifier(n_neighbors=2,metric="cosine")
# Build step feature selection
k_sfs = sfs(knn_clf,
            k_features=(1,6),
            forward= True,
            floating=True,
            verbose=2,
            scoring='accuracy',
            cv=5,
            n_jobs=-1)

# Perform SFFS
k_sfs = k_sfs.fit(X_train, y_train)
X_train_efs = k_sfs.transform(X_train)
X_test_efs = k_sfs.transform(X_test)

# Fit the estimator using the new feature subset
# and make a prediction on the test data
knn_clf.fit(X_train_efs, y_train)
y_pred = knn_clf.predict(X_test_efs)

# Compute the accuracy of the prediction
acc = float((y_test == y_pred).sum()) / y_pred.shape[0]
# print('Test set accuracy: %.2f %%' % (acc*100))

feat_cols = list(k_sfs.k_feature_idx_)
print(feat_cols)

cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
print(cnf_matrix)
print(metrics.classification_report(y_test, y_pred))
```

Şekil 4.62.Cosine-SFFS-kNN_vk2 mimarisinin simülasyon komutları.

Son olarak, Cosine mesafe ölçümü baz alınarak inşa edilebilecek SFFS-kNN algoritması ele alınmıştır. Tablo 4.11’de paylaştığımız algoritma ayarlama evresi sonuçlarına baktığımızda burada $k=2$ almamız gerektiği anlaşılmaktadır¹⁷⁵. Bu analizin Python kodları Şekil 4.62’de sunulmuştur.

¹⁷⁵ Bu mimari kombinasyonu kısaca “Cosine-SFFS-kNN_vk2” şeklinde kodlanmıştır.

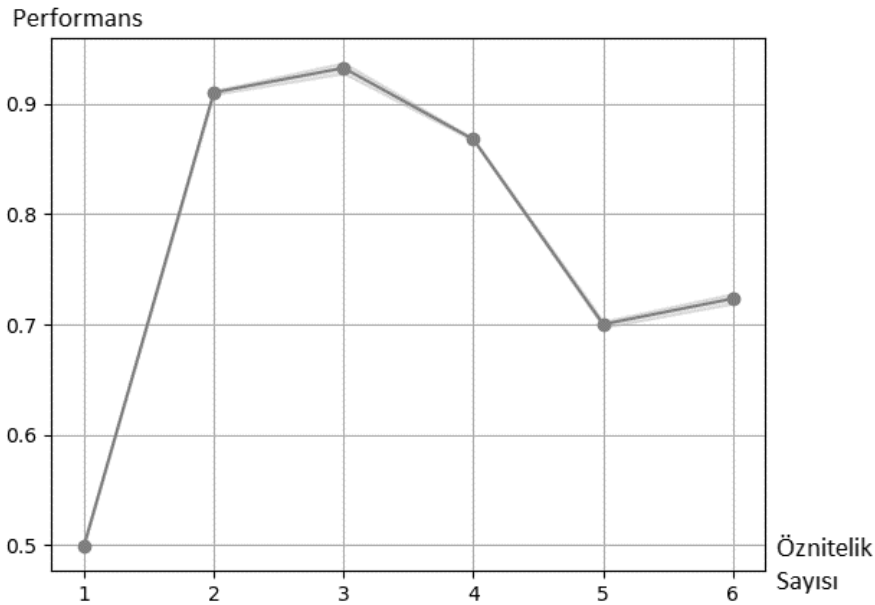
$$C_{Cosine} = \begin{pmatrix} DN & YN \\ DP & YP \end{pmatrix} = \begin{pmatrix} 26626 & 26 \\ 3443 & 23195 \end{pmatrix}. \quad (4.6)$$

Yukarıdaki denklemde Cosine-SFFS-kNN_vk2 algoritmasından elde edilen karışıklık matrisi yazılmıştır. Bu matris bileşenleri bizi Tablo 4.24'te paylaşılan performans değerlerine götürmektedir.

Tablo 4.24. Cosine-SFFS-kNN_vk2 için performans raporu.

Ağırlıklı Mod (0,1)	Kesinlik	Hatırlama	F1-puanı	Destek
0	0.89	1.00	0.94	26652
1	1.00	0.87	0.93	26538
Toplam				
Akürezi			0.94	53190
Makro Ortalama	0.94	0.94	0.94	53190
Ağırlıklı Ortalama	0.94	0.94	0.94	53190

Okuyucu yaklaşık 70 dakika süren Cosine mesafe fonksiyonu tabanlı SFFS-kNN yaklaşımının sınıflandırma analizini genel hatlarıyla Şekil 4.63'ten kontrol edebilirken Tablo 4.25'ten ise ayrıntılı nümerik çıktılara ulaşabilecektir. Bu yöntem, "Sales (Satışlar)" özelliğini doğrudan etkileyen 3 özelliği araştırdığımızda en iyi tahminlerini üretirken satışlarla ilintili olan kritik faktörlerin "Product ID (Ürün Kimliği)", "Gender (Cinsiyet)" ve "Call Count (Arama Sayısı)" olduğu sonucunu sunmaktadır. Bilindiği üzere, bu özelliklerin tahmin analizi sonucunda ortaya çıkması herhangi bir şirket için oldukça faydalı bilgilerdir. Bu bağlamda, bilgisayarın Coside tabanlı SFFS-kNN mimarisi üzerinden tutarlı tahminler ürettiğini burada vurgulamak yerinde olacaktır.



Şekil 4.63. Cosine-SFFS-kNN_vk2 mimarisinin performans grafiği.

Tablo 4.25. Cosine-SFFS-kNN_vk2 mimarisinin detaylı sınıflandırma çıktıları.

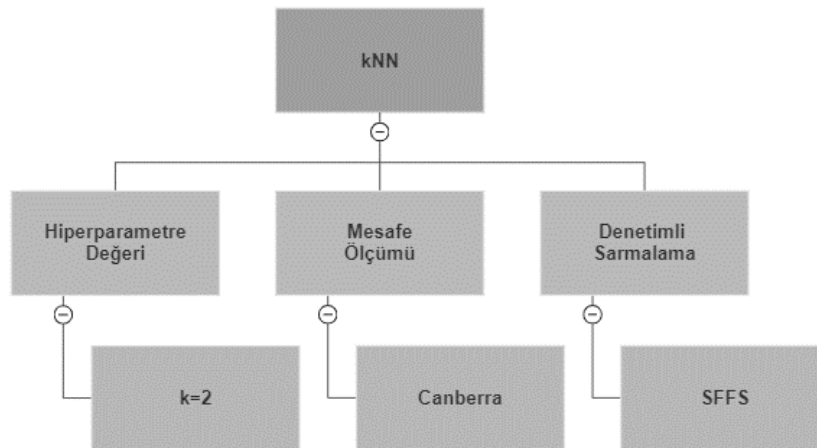
Öznitelik Sayısı	1	2	3	4	5	6
Akürezi	49.97	91.01	93.22	86.75	70.01	72.34
Agent ID	X	X		X	X	X
Age						X
Product ID		X	X	X	X	X
Area Code					X	X
Gender			X	X	X	X
Call Count			X	X	X	X

4.4. Ne Öğrendik?

Araştırmamızda kapsamında kNN öğrenme yaklaşımı için çeşitli algoritma alt birimleri ve farklı hiper-parametre değerleri kullanılmış ve böylece karşılaştırmalı istatistiksel analizlerle incelenen tele-pazarlama verilerine en uygun kNN mimarisi oluşturulmuştur. Seçilen tele-pazarlama veri setinin ML analizi kNN mimarisinin çeşitli biçimlerinden yararlanılarak gerçekleştirilmiş elde edilen en çarpıcı sonuçlar Tablo 4.26'da özetlenmiştir. Burada sunulan yüksek başarı oranlı durumlarda dikkat çekici bir sonuç çıkmaktadır: müşterinin yaşı (Age), müşterinin cinsiyeti (Gender) ve pazarlanan ürünün çeşidi (Product ID) satışları (Sales) etkileyen en önemli faktörlerdir. Hızla tamamlanacak bir analiz süreci tercih edildiğinde elde edilecek potansiyel başarı oranı görece düşerken, maksimum faydalı tahmin sonuçlarının elde edilmesi hedeflendiğinde ise zamandan fedakârlık etmek gerekmektedir. Burada kullanışlı bir mimari tasarımı da açıkça ön plana çıkmaktadır.

