E-ticaret Sektöründe Müşteri Kaybının Yapay Öğrenme Teknikleri ile Tahminlenmesi

Kübra Yazır

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Ekim 2022

Churn Customer Prediction in the E-commerce Industry with Machine Learning Techniques

Kübra Yazır

MASTER OF SCIENCE THESIS

Department of Industrial Engineering

October 2022

E-ticaret Sektöründe Müşteri Kaybının Yapay Öğrenme Teknikleri ile Tahminlenmesi

Kübra Yazır

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Endüstri Mühendisliği Anabilim Dalı
Üretim ve Servis Sistemleri Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Şerafettin Alpay

Ekim 2022

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Şerafettin Alpay danışmanlığında hazırlamış olduğum "E-ticaret Sektöründe Müşteri Kaybının Yapay Öğrenme Teknikleri ile Tahminlenmesi" başlıklı YÜKSEK LİSANS tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim.

04/11/2022

Kübra Yazır

İmza

ÖZET

Teknolojinin gelişmesi ve internetin etkin bir şekilde kullanılması birçok durumu etkilediği gibi alışverişi de etkilemiştir. Bu durum tüketici davranışının doğasında temel değişimlere yol açmış ve internet alışverişine hız kazandırmıştır. Özellikle pandeminin tüm dünyayı etkisi altına aldığı süreçte internet alışverişine talep artmış ve bu süreçle birlikte e-ticaret sektörü canlanmıştır. E-ticaretin tüketicinin hayatını kolaylaştırmasıyla tüketiciler artık internet alışverişini daha fazla tercih eder hale gelmişlerdir. E-ticaret sektörünün hayatımızda geniş bir yer tutmasıyla e-ticaret firmaları arasında da rekabet ortamı oluşmaya başlamıştır. Bu sebeple firmalarda müşteriyi kaybetme endişesi oluşmuş ve bu probleme karşı çözüm aramaya başlamışlardır. Müşterileri elde tutmanın yenilerini elde etmekten çok daha rasyonel olması sebebiyle, rakip firmalar için müşteri kaybı yönetimi hayati bir önem kazanmıştır. Müşteriyi elde tutabilmek için e-ticaret firmalarının müşteri kaybını tahmin edebilmesi ve müşteri kaybı ile kontrolleri altındaki faktörler arasında bağlantılar kurabilmesi büyük önem teşkil etmektedir.

Bu çalışmada önde gelen bir e-ticaret firmasının açık kaynaklı olan veri seti kullanılarak makine öğrenmesi algoritmaları aracılığıyla müşteri kaybı tahmini modeli geliştirilmiştir. Kodlama için Python programlama dili kullanılmış ve PyCharm uygulama geliştirme ortamında kodlama yapılmıştır. Veri setinde tespit edilen dengesizlik durumunu ortadan kaldırmak için SMOTE metodu uygulanmış ve veri seti dengeli hale getirilmiştir. Tahmin modeli geliştirmek için makine öğrenmesine ait sınıflandırma algoritmaları olan Lojistik Regresyon, K-En Yakın Komşu, Destek Vektör Makine, Karar Ağaçları, Rastgele Orman, XGBoost ve LightGBM kullanılmış ve belirlenen performans kriterlerine (doğruluk oranı(accuracy) ve AUC değerine göre kıyaslanmıştır. Yapılan çalışma sonucunda hem doğruluk oranı hem de AUC değerine göre en yüksek performansı gösteren ve uygulanabilir yapay öğrenme sınıflandırma modeli, LightGBM olarak seçilmiştir. Son olarak en iyi modele göre öznitelik seçimi yapılmış ve model anlamlı hale getirilmiştir.

Anahtar Kelimeler*:* E-ticaret, Müşteri Kaybı, Yapay Öğrenme, Sınıflandırma, SMOTE, Öznitelik Seçimi

SUMMARY

The development of technology and the effective use of the internet have affected many situations as well as shopping. This situation has led to fundamental changes in the nature of consumer behavior and accelerated internet shopping. Especially in the period when the pandemic affected the whole world, the demand for internet shopping increased and with this process, the e-commerce sector revived. With e-commerce facilitating the life of the consumer, consumers have become more preferable to online shopping. With the e-commerce sector taking a large place in our lives, a competitive environment has started to emerge among e-commerce companies. For this reason, companies have been worried about losing customers and they have started to look for a solution to this problem. Customer churn management has gained vital importance for competitors, as retaining customers is much more rational than acquiring new ones. In order to retain customers, it is of great importance for e-commerce companies to be able to predict customer loss and to establish links between customer loss and factors under their control.

In this study, a customer loss estimation model was developed by using machine learning algorithms using the open-source data set of a leading e-commerce company. By choosing machine learning, instead of writing programs that take days, it is aimed to feed an algorithm with data and automate all business processes. Python software was used for coding and coding was done in PyCharm interface. Since there was an imbalance in the data set, the SMOTE method was applied and the data set was balanced. To develop the prediction model, the classification algorithms of machine learning, Logistic Regression, K-Nearest Neighbor, Support Vector Machine, Decision Trees, Random Forest, XGBoost and LightGBM were used and compared according to the determined performance criteria accuracy and AUC value. As a result of the study, LightGBM was chosen as the applicable machine learning classification model, which showed the highest performance in terms of both accuracy and AUC value. Finally, the feature selection was made according to the best model and the model was made meaningful.

Key Words: E-Commerce, Churn Customer, Machine Learning, Classification, SMOTE, Feature Selection

İÇİNDEKİLER

Sayfa

ÖZET vi
SUMMARY vii
TEŞEKKÜR viii
İÇİNDEKİLER ix
ŞEKİLLLER DİZİNİ xi
ÇİZELGELER DİZİNİ xiii
SİMGELER VE KISALTMALAR DİZİNİ xiv
1. GİRİŞ VE AMAÇ 1
2. LİTERATÜR ARAŞTIRMASI 3
3. E-TİCARET SEKTÖRÜNE GENEL BAKIŞ 7
4. MÜŞTERİ KAYBI TAHMİNİ GENEL BAKIŞ 11
5. MATERYAL VE YÖNTEM 11
5.1. Yapay Zeka (Artificial Intelligence) 13
5.2. Yapay Öğrenme (Machine Learning) 14
5.2.1. Denetimli öğrenme (Supervised learning) 15
5.3. Sınıflandırma (Classification) 16
5.3.1. Lojistik Regresyon (Logistic Regression) 18
5.3.2. K- En Yakın Komşu (K-Nearest Neighbors) 19
5.3.3. Destek Vektör Makine (Support Vector Machine) 21
5.3.4. Karar Ağaçları (Decision Tree) 23
5.3.5. Rastgele Orman (Random Forest) 24
5.3.6. Artırma (Boosting) 26
5.3.6.1. Aşırı Gradyan Artırma (Extreme Gradient Boosting) 27
5.3.6.2. Hafif Gradyan Artırma (Light Gradient Boosting Model) 28
5.4. Model Performans Değerlendirme Ölçütleri 28
5.4.1. Karışıklık Matrisi (Confusion Matrix) 29
5.4.2. Doğruluk (Accuracy) 30
5.4.3. Kesinlik (Precision) 31
5.4.4. Duyarlılık (Recall) 31
5.4.5. F1-Skor (F-Score) 31

5.4.6. AUC (Area Under the ROC Curve) 32
5.5. Hiper Parametre Ayarlama (Hyperparameter Tuning) 33
5.5.1. GridSearchCV 34
5.6. SMOTE (Synthetic Minority Over-sampling Technique) 34
6. BULGULAR VE TARTIŞMA 35
6.1. Veri Setinin Tanımı 37
6.1.1. Öznitelik tanımı 37
6.2. Kodlama ve Kütüphaneler 39
6.3. Veri Ön İşleme (Data Preprocessing) 40
6.3.1. Eksik veri analizi (Missing value analysis) 40
6.3.2. Değişken dönüştürme 41
6.3.2.1. Veri setinde bulanan kategorik verileri tanımlama 41
6.3.2.2. Veri setinde bulunan kategorik verileri dönüştürme 46
6.3.3. Özellik ölçeklendirme (Feature scalling) 47
6.3.4. Veri setini ayırma (Split the data set) 48
6.4. Veri Setine SMOTE Yöntemi Uygulama 49
6.5. Modelleme 50
6.5.1. Lojistik Regresyon modelleme 50
6.5.2. Destek Vektör Makine (DVM) modelleme 53
6.5.3. K-En Yakın Komşu (K-NN) modelleme 56
6.5.4. Karar Ağaçları modelleme 58
6.5.5. Rastgele Orman modelleme 61
6.5.6. XGBoost modelleme 64
6.5.7. LightGBM modelleme 67
6.6. Modellerin Karşılaştırılması 70
6.7. Öznitelik Önem Düzeyi Belirleme 71
7. SONUÇ VE ÖNERİLER 74
KAYNAK DİZİNİ 76

ŞEKİLLLER DİZİNİ

Şekil **Sayfa**

3.1. Türkiye'de sektörlere göre e-ticaret istatistikleri……..8
4.1. Müşterilerin yaşam döngüsü……..12
5.1. Yapay zeka ve alt alanları……...14
5.2. Denetimli öğrenme algoritmalarının iş akışı……..16
5.3. Sınıflandırma şeması…….…….…….....18
5.4. K-NN algoritmasının çalışma mantığının gösterimi……..21
5.5. DVM algoritmasının çalışma mantığının gösterimi……...22
5.6. Karar ağacının yapısı…….24
5.7. Rastgele Orman algoritmasının çalışma mantığı……....25
5.8. Boosting algoritmasının temel düzeyde çalışma mantığı……...26
5.9. LightGBM algoritmasının ağaç oluşturma stratejisi…….….28
5.10. Karışıklık Matrisi…….30
5.11. AUC-ROC eğrisi…….....33
6.1. Çalışmanın iş akış şeması……..…….....36
6.2. Müşteri kaybının veri setinde dağılımı……...37
6.3. Her bir öznitelik sütunu için dolu veri sayısı……..……....40
6.4. "PreferredOrderCat" özniteliğine ait grafik……...42
6.5. "PreferredPaymentMode" özniteliğine ait grafik……...43
6.6. "PreferredLoginDevice" özniteliğine ait grafik…….....44
6.7. "MaritalStatus" özniteliğine ait grafik……...45
6.8. "Gender" özniteliğine ait grafik…….....46
6.9. Lojistik Regresyon modeli için karışıklık matrisi……...51
6.10. Lojistik Regresyon modeli için ROC-AUC eğrisi……....52
6.11. DVM modeli için karışıklık matrisi…….....54
6.12. DVM modeli için ROC-AUC eğrisi…….55