Tablo 4.26. Tasarlanan kNN algoritmalarının karşılaştırılması.

kNN Mimarisi	Tutarlılık	Performans	Etkin Faktörler	Süre
CityBlock-SFFS_vk2	0.93	0.9322	Age, Product ID	1 dk
ChebyShev-SFFS_vk2	0.93	0.9327	Age, Product ID	1 dk
Euclidean-SFFS_vk2	0.93	0.9336	Age, Product ID	1 dk
Correlation-SFFS_vk1	0.90	0.9032	Age, Product ID, Gender	47 dk
Canberra-SFFS_vk2	0.94	0.9329	Age, Gender	40 dk
Dice-SFFS_vk1	0.63	0.5983	Gender, Call Count	75 dk
Dice-SFFS_vk5	0.63	0.6225	Product ID, Gender	65 dk
Cosine-SFFS_vk2	0.94	0.9322	Product ID, Gender, Call Count	70 dk

**Şekil 4.64.**İncelenen veri setinin analizi için en uygun/faydalı kNN mimarisi.

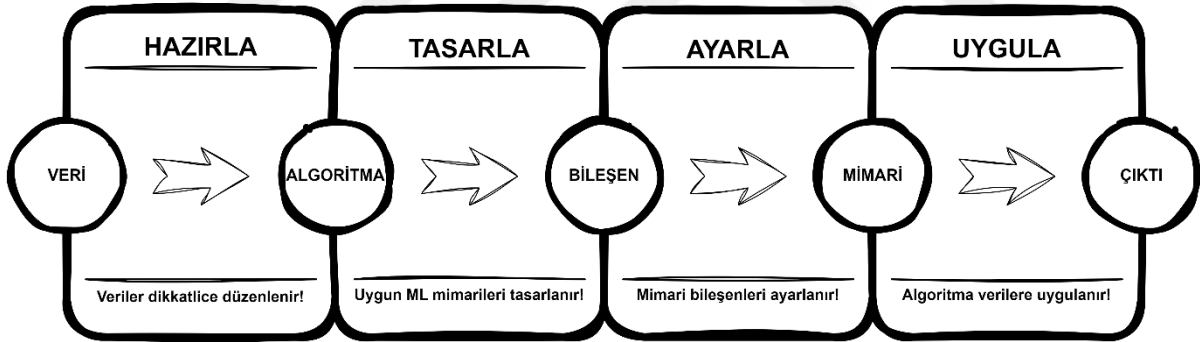
Yüksek tutarlılık oranı ve başarı performansı temel kriterler olarak ele alındığında Şekil 4.64'te paylaşılan algoritmanın incelenen tele-pazarlama veri havuzunun analizi için en uygun seçenek olduğunu söylemek mümkündür.



SONUÇ ve ÖNERİLER

Günümüzde veri ve bilginin öneminin artmasında insan beklentileri ve teknolojik gelişmeler hayati rol oynamaktadır. Bu doğrultuda çeşitli kaynaklardan toplanan veriler farklı veri depolama ortamlarında muhafaza edilmektedir. Sonuç olarak, boyutları artan ve yapısı değişen veri yığınlarını işlemek için literatürde çeşitli yaklaşımlar sunulmuş ve yeni yaklaşımlar için de araştırmalar devam etmektedir. “Gerçek dünya” senaryolarını temsil eden matematiksel denklemlere (ya da fonksiyonlara) odaklanma ile ilgili olan yapay öğrenme yaklaşımları bu yöntemler arasındadır. Yapay öğrenme yöntemlerinde ele alınan temel konular arasında regresyon, sınıflandırma, kümeleme, transkripsiyon, makine çevirisi, anormallik tespiti, sentez ve örnekleme, olasılık yoğunluğunun tahmini, olasılık kütle fonksiyonunun tahmini, benzerlik eşleştirme, benzerlik gruplama, nedensel modelleme ve bağlantı profili oluşturma gelmektedir. Öte yandan, yukarıda listelenen farklı görevlerin üstesinden gelmek için kullanılan temel geliştirme aşamaları ise keşfedici veri analizi, veri ön işleme, özellik mühendisliği (özellik oluşturma/çıkarma, özellik seçimi, özellik boyutluluk azaltma), eğitim algoritması, test-eşleştirme, model (mimari) seçimi, yaklaşım izleme ve yeniden eğitimidir.

Bu araştırma kapsamında temel olarak bir X firmasının telefonla pazarlama verilerini sınıflandırma yapabilen bir ML mimarisi ile analiz etmeyi amaçladık. Araştırmamızın iskeletini özetlediğimiz işlem akış şeması Şekil 4.65’te sunulmuştur.



Şekil 4.65. Tez analizlerinin genel akış şeması.

Söz konusu veri setini açıklayan tablo 100.000 satır ve 12 sütundan oluşmaktadır. Bu veri setindeki bazı bilgiler sayısal formatta iken bazıları da kategoriktir, ayrıca sınıflandırma analizi yaparken odaklandığımız ana özellik ise tablodaki "Sales (Satışlar)" sütunudur. Bir veri setini başarılı bir şekilde sınıflandırabilen ML algoritmaları arandığında literatürde en çok kullanılanların ön plana çıktığı kolaylıkla anlaşılmaktadır. Örneğin, kNN, SVMs, DL, DTs, NBC, ELM ve C5.0 kod adıyla bilinen algoritmalar iyi bilinen ve sıklıkla kullanılan yaklaşımlar arasındadır. Bu çalışmada kapsamında literatürde en başarılı sınıflandırma algoritmaları arasında olduğu düşünülen kNN mimarisi kullanılmıştır.

Yapay öğrenme mimarileri çeşitli alt birimler ile kritik hiper-parametrelerden oluşur ve en uygun bileşen kombinasyonunun seçilmesi bir öğrenme algoritmasının başarısı için kritik öneme

sahiptir. Ne yazık ki algoritma en uygun alt birimlerinin seçimini kendisi yapamaz ve en iyi hiper-parametre değerini öğrenemez, bu nedenle algoritmaya dışarıdan yapılacak tasarım müdahaleleri çok önemlidir. Araştırmamızın ilk aşamasında, incelenen tele-pazarlama veri setindeki anlamsız (ya da işe yaramaz) bilgiler temizlenmiş, havuzdaki veri tipleri aynı olacak şekilde düzenlemeler yapılmış ve veri seti dengelenmiştir. Bu görevler, Python'un çeşitli araçları sayesinde bir dizi komutla yapılabilmektedir. Araştırmamızdaki ikinci ve üçüncü önemli aşamalar ise kullanılacak en uygun ML mimarisinin tasarlanması ve ayarlanmasıdır. Bu görevler genellikle harcanan araştırma süresi açısından bir yapay öğrenim araştırmasının önemli kısmını kapsamaktadır. Araştırmamızda çeşitli alt birimler ve farklı hiper-parametre değerleri kullanılmış ve ardından karşılaştırmalı istatistiksel analizlerle en uygun kNN mimarisi inşa edilmiştir. Oluşturulan kNN mimarileri arasında en faydalı olanın Canberra-SFFS-kNN_vk2 kod adını verdiğimiz algoritmanın olduğu görülmüştür. Bu yapay sınıflandırma algoritmasına göre ele alınan tele-pazarlama veri havuzunda "Sales (Satış)" özelliğini etkileyen en önemli faktörlerin "Age (Yaş)" ve "Gender (Cinsiyet)" bilgileri olduğu anlaşılmıştır. Diğer yandan, araştırmamızın tamamı göz önüne alındığında bu iki etkili özelliğin yanına "Product ID (Ürün Kimliği)" bilgisinin eklenmesi uygun olacaktır.