ŞEKİLLLER DİZİNİ (devam)

Şekil **Sayfa**

6.13. K-NN modeli için karışıklık matrisi…….57
6.14. K-NN modeli için ROC-AUC eğrisi…….....58
6.15. Karar Ağaçları model için karışıklık matrisi…….....60
6.16. Karar Ağaçları modeli için ROC-AUC eğrisi……...61
6.17. Rastgele Orman modeli için karışıklık matrisi…….63
6.18. Rastgele Orman modeli için ROC-AUC eğrisi…….64
6.19. XGBoost model için karışıklık matrisi…….66
6.20. XGBoost modeli için ROC-AUC eğrisi……...67
6.21. LightGBM model için karışıklık matrisi……..69
6.22. LightGBM modeli için ROC-AUC eğrisi…….....70

ÇİZELGELER DİZİNİ

Çizelge **Sayfa**

6.1. Veri setindeki öznitelik tanımı……...38
6.2. SMOTE yöntemi uygulama öncesi ve sonrası……....49
6.3. Modellerin doğruluk ve AUC değerleri……..71
6.4. LightGBM modeli için değişken önem düzeyleri…….…..72

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar	Açıklamalar
AUC	Area Under the ROC Curve
DN	Doğru Negatif
DP	Doğru Pozitif
DVM	Destek Vektör Makine
ETBIS	Elektronik Ticaret Bilgi Sistemi
K-NN	K- Nearest Neighbors
LightGBM	Light Gradient Boosting Model
ROC	Receiver Operating Characteristic Curve
SMOTE	Synthetic Minority Over-sampling Technique
XGBoost	Extreme Gradient Boosting
YN	Yanlış Negatif
YÖ	Yapay Öğrenme
YP	Yanlış Pozitif
YZ	Yapay Zeka

1. GİRİŞ VE AMAÇ

İnternet teknolojilerinin gelişmesi ve yaygınlaşması yaşamdaki birçok durumu etkilediği gibi kişilerin alışveriş kültürünü de etkilemiştir. Bu durum tüketicinin doğasında değişimlere neden olmuş, internet alışverişine ivme kazandırmış ve e-ticaret sektörüne katkıda bulunmuştur. Böylece e-ticaret platformları müşteriler için tatmin edici ürünler sunmaya başlamış ve müşteri sadakatini geliştirmeye çalışmışlardır. Özellikle tüm dünyayı etkisi altına alan pandemi sürecinde kişiler kendilerini korumak amacıyla dış dünya ile fiziksel etkileşimini minimize etmiştir. Bu süreçte müşteriler neredeyse tüm ihtiyaçlarını internet alışverişi aracılığıyla temin etmeye başlamışlardır. İnternet alışverişine olan talebin artması e-ticaret sektörünü oldukça canlandırmıştır. Böylelikle teknoloji pazarlamayı da yönlendirmeye başlamıştır. Küresel ekonomik ortamda iş yapmanın modern bir yolu olan e-ticaret, işletmelerdeki geleneksel iş süreçleriyle entegre olarak önemli bir bileşen haline gelmiştir. E-ticaret sektörü günümüzde piyasada baskın hale gelmeyi başarmıştır. Türkiye'de ETBİS (Elektronik Ticaret Bilgi Sistemi) verilerine göre 2021 yılında e-ticaret hacminin 2020 yılına göre %69 oranında artış gösterdiği görülmektedir (ETBİS, 2021). Bu da Türkiye'de e-ticaretin giderek yaygınlaştığını ve en hızlı büyüyen sektörler arasında yer aldığını göstermektedir.

E-ticaret sektörünün gelişmesi ve yaygınlaşmasıyla e-ticaret firma sayılarında artış yaşanmış ve rekabet ortamı artmıştır. Bu durum, müşterilerin alışveriş yaptıkları e-ticaret platformlarından memnun kalmadıkları durumlarda rakip platformlara yönelmelerine imkân sağlamıştır. İşletmeler açısından, müşteri ilişkilerinin bozulması ile müşteri sayısında ve satın alma oranlarında azalma muhtemel müşteri kaybı olarak tanımlanmaktadır. Müşteri kaybı, müşteri yıpranması veya müşteri kusuru olarak da adlandırılır. E-ticaret sektörü için müşteri kayıp oranları genel olarak yüksektir. Bu nedenle işletmelerin çevrimiçi alışverişlerdeki müşteri kaybını belirlemeye ve azaltmaya yönelik daha fazla çaba harcamaları ve geleceğe dönük gerekli tedbirleri önceden almaları gerekmektedir.

E-ticaret 'in öneminin gittikçe artması ve müşteri sayılarındaki önemli artışlar e-ticaret firmalarının gelirlerini de arttırmış ve daha fazla firmanın e-ticarete yönelmesine neden olmuştur. Rekabetin artması ve e-ticaret firma sayısının çoğalması ile sektördeki

firmalarda mevcut müşterilerini kaybetme korkusu da artmış ve olası müşteri kaybının önceden nasıl tahminleneceği çok önemli bir sorun haline gelmiştir. Ayrıca firmalar açısından iyi bilinen önemli bir gerçek de yeni müşteri elde etmenin maliyeti mevcut müşterileri elde tutmanın maliyetinden çok daha yüksektir. Bu nedenle müşteri davranışlarını analiz etmek, mevcut müşterilerin memnuniyetini artırmak, pazar rekabetini ve firma sürekliliğini sağlamak büyük önem taşımaktadır. Diğer yandan e-ticaret firmaları müşterilerinin gerçekten kaybolup kaybolmadığını doğru bir şekilde yargılayamamakta ve kayıp müşteri tahmininin karmaşıklığı sebebiyle zorluk yaşamaktadırlar. Firmalar açısından doğru şekilde müşteri kaybını tahminlemek için sofistike modellere ihtiyaç vardır.

Bu tez çalışmasında, e-ticaret sektöründeki müşteri kaybı tahmini için makine öğrenmesi algoritmaları kullanılarak tahmin modelleri kurmak ve belirlenen performans kriterlerine göre bunları kıyaslayarak hangi model ya da modellerin daha uygulanabilir olduğunu tespit etmek hedeflenmiştir. Ayrıca çalışmada, tespit edilen bu model ya da modeller için müşteri kaybına etki eden faktörlerin önem düzeylerinin analiz edilerek belirlenmesi de amaçlanmaktadır. Literatürde bu alanda nispeten az sayıda çalışma bulunmaktadır ve bu tez çalışmasının alana önemli katkılar sağlayacağı düşünülmektedir. Çalışmanın diğer bir önemi ise, geliştirilecek tahmin modeli ya da modelleri ile e-ticaret sektöründeki firmalara, müşteri kaybını en aza indirmeleri ve yeni müşteriler kazanımları için pazarlama stratejileri geliştirmelerine olanak sağlamak olacaktır.

Bu çalışma yedi bölümden oluşmaktadır. İlk bölümde çalışmanın kapsamı, önemi ve amacından bahsedilmiştir. İkinci bölümde e-ticaret sektöründeki müşteri kaybı tahmini için geçmişten günümüze kadar yapılmış olan çalışmalardan bahsedilmiştir. Üçüncü bölümde e-ticaret sektörüne ait genel bilgiler ele alınmıştır. Dördüncü bölümde müşteri kaybı tahmininden genel olarak bahsedilmiş ve bilgi verilmiştir. Beşinci bölümde bu tezde kullanılan yöntemler anlatılmıştır. Altıncı bölümde tez çalışmasının uygulama kısmı anlatılmıştır ve çalışma bulguları verilip tartışılmıştır. Yedinci ve son bölümde çalışma sonuçları değerlendirilip sonuç ve öneriler üzerinde durulmuştur.

2. LİTERATÜR ARAŞTIRMASI

Yanfang ve Chen (2017) yapmış oldukları çalışmada, Lojistik Regresyon modeline dayanan e-ticaret müşteri kaybı üzerine araştırma yapmak amacıyla e-ticaret kullanıcı davranışı analiz edilmiş ve e-ticaret kullanıcı kaybı tahmin modeli oluşturulmuştur. Faktör analizi yöntemi kullanılarak kullanıcı davranış faktörleri analiz edilmiş ve kullanıcı kaybını etkileyen faktörler belirlenmiştir. Tahmin modeli olarak Lojistik Regresyon modeline dayanan Elektronik İş Kullanıcısı Tutma Modeli (EBURM) önerilmiştir. Bu model, örnek veriler ile eğitilip maksimum olabilirlik yöntemi kullanılarak modelin her bir parametresinin tahmin değeri elde edilmiş ve böylelikle nihai e-ticaret kullanıcısını elde tutma durumu elde edilmiştir. Oluşturulan model, kullanıcıyı elde tutma oranının farklı etki faktörlerine dayalı olarak kişiselleştirilmiş bir operasyonel öneri stratejisi sağlamıştır. EBURM modeli, AUC test yöntemi ile değerlendirilmiştir. Sonuçlara göre EBURM modeli ile, ayrılan ve kullanmaya devam eden kullanıcılar için gerçek beklentilerle tutarlı olduğu gösterilmiş ve önerilen EBURM modelinin kullanıcı kayıp davranışını yüksek bir güven düzeyinde tahmin edebileceği kanıtlanmıştır.

Raeisi ve Sajedi'nin (2020) yapmış olduğu karşılaştırmalı çalışmanın amacı, çevrimiçi özellikler ve kullanıcı davranışlarını kullanarak müşteri kaybı tahmini yapmaktır. Tahmin modellemesi için Gradient Boosted Trees (GBT) kullanılmıştır. Bu modelin temel model olarak kullanılma sebepleri sınıflandırma için uygun bir algoritma olmasıdır. GBT kullanımına ek olarak, toplanan veri setinin sınıflandırılması için farklı türlerden başka sınıflandırma yöntemleri de dikkate alınmıştır. Kullanılan diğer sınıflandırma algoritmaları K-En Yakın Komşu, Naive Bayes, Karar Ağacı, Rastgele Orman ve Kural Tümevarım yöntemleri kullanılmıştır. Kullanıcıların çevrimiçi davranışlarına dayalı olarak ayrılma tahmin edilmiştir. Son sipariş tarihi altı ay öncesi olan kişiler kayıp müşteri kabul edilmiştir. Algoritmalar kullanılmış ve doğruluk oranına göre en iyi algoritma, makalenin temeli olan GBT algoritması olarak sonuç vermiştir. Son olarak, B2C e-ticaret platformu üzerindeki ampirik çalışma ile, bu modelin olgun müşteri kaybı tahmin algoritmalarına kıyasla daha iyi verimliliğe ve doğruluğa sahip olduğu kanıtlanmıştır.

Wu ve Meng'in (2016) yayınlamış olduğu makalede, kayıp müşterilerin tahmin doğruluğunu iyileştirmek ve aynı zamanda kayıp olmayan müşterileri belirlemeyi güçlendirmek için geliştirilmiş SMOTE ve AdaBoost algoritmasına dayalı e-ticaret müşteri kaybı tahmin modeli öne sürülmüştür. Veri seti dengesizliği göz önünde bulundurularak geliştirilmiş SMOTE ve AdaBoost algoritmasına dayalı bir e-ticaret müşteri kaybı tahmin modeli önermektedir. İlk olarak, dengesizlik sorununu çözmek için aşırı örnekleme ve yetersiz örnekleme yöntemlerini birleştiren ve ardından tahmin için AdaBoost algoritmasına entegre edilen gelişmiş SMOTE ile kayıp verilerinin işlenmesi yapılmıştır. İkinci olarak, dengeli veri kümesini eğitmek üzere AdaBoost modeli kullanılmıştır. Sonuç olarak önerilen modelin diğer tahmin algoritmalarına kıyasla daha iyi verimliliğe ve doğruluğa sahip olduğu kanıtlanmıştır.

Li ve Li'nin (2019) bir e-ticaret platformunun gerçek verilerine dayanan çalışmasında, Lojistik Regresyon ve Aşırı Gradyan Artırma (XGBoost) algoritmasına dayalı müşteri kaybı için hibrit bir tahmin modeli oluşturulmuş ve bu durumun tahmini daha kapsamlı ve doğru hale getirdiği öne sürülmüştür. Başka bir ifadeyle iki teknik, e-ticaret platformları için müşteri kaybını hibrit bir tahmin modelinde birleştirmiş ve Lojistik Regresyon tekniğinin tahmin doğruluğunu artırmak için XGBoost tanıtılmıştır. Doğruluk, kesinlik ve geri çağırma ile değerlendirildiğinde, hibrit modelin müşteri kaybını Lojistik Regresyon tekniğinden daha doğru tahmin edebildiği görülmüş ve öne sürülen iddia kanıtlanmıştır.

Guo ve Qin (2015), e-ticaret müşterilerinin temel bilgilerini analiz etmek ve müşteri değişimlerinin özelliklerini bulmak için veri madenciliği tekniklerinden Karar Ağacı algoritmasını kullanmıştır. Müşteri kaybının tahmini, Karar Ağacı ile çözülebilecek bir sınıflandırma problemi olarak düşünülmüş ve bu yüzden bu algoritma tercih edilmiştir. Karar Ağacı algoritmasına dayalı RFM-DR modeli oluşturulmuştur. RFM-DR modeline göre, karar ağacının öznitelik seti {R, F, M, DR} olarak tanımlanmış, burada R son işlem zamanı, F işlem sıklığı, M işlem tutarı ve DR iskonto gelirini temsil etmektedir. Dört nitelik arasından indirim gelirinin (DR) en yüksek bilgi kazancına sahip olduğunu tespit edilmiştir. Kendi eğitim setleri sonucu elde ettikleri eşik değerleri değerlendirme kuralı olarak kullanılmış ve modelin tahmin doğruluğu hesaplanmıştır.

Zhuang'ın (2018) e-ticaret firmasının müşteriyi doğru bir şekilde alt bölümlere ayırmasına yardımcı olmak için yapmış olduğu çalışmanın amacı, sosyal ağın değerini bütünleştiren bir müşteri değeri modeli oluşturmaktadır. Müşterileri segmentlere ayırmak için RFM'ye dayalı RFMI değer modelini önermektedir. R (Yenilik) son tüketimi, F (Frekans) tüketim sıklığını, M (Parasal) tüketim miktarını, I her müşterinin ağ etkisi değerini ifade etmektedir. Ardından, müşteri gruplarının segment öncesi ve sonrasındaki değişimini tahmin etmek için makine öğrenimi algoritması olan XGBoost algoritması kullanılmıştır. Bu çalışma ayrıca, XGBoost algoritması ile karşılaştırmak için müşteri kayıp tahmini araştırmalarında yaygın olarak kullanılan Lojistik Regresyon, Destek Vektör Makinesi ve Sinir Ağı algoritmalarını seçmiştir. Modeli değerlendirmek için doğruluk oranı, kaldırma katsayısı ve AUC değeri kullanılmıştır. Deneysel sonuçlar, müşterileri segmentlere ayırdıktan sonra tahmin sonuçlarının daha doğru olduğunu kanıtlamıştır. Aynı zamanda, XGBoost algoritması diğer tahmin algoritmalarından daha iyi performans göstermiştir.

Çakırdoğan'ın (2021) tez çalışmasında, e-ticaret perakende satış sitesi için müşteri kaybı tahmin modeli geliştirilmiştir. Model de RFM analizine dayanan değişken seçimi yapılmıştır. Veri kümesinde bağımlı değişken olmadığı için kalibrasyon ve gözlem olmak üzere iki kısma ayrılmıştır. Böylelikle müşterileri iki dönemde aktiflik durumuna göre kayıp veya kayıp olmayan olarak sınıflandırarak bağımlı değişken belirlenmiştir. Tahminleme için Boosting, Lojistik Regresyon ve Destek Vektör Makine algoritmaları kullanılmış ve algoritmaların performansı kıyaslanmıştır. Performans kriteri olarak doğruluk ve AUC değeri kullanılmıştır. En başarılı sonucu XGBClassifier algoritması vermiştir.

Xiahou ve Harada'nın (2022) çalışmasında, B2C e-ticaret müşterilerinin alışveriş davranışlarının boylamsal zaman çizelgelerinin ve çok boyutlu veri değişkenlerinin özelliklerine göre, K-ortalama müşteri segmentasyonu ve Destek Vektör Makinesi (DVM) tahmininin kombinasyonuna dayalı bir kayıp tahmin modeli önermektedir. Bu araştırmada iki önemli adım vardır: müşteri segmentasyonu ve müşteri kaybı tahminidir. Müşteri segmentasyonu için K-ortalamalar algoritması ve değişken özellikleri seçmek için Rastgele Orman metodu kullanılmıştır. Veri segmentlere ayrıldıktan sonra tahmin modelleri oluşturmak ve karşılaştırmak için Lojistik Regresyon algoritması ve DVM algoritması kullanılmıştır. Ayrıca veri setinde dengesizlik olduğu için SMOTE ile dengesizlik giderilmiştir. Veri setindeki üç kategorinin performansını değerlendirmek için her kategori

doğruluk, duyarlılık ve kesinlik değerleri karışıklık matrisine göre hesaplanmıştır. Sonuç olarak, DVM modelinin tahmin doğruluğunun, Lojistik Regresyon modelinin tahmin doğruluğundan daha yüksek olduğu kanıtlanmış ve müşterileri segmentlere ayrıldıktan sonra tahmin modelinin doğruluğunu olumlu yönde etkilediği gösterilmiştir.

Yu vd. (2011), Çin'de bulunan popüler bir e-ticaret sitesinin veri setini kullanarak yapmış olduğu çalışmalarında Genişletilmiş Destek Vektör Makinesinin (ESVM) etkinliği ve doğruluğunu göstermek için müşteri kaybı uygulaması yapmıştır. Müşteri kaybının dengesizliği ve doğrusal olmayan durumu için kayıplı, kayıpsız ve doğrusal olmayanın etkisini anlatan parametreler tanıtılarak genişletilmiş bir destek vektör makinesi (ESVM) önerilmiştir. Böylelikle yeni bir müşteri kaybı tahmini modeli oluşturulmuştur. Çalışmanın amacı e-ticaret sitesinde ki kayıp müşteriyi tahmin etmektir ve bu yüzden veri setinin amacı müşteri davranışlarıdır. ESVM ile birlikte alternatif olan Yapay Sinir Ağları, Karar Ağaçları, Destek Vektör Makine algoritmaları da kullanılmıştır. Algoritmalar ayrı bir şekilde eğitilmiş ve sonuçlar elde edilmiştir. Performans kriteri olarak doğruluk, isabet oranı, kapsama oranı, kaldırma katsayısı değerleri seçilmiştir. Algoritmalar bu değerlere göre kıyaslanmış ve en uygun model seçilmiştir. Diğer modellere göre en iyi sonucu veren model ESVM çıkmış ve diğer modellerden daha avantajlı olduğunu ortaya koymuştur. ESVM modelinin diğer modellerden daha avantajlı olma nedeni verinin dengesiz ve doğrusal olmayan olma durumunu dikkate almış olmasıdır. Müşteri kaybını tam olarak tahmin ettiği için bu model veri seti için uygun görülmüştür. Bu sonuç sayesinde e-ticaret firması bazı reklam, indirim, e-posta bildirimi, teslimat hızının iyileştirilmesi ve çevrimiçi paket takibi gibi bazı eylemler gerçekleştirmiştir. Bu sayede müşteri kaybı %3 azalmıştır.

Bagul vd, (2021), müşterilerin segmentasyonu için RFM tekniği ve K-ortalama algoritması kullanmıştır. Veri seti kümeleme amaçlı kullanılmıştır. Her müşteri için Yenilik, Sıklık, Parasal puan almak için RFM modeli uygulanmış ve ardından müşteriyi ait olduğu kümeye atamak için K-ortalama algoritması uygulanmıştır ve kümeler oluşturulmuştur.

3. E-TİCARET SEKTÖRÜNE GENEL BAKIŞ

Son yıllarda internetin yaygınlaşması ve ticari hale gelmesi ile elektronik ticaret ön plana çıkmıştır ve elektronik ticaret kavramı yeni bir olgu olmayıp uzun yıllardır ticaretin önemli bir parçası olmuştur (Murthy, 2007). İnternet ticareti olarak da bilinen e-ticaret, telekomünikasyon ağları aracılığıyla ticari ilişkilerin sürdürülmesi, ticari bilgilerin paylaşılması ve ticari işlemlerin yürütülmesidir (Vladimir, 1996). E-ticaret internet aracılığıyla kolaylaştırılan her türlü ticari işlem olarak da söylenebilir. E-ticaret, ürünlerin ve hizmetin çevrimiçi perakendeciler ve pazar yerleri aracılığıyla keşfedilmesini ve satın alınmasını kolaylaştırmak için geliştirilmiştir. E-ticaret internetten beslenmektedir. Tüketiciler, kendi cihazları aracılığıyla ürün veya hizmetlere göz atmak ve bu hizmetlerden yararlanmak için bir çevrimiçi mağazaya erişmektedir.