Özetle bu tez çalışması yapay öğrenme işlevine sahip bilgisayar temelli algoritmaların tele-pazarlama verilerinin işlenmesi görevini de başarıyla yapabileceğini göstermektedir. Diğer yandan, araştırılan veri havuzunun SVMs ve DTs gibi farklı sınıflandırma temelli öğrenme algoritmaları perspektifinden ele alınması farklı öğrenme mimarilerini karşılaştırma imkânı sunacaktır. Bu bağlamda, tez kapsamında incelenen araştırma konusunun “ucu acık bir problem” olarak nitelendirilebileceği aşikardır. Araştırmamız yeni tez, makale ve bildiri çalışmalarına özgün konular sunabilme potansiyeli taşımaktadır. Burada son olarak tez kapsamında sunulan sonuçların bilim insanlarını ve tele-pazarlama yöntemini kullanan firma yetkililerini bu alana yönelme bağlamında motive edeceğine inandığımızı vurgulamak yerinde olacaktır.

KAYNAKLAR

- Abbad, A. ve Tairi, H. (2016). Combining Jaccard and Mahalanobis Cosine distance to enhance the face recognition rate. *WSEAS Transactions on Signal Processing*, 16, 171-178.
- Ade, P. A. R., et al. (2016). Planck 2015 results - XIII. Cosmological parameters. *A&A*, 594, A13.
- Adek, R. T. ve Ula, M. (2020, Temmuz, 19-27). *A Survey on The Accuracy of Machine Learning Techniques for Intrusion and Anomaly Detection on Public Data Sets*. International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA), Medan, Indonesia.
- Aghanim, N. et al. (2020). Planck 2018 results - VI. Cosmological parameters. *A&A*, 641, A6.
- Agrawal, S. What is Instance-Based Learning? <https://medium.com> adresinden 28 Ocak 2023 tarihinde alınmıştır.
- Aha, D. W., Kibler, D. ve Albert, M. K. (1991). Instance based learning algorithms. *Machine Learning*, 6, 37-66.
- Ahn, J.-H. ve Ezawa, K. J. (1997). Decision support for real-time telemarketing operations through Bayesian network learning. *Decision Support Systems*. 21, 17-27.
- Akila, A. ve Chandra, E. (2013). Slope finder - A distance measure for DTW based isolated word speech recognition. *International Journal of Engineering And Computer Science*, 2, 3411-3417.
- Alam, S., et al. (2017). The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: cosmological analysis of the DR12 galaxy sample. *MNRAS*, 470, 2617-2652.
- Alfeilat, H. A. A., et al. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data*, 7, 221-248.
- Alpar, R. (2013). *Çok Değişkenli İstatistiksel Yöntemler*. Detay Yayıncılık.
- Alpaydın (2011). E. *Yapay Öğrenme*. Boğaziçi Üniversitesi Yayınevi.
- Arnott, D. ve Pervan, G. (2008). Eight key issues for the decision support systems discipline. *Decision Support Systems*. 44, 657-672.
- Arslan, E. Makine Öğrenmesi-KNN (K-Nearest Neighbors) Algoritması Nedir? <https://www.medium.com> adresinden 04 Şubat 2023 tarihinde alınmıştır.
- Arjona, R. (2020). Machine learning meets the redshift evolution of the CMB temperature. *J. Cosmol. Astropart. Phys*, 08, 009.
- Baccouche, M., Mamalet, F., Wolf, C., Garcia, C. ve Baskurt, A. (2011). Sequential Deep Learning for Human Action Recognition. *Lecture Notes in Computer Science*, 7065, 29-39.
- Barghout, L. (2015). Spatial-Taxon Information Granules as Used in Iterative Fuzzy-Decision-Making for Image Segmentation. *Granular Computing and Decision-Making. Studies in Big Data*, 10, 285-318.
- Battistelli, E. S., et al. (2002). Cosmic Microwave Background Temperature at Galaxy Clusters. *Astrophys. J.*, 580, L101.

- Bajaj, A. Performance Metrics in Machine Learning [Complete Guide]. <https://neptune.ai> adresinden 03 Şubat 2023 tarihinde alınmıştır.
- Bennett, C. L., et al. (2013). Nine-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Final Maps and Results. *ApJS*, 208, 20.
- Benvenuto, F., Piana, M., Campi, C. ve Massone, A. M. (2018). A hybrid supervised/unsupervised machine learning approach to solar flare prediction. *The Astrophysical Journal*, 853, 90.
- Berry, M. J. ve Linoff, G. S. (2004). *Data mining techniques: for marketing, sales and customer relationship management*. John Wiley & Sons Press.
- Bharkad, S. D. ve Kokare, M. (2011). Performance evaluation of distance metrics: application to fingerprint recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 25, 777-806.
- Bhattachayya, A. (1943). On a measure of divergence between two statistical population defined by their population distributions. *Bulletin Calcutta Mathematical Society*, 35, 99-109.
- Blake, C., et al. (2012). The WiggleZ Dark Energy Survey: joint measurements of the expansion and growth history at $z < 1$. *Monthly Notices of the Royal Astronomical Society*, 425, 405-414.
- Bonamente, M., et al. (2006). Determination of the Cosmic Distance Scale from Sunyaev-Zel'dovich Effect and Chandra X-Ray Measurements of High-Redshift Galaxy Clusters. *ApJ*. 647, 25-54.
- Borugadda, P., Nandru, P. ve Madhavaiah, C. (2021). Predicting the Success of Bank Telemarketing for Selling Long-term Deposits: An Application of Machine Learning Algorithms. *St. Theresa Journal of Humanities and Social Sciences*. 7, 91-108.
- Bose, I. ve Mahapatra, R. K. (2011). Business data mining - a machine learning perspective. *Inf. Manage*, 39, 211-225.
- Boser, B. E., Isabelle G.M. ve Vapnik, V.N. (1992, Temmuz, 27-29). *A training algorithm for optimal margin classifiers*. Proceedings of the fifth annual workshop on Computational learning theory, New York, USA.
- Boyd, C. R., Tolson, M. A. ve Copes, W. S. (1987). Evaluating trauma care: The TRISS method - Trauma Score and the Injury Severity Score. *The Journal of Trauma*, 27, 370-378.
- Brout, D., et al. (2022). The Pantheon+ Analysis: Cosmological Constraints. *ApJ*. 938, 110.
- Buhl, H. U., Roglinger, M., Moser, F. ve Heidemann, J. (2013). Big data. *Business Information Systems Engineering*, 5, 65-69.
- Burges, C. J. C. (1998). *A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery*. Kluwer Academic Press.
- Busca, N. G., et al. (2013). Baryon acoustic oscillations in the Ly α forest of BOSS quasars. *A&A*, 552, 18.
- Caldeira, J., Wu, W. L. K., Nord, B., Avestruz, C., Trivedi, S. ve Story, K.T. (2019). DeepCMB: Lensing Reconstruction of the Cosmic Microwave Background with Deep Neural Networks. *Astron. Comput.*, 28, 100307.

- Campbell, C. ve Ying, Y. (2011). *Learning with support vector machines. Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Press.
- Caruana, R. ve Niculescu-Mizil, A. (2004). *Data mining in metric space: an empirical analysis of supervised learning performance criteria*. Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle-WA, USA.
- Cesare, S. ve Xiang, Y. (2012). *Software Similarity and Classification*. Springer.
- Cha, S.-H. (2007). Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *Int. J. Math. Mod. Met. Appl. Sci.*, 1, 300-307.
- Chandrashekar, G. ve Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40, 16-28.
- Chen, G., Xiao, S., Zhang, C. ve Zhao, H. (2023). A Theory-Driven Deep Learning Method for Voice Chat-Based Customer Response Prediction. *Information Systems Research*, Baskıda, <https://doi.org/10.1287/isre.2022.1196>.
- Chetty, M., Ngom, A. ve Ahmad, S. (2008). *Pattern Recognition in Bioinformatics*. Springer.
- Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining*, 10, 35.
- Choi, E., Bahadori, M.T., Schuetz, E., Stewart, W. ve Sun, J. (2016, Ağustos, 19-20). *Doctor AI: Predicting Clinical Events via Recurrent Neural Networks*. Proceedings of the 1st Machine Learning for Healthcare Conference, Los Angeles, USA.
- Clark, P. J. (1952). An extension of the coefficient of divergence for use with multiple characters. *Copeia*, 1952(2), 61-64.
- Clevert, D.-A., Unterthiner, T. ve Hochreiter, S. (2015). *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*, 4th International Conference on Learning Representations (ICLR'16), San Juan, Puerto Rico.