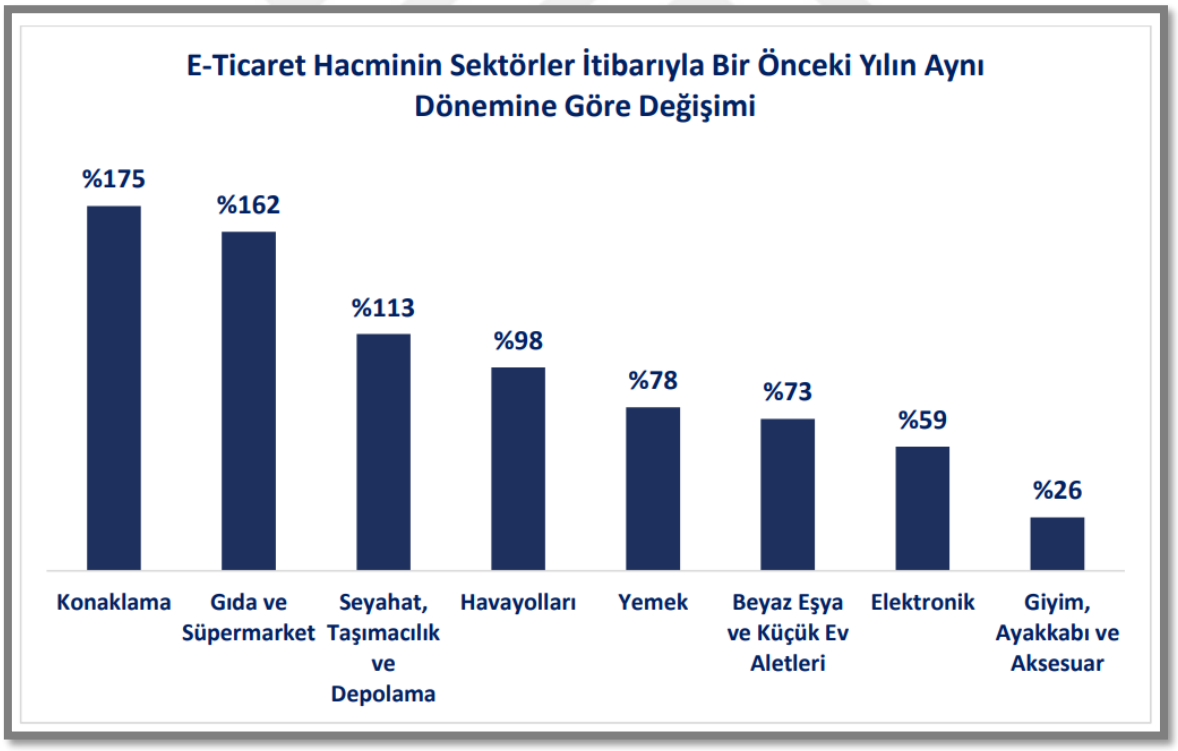
Ticari işlemin niteliğine göre literatürde farklı e-ticaret türleri vardır (Gedik, 2021):

- B2C, bir işletmenin bireysel tüketiciye mal veya hizmeti çevrimiçi ortamda satmasıdır.
- C2B, bir tüketicinin kendi mal veya hizmetini bir işletmeye çevrimiçi ortamda satmasıdır.
- B2B, bir işletmenin başka bir işletmeye mal veya hizmetini çevrimiçi ortamda satmasıdır.
- C2C, bir tüketicinin başka bir tüketiciye kendi mal veya hizmetini çevrimiçi ortamda satmasıdır.
- B2G, bir işletmeden devlete gerçekleşen e-ticaret türüdür. Devletin ihale açması ve işletmelerin bu ihaleye çevrimiçi ortamda takip edip, mal ve hizmet satışının yapılması bu duruma örnektir.
- G2B, devletten işletmeye gerçekleşen bir e-ticaret türüdür. Devlet tarafından verilen destekler bu duruma örnek olarak verilebilmektedir.
- G2G, devletler arası ürün ve hizmetin sağlanması durumudur.
- C2G, tüketiciden devlete olan bir e-ticaret türüdür. Vergi, trafik cezaları bu duruma örnektir.

- G2C, devletten tüketiciye doğru bir hizmet akışı durumudur. Kimlik belgesi ve pasaport alımı, e-devlet sistemi bu duruma örnek olarak gösterilebilmektedir.

Yukarıda bahsedilen ticari ilişkideki taraflar için B (Business) işletmeyi, C (Consumer) tüketiciyi, G (Government) devleti ifade etmektedir.

Son yıllarda elektronik ticaretin hızla gelişmesiyle, e-ticaret işletmelerinin sayısı her geçen gün daha fazla ve büyük ölçekte artmakta ve hizmet giderek homojen olmakta, e-ticaret işletmeleri arasında rekabet daha yoğun hale gelmektedir (Bagul vd., 2021). Türkiye için 2021 yılı e-ticaret verilerine yönelik istatistikler incelendiğinde, 2021 yılında e-ticaret hacminin 2020 yılına göre %69 oranında artış göstermiş olduğu görülmektedir (ETBİS, 2021). Sektörlere göre e-ticaret istatistikleri Şekil 3.1'de verilmiştir.



Şekil 3.1. Türkiye'de sektörlere göre e-ticaret istatistikleri (ETBİS, 2021)

Şekil 3.1'de görüldüğü gibi Türkiye'de 2020 yılına göre 2021 yılında en fazla artışın olduğu sektör konaklama, gıda ve süpermarket, seyahat, taşımacılık ve depolama sektörüdür.

E-ticaretin uluslararası ticarette daha geniş uygulaması ile, e-ticaretin avantajları giderek daha belirgin hale gelmektedir (Ju, 2022). Bu avantajlar:

- Müşteri 7/24 e-ticaret platformlarına ulaşabilmekte ve istedikleri zaman hizmetlerden yararlanma imkanına sahip olmaktadır.
- Müşterinin çevrimiçi alışverişi fiziksel olarak yaptığı alışverişten daha hızlı olmaktadır. Çevrimiçi alışverişin erişim hızı daha yüksektir.
- E-ticaret platformu fiziksel mağazalara göre daha geniş aralıkta ürün ve hizmet sağlamaktadır. Müşteri ulaşmak istediği çoğu ürün ve hizmete kolay bir şekilde ulaşmaktadır.
- E-ticaret işletmeleri, kira, envanter ve kasiyer gibi fiziksel mağaza işletme maliyetleri olmadığı için ürünlerini ona göre fiyatlandırmaktadır. Böylelikle müşteri çevrimiçi platformda satın almak istediği ürün ve hizmeti daha uygun fiyata satın alma imkanına sahip olmaktadır.
- Müşterilere satın almak istedikleri ürün ve hizmetler için uluslararası bir erişime sahiptir. Fiziksel olarak gidip alamadıkları ürün ve hizmeti çevrimiçi ortam aracılığıyla istediği herhangi bir ülkeden kolaylıkla erişim sağlayabilmektedir.

E-ticaret sektörünün avantajları olduğu kadar dezavantajları da vardır. Bu dezavantajlar:

- Bekleme süresi, müşteri fiziksel mağazada ürünü anında satın alır fakat çevrimiçi alışverişte ise müşteri ürünün eline ulaşması için belli bir süre beklemesi gerekmektedir.
- Müşteri satın alacağı ürüne ve hizmete sınırlı bir erişime sahiptir. Satın almak istenilen ürüne e-ticaret platformda bulunan fotoğrafa bakılarak ancak satın alınabilmektedir. Ürünle doğrudan temas edememektedir.
- E-ticaret platformları ile müşteri birçok kişisel verilerini paylaşmaktadır. Müşteriler, herhangi bir güvenlik sızıntısı durumunda kişisel bilgilerinin ele geçirilmesi riskine sahiptirler.

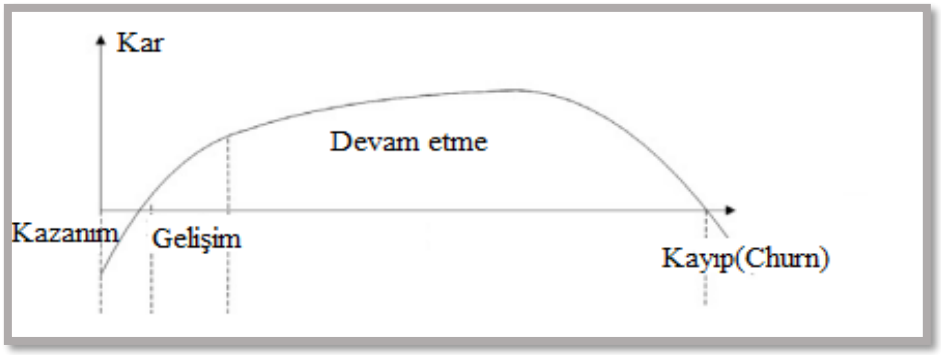
E-ticaret sektörünün en önemli yanlarından birisi ise veri depolama imkanına sahiptir. Bir e-ticaret ortamında şirketler, müşterilere hizmet vermek için internet platformunu kullanır, müşteriler ağ platformuna göz atar, satın alma süreci büyük miktarda veri trafiği üretir (Bagul vd., 2021). Böylelikle müşterilerin deneyimi veri deposu niteliği taşımaktadır. Çalışma yapmak, analiz etmek isteyen kişiler veya kurumlar için büyük kolaylık sağlamaktadır.