- Cortes, C. ve Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- Cox, R.T. (1946). Probability, Frequency and Reasonable Expectation. *Amer. J. Phys.*, 14(1), 1–13.
- Cramér, H. (1946). A contribution to the theory of statistical estimation. *Skand. Aktuarietidskr. (Scandinavian Actuarial Journal)*, 1946(1), 85-94.
- Cunningham, S. J., Littin, J. ve Witten. I. H. (1997). Applications of machine learning in information retrieval. *Technical Report (University of Waikato)*, 97/6.
- DeCoste, D. (2002). Training Invariant Support Vector Machines. *Machine Learning*, 46, 161–190.
- Delubac, T. et al. (2015). Baryon Acoustic Oscillations in the Ly α forest of BOSS DR11 quasars. *A&A*, 574, A59.
- Deza E. ve Deza, M.M. (2009). *Encyclopedia of distance*. Springer.
- Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3), 297-302.

- Dixon, M. F., Halperin, I. ve Bilokon, P. (2020). *Machine Learning in Finance from Theory to Practice*. Springer Nature Press.
- Donalek, C. (2011). *Supervised and unsupervised learning*. Astronomy Colloquia, California Institute of Technology, USA.
- Domingos, P. ve Pazzani, M. (1996). *Beyond independence: Conditions for the optimality of the simple Bayesian classifier*. Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96), 105-112, Bari, Italy.
- Dozat, T. (2016, Mayıs, 2-4). *Incorporating Nesterov Momentum into Adam*. 4th International Conference on Learning Representations, San Juan, Puerto Rico.
- Duan, X., Wang, L., Zhai, C., Zheng, N., Zhang, Q., Niu, Z. ve Hua, G. (2018, Ekim, 7-10). *Joint Spatio-Temporal Action Localization in Untrimmed Videos with Per-Frame Segmentation*. 25th IEEE International Conference on Image Processing (ICIP), Megaron Athens International Conference Centre, Athens, Greece.
- Duchi, J., Hazan, E. ve Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.*, 12, 2121-2159.
- Eck, D., ve Schmidhuber, J. (2002). Learning the Long-Term Structure of the Blues. Artificial Neural Networks. *Lecture Notes in Computer Science*, 2415, 284-289.
- Escamilla-Rivera, C., Quintero, M. A. C. ve Capozziello, S. (2020). A deep learning approach to cosmological dark energy models. *JCAP*, 2020(03), 008.
- Fernández, S., Graves, A. ve Schmidhuber, J. (2007). *An Application of Recurrent Neural Networks to Discriminative Keyword Spotting*. Proceedings of the 17th International Conference on Artificial Neural Networks (ICANN'07), Berlin, Heidelberg.
- Font-Ribera, A. et al. (2014). Quasar-Lyman α forest cross-correlation from BOSS DR11: Baryon Acoustic Oscillations. *J. Cosmol. Astropart. Phys.*, 05, 027.
- Freedman, W.L., et al. (2001). Final Results from the Hubble Space Telescope Key Project to Measure the Hubble Constant. *ApJ*, 553(1), 47-72.
- Freedman, W.L., et al. (2019). The Carnegie-Chicago Hubble Program. VIII. An Independent Determination of the Hubble Constant Based on the Tip of the Red Giant Branch. *ApJ*, 882, 34.
- Freedman, W.L. (2021). Measurements of the Hubble Constant: Tensions in Perspective. *ApJ*, 919, 16.
- Fulekar, M. H. (2009). *Bioinformatics: Applications in life and environmental sciences*. Springer.
- Gal, Y. ve Ghahramani, Z. (2016). *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. Proceedings of the 33rd International Conference on Machine Learning, New York, USA.
- Gan, G., Ma, C. ve Wu, J. (2007). *Data clustering: Theory, algorithms, and applications*. SIAM.
- Gangwal, R. A Data Scientist's Guide to 8 Types of Sampling Techniques. <https://www.analyticsvidhya.com> adresinden 20 Ocak 2023 tarihinde alındı.

- Gaztanaga, E., Cabre, A. ve Hui, L. (2009). Clustering of luminous red galaxies – IV. Baryon acoustic peak in the line-of-sight direction and a direct measurement of $H(z)$. *Monthly Notices of the Royal Astronomical Society*, 399(3), 1663-1680.
- Ge, J., Bechtold, J. ve Black, J. H. (1997). A New Measurement of the Cosmic Microwave Background Radiation Temperature at $z = 1.97$. *Astrophys. J.*, 474, 67.
- Gers, F.A. ve Schmidhuber, J. (2001). LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages. *IEEE Transactions on Neural Networks*, 12(6), 1333–1340.
- Gers, F., Schraudolph, N. ve Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, 3, 115–143.
- Giusti, R. ve Batista, G. (2013). *An empirical comparison of dissimilarity measures for time series classification*. Brazilian Conference on Intelligent Systems (BRACIS), Fortaleza, IEEE.
- Goodfellow, I., Bengio, Y. ve Courville, A. (2016). *Deep Learning (Adaptive Computation and Machine Learning Series)*, MIT Press.
- Grabusts, P. (2011). *The choice of metrics for clustering algorithms*. Proceedings of the International Scientific and Practical Conference, 70-76, Rezekne Academy of Technologies.
- Graves, A. ve Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6), 602–610.
- Graves, A., Fernández, S., Liwicki, M., Bunke, H. ve Schmidhuber, J. (2007). *Unconstrained Online Handwriting Recognition with Recurrent Neural Networks*. Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS'07), 577-584, Curran Associates Inc., USA.
- Graves, A., et al. (2009). A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(1), 855-868.
- Graves, A. ve Schmidhuber, J. (2009). *Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks*. MIT Press.
- Graves, A., Mohamed, A. ve Hinton, G. (2013). *Speech Recognition with Deep Recurrent Neural Networks*. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6645-6649.
- Grootendorst, M. (2021). 9 Distance Measures in Data Science - The advantages and pitfalls of common distance measures. <https://towardsdatascience.com/adresinden-01-Ocak-2023-tarihinde-alinmistir>.
- Gupta A., Müller A.T., Huisman B.J.H., Fuchs J.A., Schneider P. ve Schneider G. (2018). Generative Recurrent Networks for De Novo Drug Design. *Mol. Inform.*, 37, Article number 1700111.
- Hagstotz, S., Reischke, R. ve Lilow, R. (2022). A new measurement of the Hubble constant using fast radio bursts. *MNRAS*, 511, 662-667.
- Hall, M.A. (1999). *Correlation-based Feature Selection for Machine Learning* [PhD Thesis, The University of Waikato], <https://www.cs.waikato.ac.nz/~mhall/thesis.pdf>

- Hamel, L.H. (2009). *Knowledge Discovery with Support Vector Machines*. Wiley.
- Hamming, R. W. (1958). Error detecting and error correcting codes. *Bell System technical journal*, 131, 147-160.
- Harper, P. R. (2005). A review and comparison of classification algorithms for medical decision making. *Health Policy*, 71, 315-331.
- Hassanat, A. B. (2014). Dimensionality invariant similarity measure. *Journal of American Science*, 10, 221-226.
- Hastie, T., Tibshirani, R. ve Friedman, J. (2008). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd edition)*. Springer-Verlag.
- Hatzigiorgaki, M. ve Skodras, A. (2003). Compressed domain image retrieval: a comparative study of similarity metrics. *Proceedings of SPIE*, 5150, 439-448.
- Haupt, R. L. ve Haupt, S. E. (1998). *Practical Genetic Algorithms*. Wiley-Interscience.
- Heavens, A. F., Jimenez, R. ve Lahav, O. (2000). Massive lossless data compression and multiple parameter estimation from galaxy spectra. *Monthly Notices of the Royal Astronomical Society*, 317, 965-972.
- Hellinger, E. E. (1909). Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *für die reine und angewandte Mathematik*, 136, 210-271.
- Hinshaw, G., et al. (2009). Five-Year Wilkinson Microwave Anisotropy Probe Observations: Data Processing, Sky Maps, and Basic Results. *ApJS*, 180, 225-245.
- Hochreiter, S., Heusel, M. ve Obermayer, K. (2007). Fast model-based protein homology detection without alignment. *Bioinformatics*, 23, 1728-1736.