4. MÜŞTERİ KAYBI TAHMİNİ GENEL BAKIŞ

Müşteri kaybı, müşteriler bir kuruluştan ürün veya hizmet satın almaya devam etmemeye karar verdiğinde ve ilişkilerini sonlandırdığında meydana gelir. Müşterilerin belirli bir zaman diliminde bir şirketle iş yapmayı veya hizmet almayı bırakma eğilimi olarak tanımlanan müşteri kaybı, önemli bir sorun haline gelmiş ve dünya çapında birçok firmanın yüzleşmek zorunda olduğu başlıca zorluklardan birisi olmuştur (Xie vd., 2009). Müşteri kaybı, müşteri yıpranması olarak da bilinmektedir (Yu vd., 2011). Bu durum ile bağlantılı müşteri kaybı tahmini ise hangi müşterilerin, bir hizmetten ayrılma veya hizmet aboneliğini iptal etme olasılığının olup olmadığının tespit edilme sürecidir (Raeisi ve Sajedi, 2020). Birçok işletme için kritik bir tahmindir çünkü yeni müşteriler edinmek çoğu zaman mevcut müşterileri elde tutmaktan daha maliyetlidir (Guo ve Qin ,2015). Müşteri kaybı, büyüyen bir işletmeyi değerlendirmek için önemli bir kriterdir, bu nedenle şirketlerin müşterilerini elde tutmak için müşteri kaybını öngörmeleri önemlidir. Farklı müşteriler farklı davranış ve tercihler sergilemekte, çeşitli nedenlerle ayrılma sebepleri olmaktadır. Bu yüzden müşteri kaybı aktif ve pasif olmak üzere iki kategoriye ayrılmaktadır (Gou ve Qin, 2015):

- Aktif müşteri kaybı, gönüllü bir şekilde müşterinin ayrılmasıdır. Müşterinin iş değiştirmesi, hizmet kalitesi, iş rekabeti, profesyonel kaybı vb. gibi kendi sebeplerinden dolayı çevrimiçi alışveriş yapmaması anlamına gelir.
- Pasif müşteri kaybı, istem dışı bir şekilde müşterinin ayrılmasıdır. Firmaların sorumlu olduğu müşteri kaybı türüdür. Firmalar, kredi sorunları nedeniyle müşterilerin üyeliklerini iptal ederler.

Eldeki müşteriyi tutmak işletmeler için çok değerlidir. Şekil 4.1'de müşterinin bir işletme için yaşam döngüsü verilmiştir.



Şekil 4.1. Müşteri yaşam döngüsü (Yu vd., 2011)

Şekil 4.1'de görüldüğü gibi ilk aşamada müşteri kazanılır. İlk aşamada müşteriyi kazanmak için maliyet harcaması sebebiyle kar bu periyotta negatif kısımdadır. İkinci aşamada müşterinin kara etkisi pozitif yönde ilerlemektedir. Üçüncü aşama müşterinin kalmaya devam etmesidir. Firmalar uzun vadeli müşteri ile ilişki kurmayı başarmıştır. Müşteri karı artırmaya devam ediyordur ve pozitif bölgededir. Dördüncü aşama ise müşterinin servis ve hizmet almayı bırakmak istemesi yani müşterinin kayıp olduğu aşamadır. Müşterilerin kara etkisi negatiftir ve işletmelere kar olarak bu durum zarar vermiştir.

Müşteri kaybı problemi, yaygın olarak bilinen ikili sınıflandırma problemi olarak görülmektedir (Zhao vd., 2005). Bir şirket büyümek istiyorsa mevcut müşteriyi elde tutması gerekir. Bir müşteri her ayrıldığında kaybedilen önemli bir yatırımı temsil etmektedir. Bu yüzden, bir müşterinin ne zaman ayrılacağını tahmin etmek ve onlara kalmaları için teşvikler sunabilmek, bir işletmeye büyük oranda tasarruf sağlamaktadır.

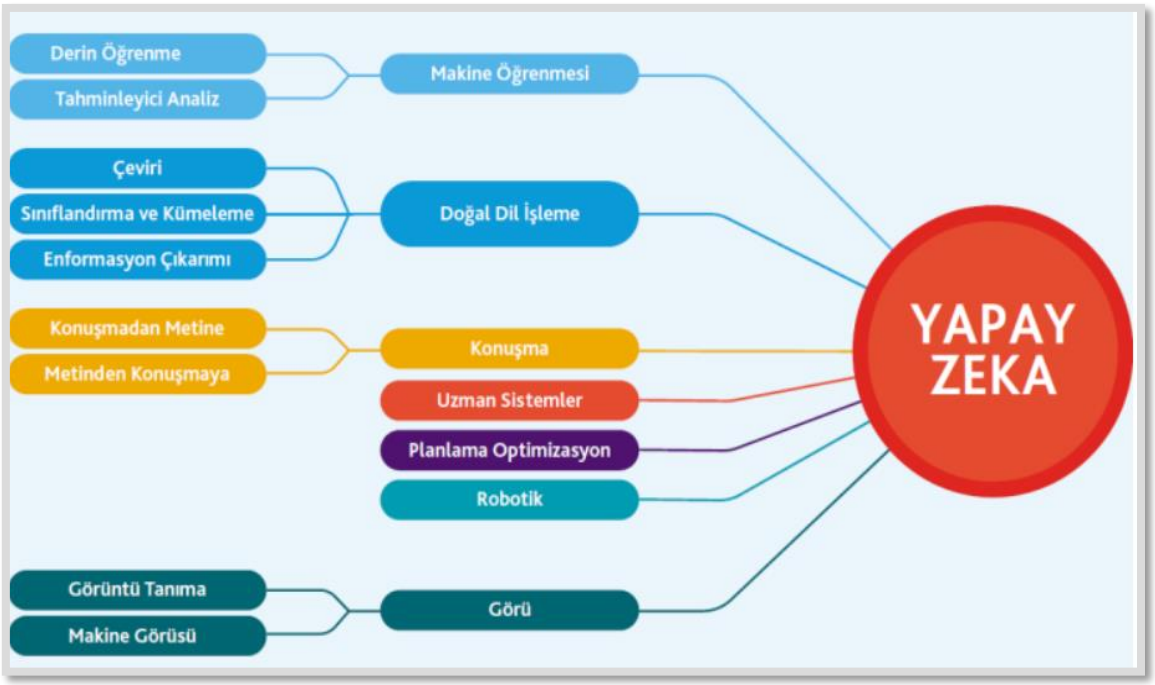
5. MATERYAL VE YÖNTEM

Bu bölümde tez çalışması için kullanılan yapay öğrenme metotları ayrıntılı bir şekilde anlatılmıştır.

5.1. Yapay Zeka (Artificial Intelligence)

Yapay zeka (YZ), 1956 yılında yapılan Darmouth Konferansı'ndan bu yana araştırma alanı olarak çalışılmaktadır (McCarthy, J. ,2004). İlk YZ kavramı yaygınlaşmaya başladığı zamanlar akıllı bilgi sistemlerine yani kavramların özerk, uyarlanabilir bir şekilde metinlere ve belgelere bağlı olarak bilgiye etkin erişim sağlayabilen sistemlere sahip olunabileceği öne sürülmeye başlanmıştır. Yapay zekanın bir araştırma disiplini olarak kurulması, öğrenmenin her yönünün veya zekanın herhangi bir özelliğinin onu simüle ederek bir makine yapılabileceğini söylemiştir. Amaç, iş başındaki süreçleri kendi zekamızda otomatikleştirilebilecek şekilde ortaya çıkararak makinelerde akıllı insan davranışını yeniden üretmektir. Ancak günümüzde çoğu araştırmacı, insan benzeri araçlar yerine herhangi bir yolla karmaşık problem alanlarında iyi performans gösteren otomatik sistemler tasarlamak istemektedir (Dick, 2019).

YZ, minimum insan müdahalesi ile akıllı davranışı modellemek için bir bilgisayarın kullanımını ima eden genel bir terim ve akıllı makineler, akıllı bilgisayar programları yapma bilimi ve mühendisliğidir (Hamet ve Tremblay, 2017). İnsan zekasını anlamak için bilgisayar kullanarak benzer görevi ilişkilendirir ancak yapay zekanın biyolojik olarak gözlemlenebilen yöntemlerle kendisini sınırlaması gerekmez. Çalışma şekli olarak büyük miktarda veriyi hızlı, yinelemeli şekilde işler ve akıllı algoritmalarla sentezler. Ayrıca verilerde ki desenlerden, özelliklerden otomatik olarak öğrenmesini sağlar. Kısaca YZ'nin amacı girdi üzerinde mantıklı ve çıktıda açıklanabilecek yazılımlar sağlamaktır. YZ çalışma alanı olarak birçok alt alana ayrılmaktadır (Abioye vd., 2021). Şekil 5.1'de alt alanlar gösterilmiştir.



Şekil 5.1. Yapay zeka ve alt alanları (Sanket, 2017)

5.2. Yapay Öğrenme (Machine Learning)

Yapay zekanın çalışma alanlarından birisidir ve makinelerin açıkça programlanmak yerine verilerden öğrenmesine olanak veren ve tüm yaklaşımları içeren bir alt alandır (Shinde ve Shah, 2018). Yapay öğrenmede (YÖ), bazı görevleri gerçekleştirmek için bir bilgisayar programı atanır ve makinenin, bu görevleri yerine getirirken daha fazla deneyim kazandıkça bu görevlerdeki ölçülebilir performansının gelişip gelişmediğini kendi deneyimlerinden öğrendiği söylenmektedir (Ray, 2019). Ayrıca örneklenmiş verilere dayalı matematiksel modeller oluşturan algoritmalar da sağlar. Bu matematiksel modeller (fonksiyonlar olarak adlandırılır) girdi verilerini istenen çıktılara eşler. Girdiler, görüntüler ve rastgele bir sayısal veya kategorik veri dizisi olabilir. YÖ'nün amacı veri ve algoritmalara dayalı olarak makineleri eğitmektir. İşlenmiş verileri ve bilgileri kullanarak makinelerin nasıl karar vereceğini öğrenir. Dinamik bir yapıya sahiptir ve daha fazla veri ile karşılaştığında kendini değiştirme ve geliştirme yeteneğine sahiptir (Janiesch vd., 2021). YÖ için öğrenmek demek algoritmaların hataları en aza indirgemeye ve tahminlerinin doğru olma olasılığını en üst seviyeye çıkarması demektir. Kısaca yapay öğrenme, YZ gerçekleştirmek için kullanılan tekniklerden bir tanesidir.