- Hosmer, D.W. ve Lemeshow, S. (2000). *Applied Logistic Regression*. Wiley-Interscience.
- Hossain, S. (2020). A complete guide to sampling techniques. <https://hotcubator.com.au> adresinden 20 Ocak 2023 tarihinde alınmıştır.
- Huang, J., Zhou, W., Zhang, Q., Li, H. ve Li, W. (2018). Video-based Sign Language Recognition without Temporal Segmentation. *e-Print*, 1801.10111 [cs.CV].
- Hubble, E. (1929). A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae. *Proc. Nat. Acad. Sci.*, 15, 168-173.
- Hurier, G., Aghanim, N., Douspis, M. ve Pointecouteau, E. (2014). Measurement of the T_{CMB} evolution from the Sunyaev-Zel'dovich effect. *Astron. Astrophys.*, 561, Article number A143.
- Hussain, A. (2020). K-Nearest Neighbors (KNN) and its Applications. <https://medium.com> adresinden 02 Şubat 2023 tarihinde alınmıştır.
- Ibrahim, S.K., Ahmed, A., Zeidan, M.A.E. ve Ziedan, I.E. (2019). Machine Learning Methods for Spacecraft Telemetry Mining. *IEEE Trans. Aerosp. Electron. Syst.*, 55, 1816-1827.
- Ilham, A. et al. (2019). Long-term deposits prediction: a comparative framework of classification model for predict the success of bank telemarketing. *IOP Conf. Series - Journal of Physics: Conf. Series*, 1175, 012035.

- Islam, M. S., Arifuzzaman, M. ve Islam, M. S. (2019). *SMOTE Approach for Predicting the Success of Bank Telemarketing*. 4th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON), Bangkok, Thailand.
- Jaccard, P. (1901). Etude comparative de la distribution orale dans une portion des Alpes et du Jura. *Bulletin de la Société vaudoise des sciences naturelles*, 547-579, Impr. Corbaz, Lausanne.
- Jaeger, T. de, et al. (2022). A 5 per cent measurement of the Hubble-Lemaître constant from Type II supernovae. *MNRAS*, 514, 4620-4628.
- Jain, A., Patel, H., Nagalapatti, L., et al. (2020, Temmuz, 6-10). *Overview and Importance of Data Quality for Machine Learning Tasks*. KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 3561–3562, Virtual Event, USA.
- Jayaswal, V. (2020). K-Nearest Neighbors (KNN) algorithm. [https://towardsdatascience.com/adresinden 02 Şubat 2023 tarihinde alınmıştır](https://towardsdatascience.com/adresinden-02-Şubat-2023-tarihinde-alınmıştır).
- Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 186, 453-461.
- Jia, R. ve Liang, P. (2016). Data Recombination for Neural Semantic Parsing, *e-Print*, 1606.03622 [cs.CL].
- Jimenez, R., Verde, L., Treu, T. ve Stern, D. (2003). Constraints on the equation of state of dark energy and the Hubble constant from stellar ages and the CMB. *ApJ*, 593, 622-629.
- Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. *Lecture Notes in Computer Science*, 1398, 137–142.
- John G. H. ve Langley, P. (1996). *Static versus dynamic sampling for data mining*. AAAI Press.
- Kalaycı, Ş. (2008). *SPSS Uygulamalı Çok Değişkenli İstatistik Teknikleri*. Asil Yayın.
- Kanade, V. (2022). What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices. [https://www.spiceworks.com adresinden 30 Ocak 2023 tarihinde alınmıştır](https://www.spiceworks.com/adresinden-30-Ocak-2023-tarihinde-alınmıştır).
- Kangal, E.E., Salti, M. ve Aydogdu, O. (2019). Machine learning algorithm in a caloric view point of cosmology. *Phys. Dark Univ.*, 26, 100369.
- Keren, G. ve Schuller, B. (2016, Temmuz, 24-29). *Convolutional RNN: an Enhanced Model for Extracting Features from Sequential Data*. International Joint Conference on Neural Networks (IJCNN), pp. 3412-3419, Vancouver, Canada.
- Kendall, M.G. ve Stuart, A. (1969). *The Advanced Theory of Statistics II*. Griffin Press.
- Kim, K.-H., Lee, C.-S., Jo, S.-M. ve Cho, S.-B. (2015, Kasım, 13-15). *Predicting the success of bank telemarketing using deep convolutional neural network*. 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR), Fukuoka, Japan.
- Kingma, D.P. ve Ba, J. (2015). *Adam: A Method for Stochastic Optimization*. 3rd International Conference on Learning Representations (ICLR), San Diego-CA, USA.

- Klambauer, G., Unterthiner, T., Mayr, A. ve Hochreiter, S. (2017). *Self-Normalizing Neural Networks*. 31st Conference on Neural Information Processing Systems (NIPS), Long Beach-CA, USA.
- Koçoğlu, F.Ö. ve Esnaf, Ş. (2022). Machine Learning Approach and Model Performance Evaluation for Tele-Marketing Success Classification. *International Journal of Business Analytics (IJBAN)*, 9(5), 1-18
- Kohavi, R. (1995). *Wrappers for Performance Enhancement and Oblivious Decision Graphs* [Doctor of Philosophy, Stanford University], <http://i.stanford.edu/pub/cstr/reports/cs/tr/95/1560/CS-TR-95-1560.pdf>
- Kohavi, R. ve John G. (1996). Wrappers for feature subset selection. *Artificial Intelligence (Special issue on relevance)*, 97(1-2), 273-324.
- Kotler, P. ve Keller, K. L. (2015). *Framework for Marketing Management*. Pearson.
- Kowsari, K. et al. (2019). Text Classification Algorithms: A Survey. *Information*, 10(4), Article number 150.
- Krawczyk, R., Colombo, T., Neufeld, N., Pisani, F. ve Valat, S. (2022). Ethernet for High-Throughput Computing at CERN. *IEEE Transactions on Parallel and Distributed Systems*, 33, 3640-3650.
- Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images* [M.Sc. thesis, University of Toronto], <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.
- Kullback, S. ve Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22, 79-86.
- Kumar, R. ve Verma, R. (2012). Classification Algorithms for Data Mining: A Survey. *International Journal of Innovations in Engineering and Technology*, 1, 7-14.
- Kumar, J., Goomer, R. ve Singh, A.K. (2018). Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters. *Procedia Comput. Sci.*, 125, 676-682.
- Kumar, A. (2020). SVM Algorithm as Maximum Margin Classifier. <https://vitalflux.com> adresinden 29 Ocak 2023 tarihinde alınmıştır.
- Lance, G. N. ve Williams, W. T. (1967). Mixed-data classificatory programs I - Agglomerative systems. *Australian Computer Journal*, 1, 15-20.
- Langley, P. ve Sage, S. (1994a). *Scaling to domains with irrelevant features (Computational Learning Theory and Natural Learning Systems)*, MIT Press.
- Langley, P. ve Sage, S. (1994b). *Oblivious decision trees and abstract cases*. AAAI Press.
- Langley, P. ve Sage, S. (1994c, Temmuz, 29-31). *Induction of selective Bayesian classifiers*. Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI1994), 399-406, Seattle-WA, USA.
- Langley P. ve Simon, H. A. (1995). Applications of machine learning and rule induction. *Communications of the ACM*, 38(1), 55-64.
- Legendre, P. ve Legendre, L. (2012). *Numerical ecology*. Elsevier.

- LIGO Scientific, VIRGO Coll., et al. (2017). A gravitational-wave standard siren measurement of the Hubble constant. *Nature*, 551, 85-88.
- Lopez-Corredoira, M., Gutiérrez, C.M. ve Genova-Santos, R.T. (2017). Analysis of the Amplitude of the Sunyaev–Zel’dovich Effect out to Redshift $z = 0.8$. *Astrophys. J.*, 840, 62.
- Lu, L., Sargent, W.L.W., Barlow, T.A., Churchill, C.W. ve Vogt, S. S. (1996a). Abundances at High Redshifts: The Chemical Enrichment History of Damped Ly alpha Galaxies. *ApJS*, 107, 475.
- Lu, L., Sargent, W.L.W., Womble, D.S. ve Barlow, T.A. (1996b). Properties of a High-Redshift Galaxy at $z = 4.4$. *Astrophys. J.*, 457, L1.