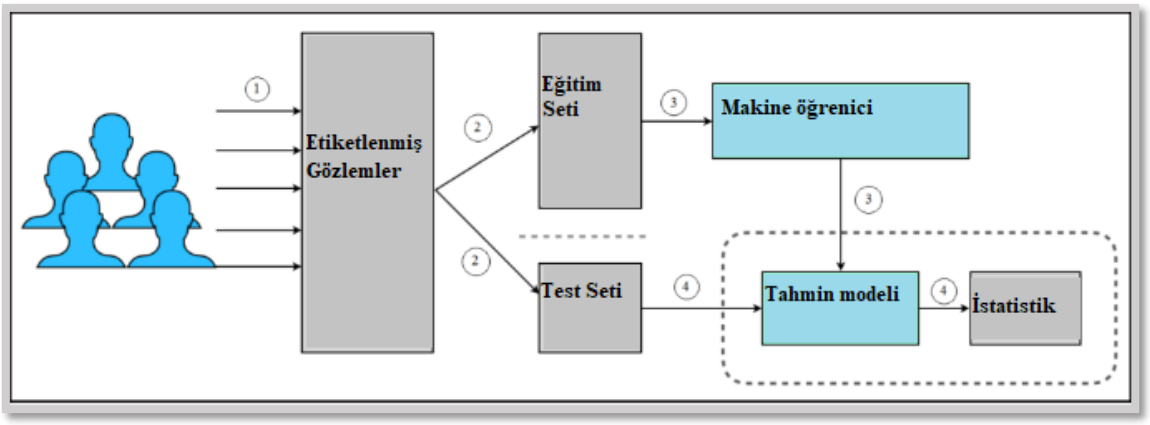
YÖ, veri bilimi alanının büyümesinde çok önemli etkenlerden birisidir. İstatistiksel yöntemler, algoritmalar kullanarak sınıflandırma, tahminleme yapacak ve veri madenciliğinde temel bilgileri açığa çıkaracak şekilde eğitilir ve test edilir. Daha sonra karar vericileri yönlendirmede rol oynar.

Yapay Öğrenme, eğitim sırasında ihtiyaç duydukları duruma göre dört temel kategoriye ayrılır (Kang ve Jameson, 2018):

- Denetimli Öğrenme (Supervised Learning)
- Denetimsiz Öğrenme (Unsupervised Learning)
- Yarı Denetimli Öğrenme (Semi-supervised Learning)
- Pekiştirmeli Öğrenme (Reinforcement Learning)

5.2.1. Denetimli öğrenme (Supervised learning)

Denetimli öğrenme örnek girdi-çıktı çiftlerine dayanarak bir girdiyi bir çıktıya eşleyen fonksiyonu öğrenmesidir (Cunningham, 2008). Çalışılan veri setinde hangi girdi değerinin hangi çıktı değerini vereceği işaretlenmiştir. Özetle istenen çıktı değerinin bilindiği girdi gibi etiketli örnekler kullanılarak eğitilmesine denetimli öğrenmedir. Eğitim örneğinden oluşan etiketli eğitim verilerinden bir işlev çıkarır. Veri seti eğitim ve test seti olarak iki kısma ayrılır. Eğitim veri seti, tahmin edilmesi veya sınıflandırılması gereken çıktı değişkenine sahiptir. Tüm makine öğrenmesi algoritmaları eğitim veri kümesinden bir tür kalıp öğrenir ve bunları test veri setine uygular. Bu şekilde verileri doğru tahminleme ve sınıflandırma yapar. Denetimli öğrenmenin tüm algoritmalarının iş akışı Şekil 5.2'de verilmiştir.



Şekil 5.2. Denetimli öğrenme algoritmalarının iş akışı (Chinnamgari, 2019)

Denetimli öğrenme problem türüne göre iki temel alt kategoriye ayrılır (Zhu vd., 2008):

- Sınıflandırma (Classification)
- Regresyon (Regression)

Denetimli öğrenmenin denetimsiz öğrenmeden, yarı denetimli öğrenmeden ve pekiştirmeli öğrenmeden temel farkı etiketli veri setini kullanıp doğru şekilde tahminleme ve sınıflandırma yapmasıdır. Denetimsiz öğrenmede etiketsiz veriler kullanılır (Biamonte vd., 2017). Makine öğrenmesi algoritmaları verileri düzenler veya bu verilerin yapısını açıklayıp anlamlı ilişkiler kurarak verileri etiketli hale getirir. Yarı denetimli öğrenmede hem etiketli hem de etiketsiz verileri birlikte kullanır. Yani büyük miktarda etiketsiz veri ile az miktarda etiketli veriyi kullanır. Pekiştirmeli öğrenmede ise sonuçlardan öğrenir.

5.3. Sınıflandırma (Classification)

Denetimli öğrenmenin tekniklerinden birisidir (Kotsiantis vd., 2007). Sınıflandırma algoritması, eğitim verilerini temel alarak yeni gözlemlerin kategorisini belirler. Sınıflandırma tekniği denetimli bir öğrenme tekniği olduğu için etiketlenmiş girdi değeri ve ona karşılık gelen çıktı değerini kullanarak çalışır. Kısaca sınıflandırma algoritmasının amacı, belirli bir veri kümesinin kategorisini belirlemektir ve bu algoritmalar temel olarak kategorik verilerin çıktısını tahmin etmek için kullanılır. Sınıflandırma verisinin çıktı değeri kategoriktir. Örneğin, evet veya hayır, kadın veya erkek, istenmeyen e-posta (spam) veya

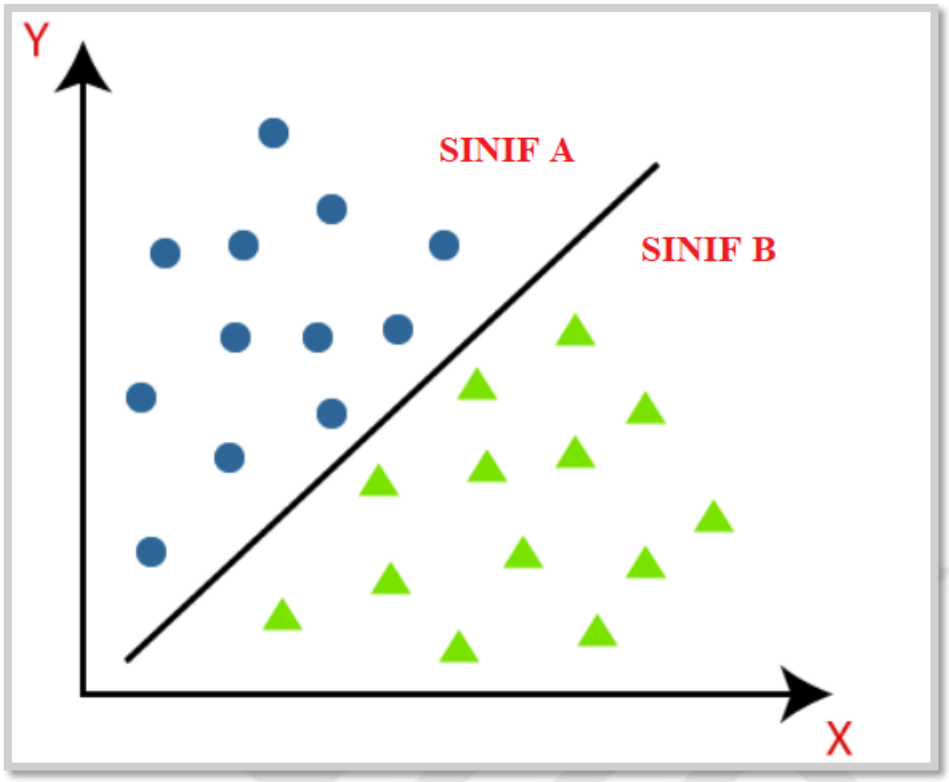
spam değil, kayıp (churn) veya kayıp değil şeklindedir. Model kurabilmek için daha sonra kategorik değerler sayısal değerlere dönüştürülür. Sınıflandırma modeli, eğitim veri setini kullanarak girdi veri örneklerinin belirli sınıf etiketlerine en iyi nasıl eşleyeceğini hesaplayacağı için eğitim veri seti, çalışılan problemin tüm senaryolarını içermeli ve her etiket yeterli veriye sahip olmalıdır.

Sınıflandırmanın regresyondan farkı, çıktı değişkeni kategorik olan veri seti üzerinde çalışmasıdır. Regresyonda tahmin edilmek istenen çıktı değişkenin değeri ise sürekli (continious) yani sayısal bir değerdir.

Problem türüne göre sınıflandırıcı türü de değişmektedir. Sınıflandırma probleminin sadece iki olası sonucu varsa buna ikili sınıflandırma (binary classification) denir (Lorena, 2008). Örneğin, evet veya hayır, kadın veya erkek şeklinde olmasıdır. Eğer bir problemin ikiden fazla sonucu varsa çok sınıflı sınıflandırıcı (multi-class classification) denir. Bitki türlerinin sınıflandırılması, müzik türlerinin sınıflandırması çok sınıflı sınıflandırıcılara örnektir.

Sınıflandırma için kullanılan birçok algoritma mevcuttur (Kang ve Jameson, 2018). Bunlardan Lojistik Regresyon, K-En Yakın Komşu, Destek Vektör Makine, Karar Ağaçları, Naive Bayes, Yapay Sinir Ağları (ANN) en çok kullanılan algoritmalardır.

Sınıflandırma algoritmalarının temel çalışma yapısı Şekil 5.3’te gösterilmiştir.



Şekil 5.3. Sınıflandırma şeması

Sınıflandırma şemasında A sınıfı ve B sınıfı olmak üzere iki sınıf vardır. A sınıfı kendi içinde ve B sınıfı kendi içinde birbirine benzer ve diğer sınıflardan farklı özelliklere sahiptir.

5.3.1. Lojistik Regresyon (Logistic Regression)

Lojistik Regresyon doğrusal regresyondan genişletilmiştir (Tsangaratos ve Ilia, 2016). Bir hedef değişkenin olasılığını tahmin eden denetimli öğrenme sınıflandırma algoritmasıdır. Tahmine dayalı analiz yapar ve olasılık kavramına dayanır (Bittencourt vd., 2007). Lojistik Regresyon farklı veri türlerini kullanarak gözlemleri sınıflandırmak için kullanılabilir ve sınıflandırma için kullanılan en etkili değişkenleri kolayca belirleyebilir. Bağımlı ve bağımsız değişkenler arasındaki belirli bir eğri ilişkisini temsil eder (Wang ve Liu, 2020). Genellikle ikili sınıflandırma problemlerinde kullanılan bir algoritmadır. Kısaca kategorik değişkenler olan bağımlı değişkenler için regresyon analizi yapan istatistiksel bir yöntemdir. Lojistik Regresyon kategorik bağımlı değişkenin çıktısını tahmin ettiği için çıktı da kategorik değer olmalıdır. Ayrıca sonucu etkileyen başka bir deyişle çıktı üzerinde etkisi olan bağımsız değişkenlerin birbiri arasında çok az ilişki olmalı veya hiç ilişkisi olmamalıdır.

Ele alınan sınıflandırma problemine göre olayın meydana gelme olasılığını hesaplamak için matematiksel olarak ifade edilen Lojistik Regresyon eşitliği kullanır ve problemdeki olayın meydana gelme olasılığı p ile gösterilir. Kullanılan p değeri ise 0 ile 1 arasında olması gerekmektedir. Denklem 5.1'de Lojistik regresyon eşitliği verilmiştir (Tsangaratos ve Ilia, 2016).