- Luzzi, G., Shimon, M., Lamagna1, L., Rephaeli, Y., De Petris, M., Conte, A., De Gregori, S. ve Battistelli, E.S. (2009). Redshift Dependence of the Cosmic Microwave Background Temperature From Sunyaev–Zeldovich Measurements. *Astrophys. J.*, 705, 1122.
- Maas, A. L., Hannun, A.Y. ve Ng, A.Y. (2013, Temmuz, 16-21). *Rectifier Nonlinearities Improve Neural Network Acoustic Models*. Proceedings of the 30th International Conference on Machine Learning, Atlanta, USA.
- Macklem, M. (2002). *Multidimensional Modelling of Image Fidelity Measures*. Simon Fraser University Press.
- Mahesh, B. (2020). Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR)*, 9, 381-386.
- Maitra, D.S., Bhattacharya, U. ve Parui, S.K. (2015, Ağustos, 23-26). *CNN based common approach to handwritten character recognition of multiple scripts*. 13th International Conference on Document Analysis and Recognition (ICDAR), Nancy, France.
- Maity, A. (2016). Supervised Classification of RADARSAT-2 Polarimetric Data for Different Land Features. *e-Print*, 1608.00501 [cs.CV].
- Malakoff, D. (1999). Bayes offers a 'new' way to make sense of numbers. *Science*, 286, 1460-1463.
- Marron, B. A. ve de Maine, P. A. D. (1967). Automatic data compression. *Communications of the ACM*, 10, 711–715.
- Mather, J. C., Fixsen, D. J., Shafer, R. A., Mosier, C. ve Wilkinson, D. T. (1999). Calibrator Design for the COBE Far-Infrared Absolute Spectrophotometer (FIRAS). *Astrophys. J.*, 512, 511-520.
- Mayer, H., Gomez, F., Wierstra, D., Nagy, I., Knoll, A. ve Schmidhuber, J. (2006). A System for Robotic Heart Surgery that Learns to Tie Knots Using Recurrent Neural Networks. *IEEE/RSJ*, 543–548.
- Menard, S. (2002). *Applied Logistic Regression Analysis*. Sage Publication.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
- Mohamed, A. Data Analysis for Telemarketing Case. www.kaggle.com adresinden 04 Ocak 2023 tarihinde alınmıştır.

- Molaro, P., et al. (2002). The cosmic microwave background radiation temperature at $z_{\text{abs}} = 3.025$ toward QSO 0347-3819. *A&A*, 381, L64-L67.
- Moro, S., Cortez, P. ve Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62, 22-31.
- Moresco, M., Verde, L., Pozzetti, L., Jimenez, R. ve Cimatti, A. (2012). New constraints on cosmological parameters and neutrino properties using the expansion rate of the Universe to $z \sim 1.75$. *J. Cosmol. Astropart. Phys.*, 7, Article number 053.
- Moresco, M. (2015). Raising the bar: new constraints on the Hubble parameter with cosmic chronometers at $z \sim 2$, *Monthly Notices of the Royal Astronomical Society: Letters*, 450, L16-L20.
- Moresco, M., et al (2016). A 6% measurement of the Hubble parameter at $z \sim 0.45$: direct evidence of the epoch of cosmic re-acceleration. *J. Cosmol. Astropart. Phys.*, 5, 014.
- Muller, S., Beelen, A., Black, J. H., Curran, S. J., Horellou, C. Aalto, S., Combes, F., Guélin, M. ve Henkel, C. (2013). A precise and accurate determination of the cosmic microwave background temperature at $z=0.89$. *A&A*, 551, A109.
- Mustapha, S.M.F.D.S. ve Alsufyani, A. (2019). Application of Artificial Neural Network and Information Gain in Building Case-Based Reasoning for Telemarketing Prediction. *International Journal of Advanced Computer Science and Applications*, 10, 300-306.
- Nair, V. ve Hinton, G.E. (2010, Temmuz, 21-24). *Rectified Linear Units Improve Restricted Boltzmann Machines*. Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel.
- Nesseris, S. ve Garcia-Bellido, J. (2012). A new perspective on dark energy modeling via genetic algorithms. *J. Cosmol. Astropart. Phys.*, 1211, 033.
- Nesseris, S. ve Shafieloo, A. (2010). A model-independent null test on the cosmological constant. *Mon. Not. R. Astron. Soc.*, 408, 1879–1885.
- Nesterov, Y. (1983). A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady ANSSSR (Soviet. Math. Doct.)*, 269, 543–547.
- Neyman, J. (1949). Contributions to the theory of the Chi-Squared test. *Proceedings of the first Berkeley symposium on mathematical statistics and probability*, 1, 239-273.
- Noterdaeme, P., Petitjean, P., Ledoux, C., López, S., Srianand, R. ve Vergani, S. D. (2010). A translucent interstellar cloud at $z = 2.69$. *A&A*, 523, A80.
- Noterdaeme, P., Petitjean, P., Srianand, R., Ledoux, C. ve López, S. (2011). The evolution of the cosmic microwave background temperature. Measurements of T_{CMB} at high redshift from carbon monoxide excitation. *A&A*, 526, L7.
- Ntampaka, M., Trac, H., Sutherland, D. J., Battaglia, N., Póczos, B. ve Schneider, J. (2015). A Machine Learning Approach for Dynamical Mass Measurements of Galaxy Clusters. *Astrophys. J.*, 803, 50.

- Orloci, L. (1967). An agglomerative method for classification of plant communities. *Journal of Ecology*, 55, 193-206.
- Orsini, F., Gastaldi, M., Mantecchini, L. ve Rossi, R. (2019). Neural networks trained with WiFi traces to predict airport passenger behavior. *e-Print*, 1910.14026.
- Overbye, D. (1999). *Lonely Hearts of the Cosmos: The Story of the Scientific Quest for the Secret of the Universe*, HarperCollins.
- Özdamar, K. (2004). *Paket Programlar ile İstatiksel Veri Analizi*. Kaan Kitabevi.
- Palei, S. K. ve Das, S. K. (2009). Logistic regression model for prediction of roof fall risks in bord and pillar workings in coal mines: An approach. *Safety Science*, 47, 88–96.
- Patel, B. ve Meshram, B. (2012). Content based video retrieval systems. *International Journal of UbiComp*, 3, 13-30.
- Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50, 157-175.
- Perez-Ortiz, J. A., Gers, F. A., Eck, D. ve Schmidhuber, J. (2003). Kalman filters improve LSTM network performance in problems unsolvable by traditional recurrent nets. *Neural Networks*, 16, 241-250.
- Pesce, D.W., et al. (2020). The Megamaser Cosmology Project. XIII. Combined Hubble Constant Constraints. *ApJL*, Vol. 891, L1.
- Piatetsky-Shapiro, G. ve Frawley, W. J. E. (1991). *Knowledge Discovery in Databases*. MIT Press.
- Pinto, D., Benedi, J.-M. ve Rosso, P. (2007). *Clustering narrow-domain short texts by using the Kullback-Leibler distance*. International Conference on Intelligent Text Processing and Computational Linguistics, 611-622, Mexico City, Mexico.
- Pradhan, S.S., et al. (2004). Shallow Semantic Parsing using Support Vector Machines. *Association for Computational Linguistics*, 233-240.
- Prasath, V.B.S., et al. (2019). Effects of Distance Measure Choice on KNN Classifier Performance - A Review. *e-Print*. 1708.04321v3 [cs.LG].
- Premaratne, P. (2014). *Human computer interaction using hand gestures*. Springer.
- Press, W.H., et al. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press.
- Quinlan, R. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81-106.
- Quinlan, J. R. (1987, Ağustos, 23-28). *Generating production rules from decision trees*. Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 304-307, Milano, Italy.
- Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4, 77–90.

- Ramakrishnan, R. ve von Lilienfeld, O. A. (2017). *Chapter 5- Machine Learning, Quantum Chemistry, and Chemical Space (Reviews in Computational Chemistry)*, 225–256, Wiley-VCH.
- Rao, C.R. (1945). Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37, 81-91.
- Ratsimbazafy, A.L, et al. (2017). Age-dating luminous red galaxies observed with the Southern African Large Telescope. *MNRAS*, 467, 3239-3254
- Reddi, S.J., Kale, S. ve Sanjiv, K. (2018). *On the Convergence of Adam and Beyond*. 6th International Conference on Learning Representations, Vancouver, Canada.
- Riess, A.G., et al. (2019). Large Magellanic Cloud Cepheid Standards Provide a 1% Foundation for the Determination of the Hubble Constant and Stronger Evidence for Physics beyond Λ CDM. *ApJ*, 876, 85.
- Riess, A.G., et al. (2022). A Comprehensive Measurement of the Local Value of the Hubble Constant with $1 \text{ km s}^{-1} \text{ Mpc}^{-1}$ Uncertainty from the Hubble Space Telescope and the SH0ES Team. *ApJL*, 934, L7.
- Rincy, N.T. ve Gupta, R. (2020, Şubat, 22-23). *A Survey on Machine Learning Approaches and Its Techniques*. IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India.
- Roth K.C. ve Bauer, J.M. (1999). The $z = 1.6748$ C I Absorber toward PKS 1756+237. *Astrophys. J.*, 515, L57.
- Saiful I. ve Hossain, E. (2020). Foreign Exchange Currency Rate Prediction using a GRU-LSTM Hybrid Network. *Soft Computing Letters*, 3, 100009.
- Sak, H., Senior, A. ve Beaufays, F. (2014, Eylül, 14-18). *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*. 15th Annual Conference of the International Speech Communication Association, 338-342, Singapore.
- Salti, M., Kangal, E.E. ve Aydogdu, O. (2021). Evolution of CMB temperature in a Chaplygin gas model from deep learning perspective. *Astronomy and Computing*, 37, 100504.
- Salti M. ve Kangal, E.E. (2022). Deep learning of CMB radiation temperatur. *Annals of Physics*, 439, 168799.
- Salti M., Kangal, E.E., Ciger, E. ve Zengin B. (2022). Data-driven predictive modeling of Hubble parameter. *Physica Scripta*, 97, 085011.
- Salti M., Kangal, E. ve Zengin, B. (2023). Deep learning assisted Hubble parameter analysis. Bir dergide inceleme aşamasındadır.
- Samushia, L., et al. (2013). The clustering of galaxies in the SDSS-III DR9 Baryon Oscillation Spectroscopic Survey: testing deviations from Λ and general relativity using anisotropic clustering of galaxies. *MNRAS*, 429, 1514-1528.

- Schmidhuber, J., Gers, F. ve Eck, D. (2002). Learning nonregular languages: A comparison of simple recurrent networks and LSTM. *Neural Computation*, 14, 2039-2041.
- Shahnaz, F. (2006). *Decision Tree Based Algorithms (Lecture Notes in Data Mining)*, World Scientific Publisher.
- Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27, 379-423.
- Sharma, D. ve Kumar, N. (2017). A Review on Machine Learning Algorithms, Tasks and Applications. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 6, 1548-1552.
- Shirkhorshidi, A. S., Aghabozorgi, S. ve Wah, T. Y. (2015). A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data. *PloS one*, 10, e0144059.
- Sibson, R. (1969). Information Radius. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 14, 149-160.
- Sidhu, R. (2019). Layman's Introduction to KNN. <https://towardsdatascience.com> adresinden 20 Şubat 2023 tarihinde alınmıştır.
- Simon, J., Verde, L. ve Jimenez, R. (2005). Constraints on the redshift dependence of the dark energy potential. *Phys. Rev. D*, 71, 123001.
- Simplilearn (2022). Support Vector Machine (SVM) in R: Taking a Deep Dive. <https://www.simplilearn.com> adresinden 20 Ocak 2023 tarihinde alınmıştır.
- Singh, A. (2020). Calculating Accuracy of an ML Model. <https://medium.com> adresinden 05 Şubat 2023 tarihinde alınmıştır.
- Soheila, E. (2000). *The enhancement of credit card fraud detection systems using machine learning methodology*. [M.A. Sci. Thesis, The University of Toronto/Canada], <https://tspace.library.utoronto.ca/handle/1807/16018>.
- Soltis, J., Casertano, S. ve Riess, A.G. (2021). The Parallax of ω Centauri Measured from Gaia EDR3 and a Direct, Geometric Calibration of the Tip of the Red Giant Branch and the Hubble Constant. *ApJL*, 908, L5.
- Songaila, A., Cowie, L.L., Hogan, C. ve Rugers, M. (1994a). Deuterium abundance and background radiation temperature in high-redshift primordial clouds. *Nature*, 368, 599-604.
- Songaila, A., Cowie, L. L., Vogt, S., Keane, M., Wolfei, A. M., Hu, E. M., Oren, A. L., Tytleri, D.-R. ve Lanzetta, K. M. (1994b). Measurement of the microwave background temperature at a redshift of 1.776. *Nature*, 371, 43-45.
- Sorensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, 5, 1-34.

- Srianand, R., Petitjean, P. ve Ledoux, C. (2000). The cosmic microwave background radiation temperature at a redshift of 2.34. *Nature*, 408, 931-935.
- Srianand, R., Noterdaeme, P., Ledoux, C. ve Petitjean, P. (2008). First detection of CO in a high-redshift damped Lyman- α system. *A&A*, 482, L39.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I ve Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15, 1929-1958.
- Stern, D., Jimenez, R., Verde, L., Kamionkowski, M. ve Stanford, S.A. (2010). Cosmic chronometers: constraining the equation of state of dark energy. I: H(z) measurements. *J. Cosmol. Astropart. Phys.* 02, 008.
- Stirnberg, R., Cermak, J., Kotthaus, S., Haeffelin, M., Andersen, H., Fuchs, J., Kim, M., Peti, J.-E. ve Favez, O. (2021). Meteorology-driven variability of air pollution (PM1) revealed with explainable machine learning. *Atmos. Chem. Phys.*, 21, 3919-3948.
- Strano, M. ve Colosimo, B.M. (2006). Logistic regression analysis for experimental determination of forming limit diagrams. *International Journal of Machine Tools and Manufacture*, 46, 673-682.
- Suarjaya, I. M. A. D. (2012). A New Algorithm for Data Compression Optimization. *Int. J. Adv. Comp. Sci. and Appl.*, 3, 14-17.
- Suresha, H.P. (2021). Sampling Techniques - Statistical approach in Machine learning. <https://medium.com> adresinden 20 Ocak 2023 tarihinde alındı.
- Szmidt, E. (2013). *Distances and similarities in intuitionistic fuzzy sets*. Springer.
- Taneja, I. J. (1995). New developments in generalized information measures. *Advances in Imaging and Electron Physics*, 91, 37-135.
- Tax, N., Verenich, I., La Rosa, M. ve Dumas, M. (2017). Predictive Business Process Monitoring with LSTM neural networks. *Lecture Notes in Computer Science*, 10253, 477-492.
- Tegmark, M., Taylor, A. ve Heavens, A. (1997). Karhunen-Loeve eigenvalue problems in cosmology: how should we tackle large data sets? *Astrophys. J.*, 480, 22.
- Tékouabou, S.C.K., Gherghina, Ş.C., Touluni, H., Mata, P.N., Mata, M.N. ve Martins, J.M. (2022). A Machine Learning Framework towards Bank Telemarketing Prediction. *J. Risk Financial Manag.* 15, 269.
- Thireou, T. ve Reczko, M. (2007). Bidirectional Long Short-Term Memory Networks for predicting the subcellular localization of eukaryotic proteins. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4, 441-446
- Thomas N. L., Chapelle, O., Weston, J. ve Elisseeff, A. (2006). *Chapter: Embedded Methods in Feature Extraction, Studies in Fuzziness and Soft Computing Book Series (STUDFUZZ)*, 207, 137-165, Springer.
- Thornton, C. J. (1992). *Techniques In Computational Learning*. Chapman and Hall.

- Tilaver, H., Salti, M., Aydogdu, O. and Kangal, E.E. (2021). Deep learning approach to Hubble parameter. *Comp. Phys. Commun.*, 261, 107809.
- Todeschini, R., Ballabio, D. ve Consonni, V. (2015). *Distances and other dissimilarity measures in chemometrics (Encyclopedia of Analytical Chemistry: Applications, Theory and Instrumentation)*, Wiley.