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots \beta_n x_n \qquad (5.1)$$

Denklem 5.1'de bulunan n, elde ki bağımsız değişkenlerin toplamını gösterir. β değerleri tahmin edilecek regresyon katsayıları ve x değerleri ise bağımsız değişkenleri ifade etmektedir.

Matematiksel eşitlikte buluna p bir olasılık değeridir. Bu değerin 0 ile 1 arasına sıkışması için Sigmoid fonksiyonu kullanılır ve bu sayede sınıflandırma yapılır. Bu fonksiyon Denklem 5.2'de verilmiştir (Tsangaratos ve Ilia, 2016).

$$P = \left(\frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\beta_3 x_3+\cdots\ldots\ldots+\beta_n x_n}}\right) \qquad (5.2)$$

Son olarak Lojistik Regresyon verimli olması ve çok fazla hesaplama kaynağı gerektirmemesi nedeniyle yaygın olarak kullanılan algoritmadır ve sınıflandırma görevleri için çok iyidir. Diğer karmaşık algoritmalardan kolay bir şekilde daha iyi performans gösterebilir ve çalışması kolay ve basittir. Bununla birlikte basit yapıda olması nedeniyle diğer karmaşık algoritmaların performansıyla karşılaştırmak için iyi bir temel olarak kullanılabilir.

5.3.2. K- En Yakın Komşu (K-Nearest Neighbors)

K-En Yakın Komşu (K-NN) basitliği, etkinliği ve sağlamlığı nedeniyle en sık kullanılan sınıflandırma algoritmalarından birisidir (Batista vd., 2009). K-NN algoritması, benzer şeylerin birbirinin yakınında olduğunu varsayar. KNN, yeni veri noktalarını en yakın veri noktalarına göre sınıflandıran denetimli bir sınıflandırma algoritmasıdır. K-NN

algoritması, yeni durum ve veriler ile mevcut durumlar arasındaki benzerliği varsayar ve yeni durumu mevcut kategorilere en çok benzeyen kategoriye koyar.

Bu algoritma, parametrik olmayan bir algoritmadır. Yani temel veriler üzerinde herhangi bir varsayımda bulunmaz. Tembel öğrenen algoritma olarak da adlandırılır (Triguero vd., 2019). Nedeni eğitim kümesinden hemen öğrenmez, bunun yerine veri kümesini depolar ve sınıflandırma anında veri kümesi üzerinde bir işlem gerçekleştirir. Eğitim aşamasında sadece veri setini saklar ve yeni veri aldığında bu veriyi yeni veriye çok benzeyen bir kategoride sınıflandırır. Özetle, KNN, en yakın komşu olarak da bilinen en yakın veri noktasına bakarak bir veri noktasının sınıflandırılmasını içerir.

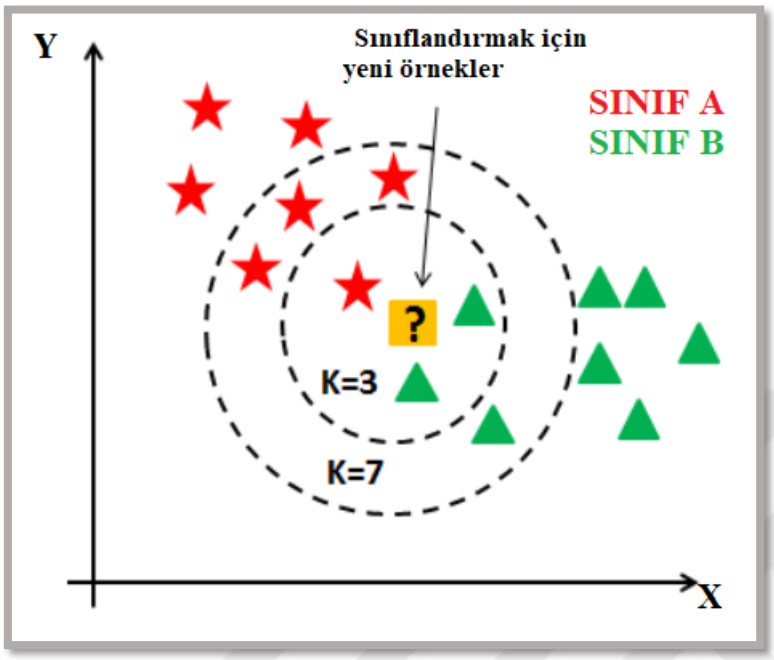
K-NN, iki örnek arasındaki farkı veya benzerliği ölçen mesafe fonksiyonuna dayanır (Jiang vd., 2007). Örnek veri setine katılacak olan yeni verinin mevcut verilere göre uzaklığı hesaplanır ve k sayıda en yakın komşuluğuna bakılır. Mesafe hesaplaması için Öklid (Euclidean) Uzaklık, Manhattan Uzaklık ve Minkowski Uzaklık fonksiyonları kullanılır. Genellikle Minkowski Uzaklık fonksiyonu kullanılır. Matematiksel olarak hesaplanması Denklem 5.3'te verilmiştir (Chomboon, 2015). Rastgele iki nokta olan P= (x_1, x_2, x_3,…., x_n) ve Q= ((y_1, x_2, y_3,…., y_n) olsun:

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{5.3}$$

Burada p farklı değerlerine göre çeşitli uzaklık ölçütleri tanımlanır. Denklemdeki p=2 ise Öklid uzaklığını, p=1 ise Manhattan uzaklığını, n= ∞ ise Chebyschev uzaklığını verir.

K-NN algoritmasının çalışma şeklinde ilk olarak k parametresi belirlenir. Bu k parametresi verilen bir noktaya en yakın komşuların sayısıdır. Sınıflandırma bu k parametresi göre yapılır (Chomboon, 2015). Ardından kullanılan uzaklık hesaplama fonksiyonu yardımı ile veri setine katılacak olan yeni verinin mevcut verilere göre uzaklığı tek tek hesaplanır. Yeni veri noktaları, ilgili uzaklığa göre en yakın k komşu ele alınır ve komşu sayısının maksimum olduğu sınıfa atanır. Böylelikle seçilen sınıf tahmin edilen

gözlemin sınıfı olarak etiketlenmiş olur. Şekil 5.4'te K-NN algoritmasının çalışma mantığı verilmiştir.



Şekil 5.4. K-NN algoritmasının çalışma mantığının gösterimi

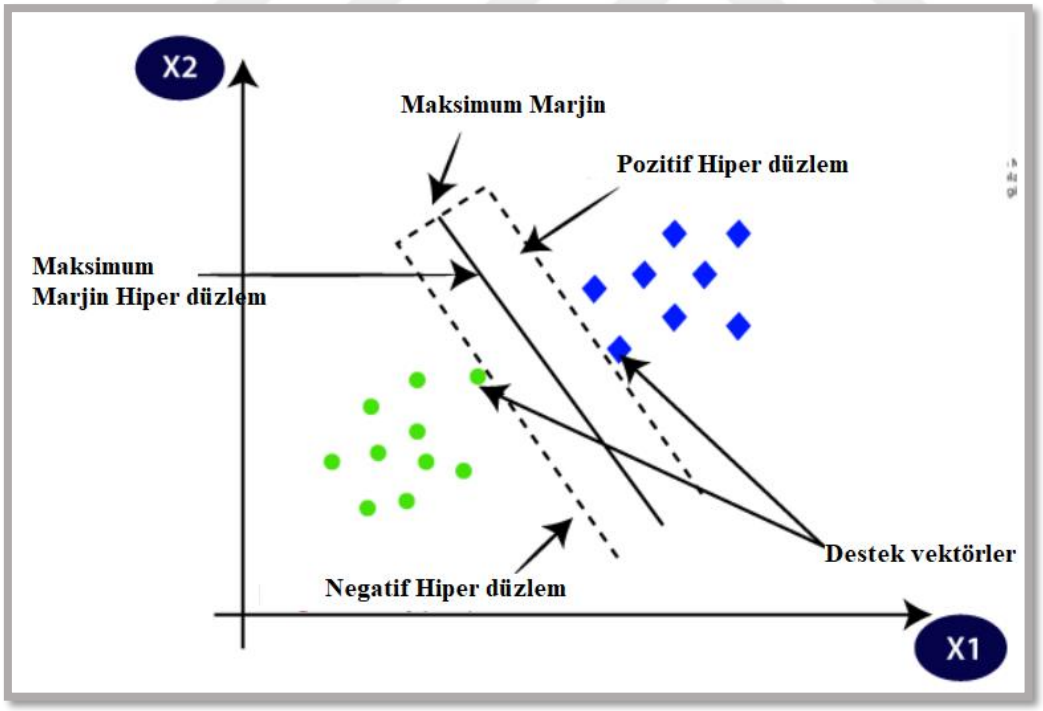
Yukarıda bulunan şekle göre, yeni örneklerin hangi sınıfa ait olduğunu bulmak için en yakın komşudan yardım alınır. K=3 demek en yakın komşu sayısı 3, K=7 ise en yakın komşu sayısı 7 demektir.

5.3.3. Destek Vektör Makine (Support Vector Machine)

Destek Vektör Makineleri (DVM) hem sınıflandırma hem de regresyon için kullanılan güçlü ve esnek bir denetimli yapay öğrenme algoritmasıdır. Genellikle sınıflandırma problemleri için kullanılır (Vafeiadis vd., 2015). DVM algoritmasının avantajı, genelleme yeteneğini geliştirebilmesi ve yüksek boyutlu veri problemleriyle başa çıkabilmesidir (Xiahou ve Harada, 2022). DVM algoritmasının amacı, gelecekte yeni veri noktasını kolay bir şekilde doğru kategoriye koyabilmek için n boyutlu uzayı sınıflara ayırabilen en iyi çizgi veya karar sınırını oluşturmaktır. Bu en iyi karar sınırını hiper düzlem olarak adlandırılır. DVM, hiper düzlemi oluşturmaya yardımcı olan uç noktaları yani vektörleri seçer. Bu uç noktalar destek vektörleri olarak adlandırılır ve bu sebeple

algoritmaya Destek Vektör Makinesi denir. Temel olarak iki sınıfı bir doğru veya düzlem ile birbirinden ayırmaya çalışır ve bu ayırma işlevini sınırda bulunan elemanlara göre yapar.