- Topsoe, F. (2000). Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on information theory*, 46, 1602-1609.
- Turban, E., Sharda, R. ve Delen, D. (2011). *Decision Support and Business Intelligence Systems*. Pearson.
- Tully, R.B., Courtois, H.M. ve Sorce, J.G. (2016). COSMICFLOWS-3. *ApJ*, 152, 50.
- Urtnasan, E., Park, J.U. ve Lee, K.J. (2020). Automatic detection of sleep-disordered breathing events using recurrent neural networks from an electrocardiogram signal. *Neural Comput. Appl.*, 32, 4733–4742.
- Vapnik, V.N. ve Chervonenkis, A. (1974). *Theorie der Zeichenerkennung (Theory of Pattern Recognition)*. Akademie-Verlag.
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.
- Verleysen, M. ve Francois, D. (2005). The Curse of Dimensionality in Data Mining and Time Series Prediction. *Lecture Notes in Computer Science*, 3512, 758–770.
- Vogt, N. (2018). Machine learning in neuroscience. *Nature Methods*, 15, 33.
- Wang, Y., et al. (2017). The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: tomographic BAO analysis of DR12 combined sample in configuration space. *MNRAS*, 469, 3762-3774.
- Wang, L., Duan, X., Zhang, Q., Niu, Z., Hua, G. ve Zheng, N. (2018). Segment-Tube: Spatio-Temporal Action Localization in Untrimmed Videos with Per-Frame Segmentation. *Sensors*, 18, 1657.
- Whittaker, R. H. (1952). A study of summer foliage insect communities in the Great Smoky Mountains. *Ecological monographs*, 22, 1-44.
- Wierstra, D., Schmidhuber, J. ve Gomez, F. J. (2005). *Evolino: Hybrid Neuroevolution/Optimal Linear Search for Sequence Learning*. Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), 853–858, Edinburgh, UK.
- Willett, P., Barnard, J. M. ve Downs, G. M. (1998). Chemical similarity searching. *Journal of chemical information and computer sciences*, 38, 983-996.
- Williams, W. T. ve Lance, G. N. (1966). Computer programs for hierarchical polythetic classification (similarity analyses). *The Computer*, 9, 60-64.
- Wolz, L., Kilbinger, M., Weller, J. Ve Giannantonio, T. (2012). On the validity of cosmological Fisher matrix forecasts. *J. Cosmol. Astropart. Phys.*, 09, 009.

- Xie, C., Zhang, J.-L., Zhu, Y., Xiong, B. ve Wang, G.-J. (2023). How to improve the success of bank telemarketing? Prediction and interpretability analysis based on machine learning. *Computers & Industrial Engineering*, 175, 108874.
- Yakut, E. ve Gemici, E. (2017). LR, C5.0, CART, DVM Yöntemlerini Kullanarak Hisse Senedi Getiri Sınıflandırma Tahmini Yapılması ve Kullanılan Yöntemlerin Karşılaştırılması: Türkiye’de BIST’de Bir Uygulama. *Ege Academic Review (Ege Akademik Bakış)*, 17, 461-479.
- Yu H., Ratra, B. ve Wang, F.-Y. (2018). Hubble Parameter and Baryon Acoustic Oscillation Measurement Constraints on the Hubble Constant, the Deviation from the Spatially Flat Λ CDM Model, the Deceleration–Acceleration Transition Redshift, and Spatial Curvature. *ApJ*, 856, 3.
- Yu, Y., Si, X., Hu, C. ve Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, 31, 1235–1270.
- Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *e-Print*, 1212.5701 [cs.LG].
- Zhang, Y. (2010). *New Advances in Machine Learning*. InTech Open.
- Zhang, C., Zhang, H., Yuan, S., Liu, S., Zhang, T.-J. ve Sun Y.-C. (2014). Four New Observational Data From Luminous Red Galaxies of Sloan Digital Sky Survey Data Release Seven. *Research in Astronomy and Astrophysics*, 14, 1221-1233.
- Zhaoping, L. (2006). Theoretical Understanding of the early visual processes by data compression and data selection. *Network: Computation in Neural Systems*, 17, 301-334.
- Zhao, Z., Chen, W., Wu, X., Chen, P.C.Y. ve Liu, J. (2017). LSTM network: A deep learning approach for Short-term traffic forecast. *IET Intelligent Transport Systems*, 11, 68-75.
- Zhou, T., Chan, K. C. ve Wang, Z. (2008). *TopEVM: using co-occurrence and topology patterns of enzymes in metabolic networks to construct phylogenetic trees*. Springer.
- Zhu, X., Suk, H.-I. ve Shen, D. (2014). A Novel Matrix-Similarity Based Loss Function for Joint Regression and Classification in AD Diagnosis. *NeuroImage*, 100, 91-105.

BENZERLİK RAPORU ÖZET SAYFASI

19.03.2023 20:49

Summary Report

MehmetS6_TEZ_Güncel.pdf

As of: Mar 19, 2023 8:34:20 PM
41,006 words - 234 matches - 70 sources

Similarity Index

7%

Mode:

sources:

575 words / 2% - Internet from 16-Mar-2014 12:00AM
www.jasstudies.com

420 words / 1% - Internet
[Erkin, Alperen, Bostancı, Özgür, Kabadayı, Menderes, Çon, Musa, Çankaya, Soner, Taşmektepligil, M. Yalcın. "A COMPARATIVE ANALYSIS OF THE FACTORS THAT INFLUENCE THE BRAND PREFERENCES OF UNIVERSITY STUDENTS STUDYING SPORTS IN TURKEY AND PORTUGAL", Open Access Publishing Group, 2017](#)

245 words / 1% - Internet from 24-Aug-2022 12:00AM
www.researchgate.net

176 words / 1% - Internet from 23-Dec-2022 12:00AM
1library.net

21 words / < 1% match - Internet from 27-Jun-2022 12:00AM
www.researchgate.net

10 words / < 1% match - Internet from 14-Feb-2023 12:00AM
www.researchgate.net

8 words / < 1% match - Internet from 01-Oct-2022 12:00AM
www.researchgate.net

72 words / < 1% match - Crossref
[Azer Celikten, Hasan Bulut, "Turkish Medical Text Classification Using BERT", 2021 29th Signal Processing and Communications Applications Conference \(SIU\), 2021](#)

49 words / < 1% match - Internet from 13-Nov-2021 12:00AM
tr.wiki.the.ru

45 words / < 1% match - Crossref
[M. Salti, E.E. Kangal, O. Aydogdu, "Evolution of CMB temperature in a Chaplygin gas model from deep learning perspective", Astronomy and Computing, 2021](#)

ÖZGEÇMİŞ

Adı ve Soyadı : Mehmet SALTİ

Doğum Tarihi :

E-mail :

Öğrenim Durumu :

<u>Derece</u>	<u>Bölüm/Program</u>	<u>Üniversite</u>	<u>Yıl</u>
Lisans	İşletme Bölümü	Anadolu Üniversitesi	2020
	Matematik Bölümü	Mersin Üniversitesi	2021
Yüksek Lisans	İşletme Bilgi Yönetimi Anabilim Dalı	Mersin Üniversitesi	2023
Doktora	-	-	-

Görevler :

<u>Görev Unvanı</u>	<u>Görev Yeri</u>	<u>Yıl</u>
-	-	-

ESERLER

- Mehmet Saltı ve Evrim Ersin Kangal,
“Deep learning of CMB radiation temperature”,
Annals of Physics 439 (2022) 168799.
- Mehmet Saltı, Emel Ciğer, Evrim Ersin Kangal ve Bilgin Zengin,
“Data-driven predictive modeling of Hubble parameter”,
Physica Scripta 97 (2022) 085011.
- Mehmet Saltı, Evrim Ersin Kangal ve Bilgin Zengin,
“Deep learning assisted Hubble parameter analysis”,
Uluslararası bir dergiye gönderilmiştir (2023).
- Mehmet Saltı, Evrim Ersin Kangal ve Bilgin Zengin,
“Machine Learning Approach Towards Telemarketing Estimation”,
Uluslararası bir dergiye gönderilmiştir (2023).