DVM algoritmasının çalışma şekli, ilk olarak sınıfları en iyi şekilde ayırmak için yinelemeli olarak hiper düzlemler üretir. Hiper düzlem, hatanın en aza indirilebilmesi için yinelemeli bir şekilde oluşturulur. Hiper düzlemin boyutu, girdi özelliklerinin miktarına bağlıdır. Girdi özelliklerinin sayısı iki ise hiper düzlem sadece bir çizgi, girdi özelliklerinin sayısı üç ise hiper düzlem iki boyutlu bir düzlem olur. Özellik sayısı üçü aştığında ise hiper düzlem boyutu artar. Hiper düzlem üretildikten sonra sınıfları doğru bir şekilde ayıran hiper düzlem seçilir (Kaynar vd., 2017). En iyi hiper düzlem seçimi, iki sınıf arasındaki en büyük ayrımı yani marjı temsil eden seçimdir. Böylelikle her iki taraftaki en yakın veri noktasına olan uzaklığı maksimize edilen hiper düzlem seçilir. DVM algoritmasının temel düzeyde çalışma mantığı Şekil 5.5'te verilmiştir. Hiper düzlem kullanarak sınıflandırılan iki farklı kategoriye göre çizilmiştir.



Şekil 5.5. DVM algoritmasının çalışma mantığının gösterimi (Kaynar vd., 2017)

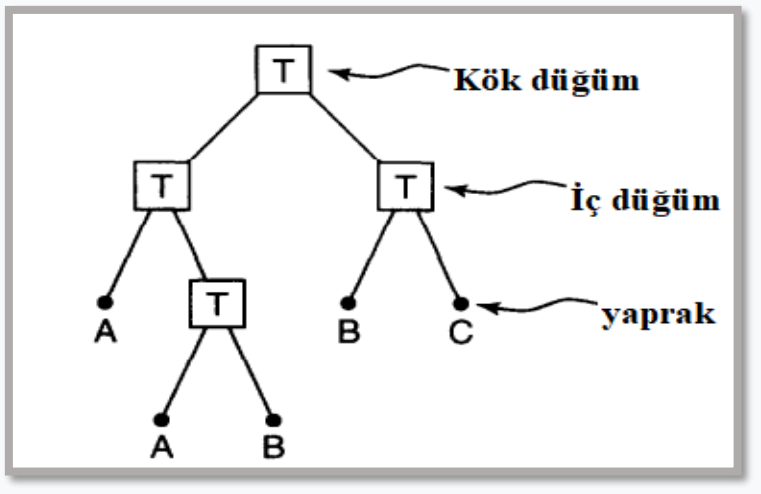
Yukarıda bulunan Şekil 5.5'e göre DVM algoritmasının çalışma mantığı özetle, en iyi çizgiyi veya karar sınırını bulmaya yardımcı olur ve bu en iyi sınır veya bölge hiper düzlemdir. DVM algoritması her iki sınıftan da doğruların en yakın noktasını bulur ve bu

noktalar destek vektörleridir. Vektörler ile hiper düzlem arasındaki uzaklık marj olarak adlandırılır. DVM algoritmasının amacı bu marjı maksimize etmektir. Maksimum marjı olan hiper düzlem, optimal hiper düzlem olarak adlandırılır.

DVM algoritması herhangi bir sayıda çekirdek (kernel) kullanır ve algoritmanın çalışma performansı bu çekirdek türüne bağlıdır (Amari vd., 1999). Yaygın olarak doğrusal çekirdek (linear kernel), polinom çekirdek (polynomial kernel) ve radyal temel fonksiyonu (radial basis function) kullanılır. Doğrusal olarak ayrılabilen sınıflandırma problemlerinde doğrusal çekirdek kullanılır. Doğrusal olmayan girdi uzayını veya eğriyi ayırt etmek için polinom çekirdek kullanılır ve doğrusal çekirdeğin genelleştirilmiş halidir. Doğrusal olmayan veri setinde sıklıkla radyal temel fonksiyonu kullanılır.

5.3.4. Karar Ağaçları (Decision Tree)

Karar Ağaçları, regresyon ve sınıflandırma için kullanılan parametrik olmayan denetimli bir öğrenme yöntemidir. Sıklıkla sınıflandırma problemleri için kullanılır (Priyanka vd., 2020). Ağaç yapısı şeklinde sınıflandırma ve regresyon modelleri oluşturulur. Karar verirken insan yeteneğini başarılı bir şekilde taklit ettiği, anlaşılması kolay olduğu için sınıflandırma ve tahminleme için en güçlü algoritmalardan birisidir. Ağaç yapısına benzer şekilde ağaç benzeri kök düğümle başlayıp ve daha fazla dalda genişlemesi sebebiyle algoritma karar ağacı olarak adlandırılmıştır. Algoritmanın çalışma mantığı ağaç yapısına benzer. Algoritmanın amacı, veri setinin özelliklerinden çıkarılan basit karar kuralları öğrenmek ve bir bağımlı değişkenin değerini tahmin eden bir model oluşturmaktır. Karar ağaçları hem kategorik hem de sürekli değişkenleri işleyebilir. Bir veri kümesini aşamalı daha küçük alt kümelere ayırırken aynı zamanda ilgili bir karar ağacı aşamalı olarak geliştirilir (Vafeiadis vd., 2015). Ağaç, bir kök düğümden başka bir ifade ile karar düğümü, iç düğüm ve yaprak düğümden oluşur (Friedl ve Brodley, 1997). Karar düğümleri tüm verilerden oluşup herhangi bir karar vermek için kullanılır ve birden fazla dal yapısına sahiptir. İç düğümler, bir veri kümesinin özelliklerini temsil eder ve bölünmeler sonucu oluşur. Yaprak düğümleri, sonucu temsil eder yani kararların çıktısıdır ve başka bir dal içermez. Karar ağaçlarının temel yapısı Şekil 5.6’ da verilmiştir.



Şekil 5.6. Karar ağacının yapısı (Friedl ve Brodley, 1997)

Yukarıda verilen Şekil 5.6 'da, T harfi kök düğümü ve iç düğümleri temsil ederken A, B ve C harfleri yaprak düğümü temsil etmektedir.

Karar ağacı algoritmasının çalışma mantığı ilk olarak verilen veri setinin sınıfını tahminlemek amacıyla ağacın kök düğümünden başlar. Bu algoritma, kök özniteliğinin değerlerini, gerçek veri setinin özniteliği ile karşılaştırır ve karşılaştırmaya dayalı olarak dalı takip eder ardından bir sonraki düğüme atlar. Bir sonraki düğüm için algoritma öznitelik değerini diğer alt düğümlerle tekrar karşılaştırır ve daha ileri gider. Ağaç, yaprak düğüme ulaşana kadar yinelenerek devam eder.

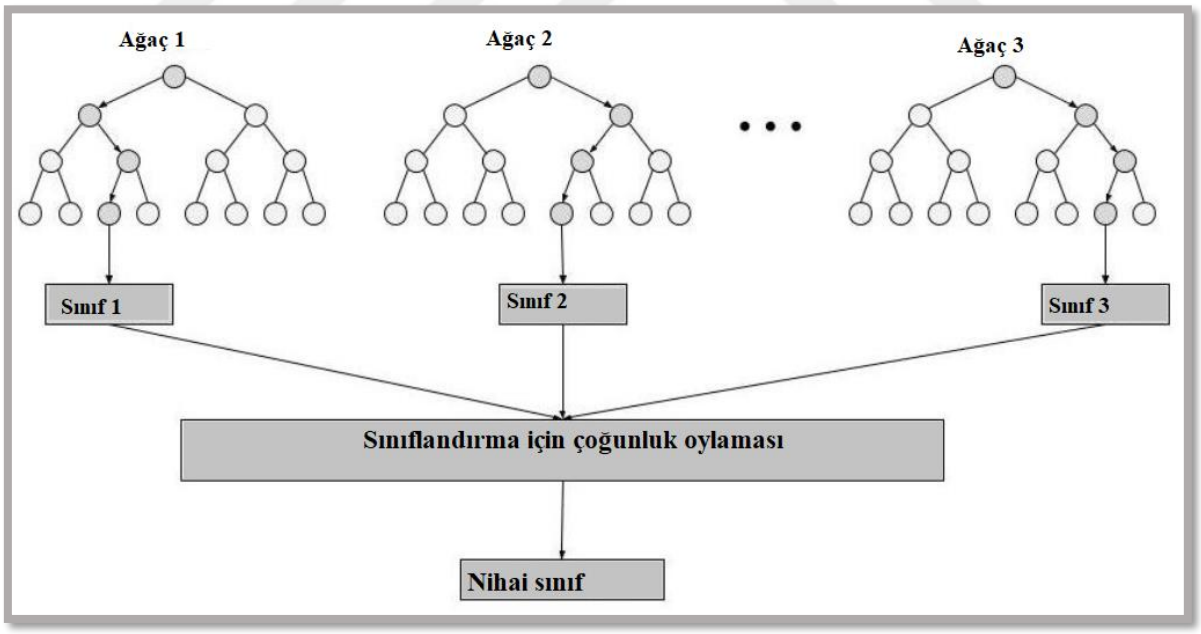
Bir karar ağacı, bir kök düğümden yukarıdan aşağıya oluşturulur ve veriler benzer değerlere sahip yani homojen örnekler içeren alt kümelere bölünmesini içerir. Homojenliği hesaplamak için entropi kullanılır. Örnek tamamen homojen ise entropi sıfır değerini alır.

5.3.5. Rastgele Orman (Random Forest)

Rastgele Orman, 2001 yılında Breiman tarafından önerilmiştir (Belavagi vd., 2016). Yapay öğrenmede kullanılan en yaygın algoritmalardan birisidir. Hem sınıflandırma hem de regresyon problemlerinde kullanılır. Rastgele ormanlar yüksek boyutlu sınıflandırma ve çarpık problemler için örüntü tanıma ve makine öğrenmesinde çok popüler ve güçlü teknikler olduğu kanıtlanmış en başarılı topluluk öğrenme tekniklerinden birisidir (Azar vd., 2014). Rastgele orman, adından da anlaşılacağı gibi, bir topluluk olarak çalışan çok sayıda

bireysel karar ağacından oluşur. Bu algoritmanın ardında bulunan temel kavram basit ama güçlü bir kavram olan kalabalıkların bilgeliğidir. Karmaşık problemleri çözmek ve model performansını daha iyi hale getirmek için birden fazla sınıflandırıcıyı birleştirme sürecidir (Belavagi vd., 2016). Rastgele orman, karar ağacı algoritmasının gelişmiş halidir. Bu algoritma, tek bir karar ağacına güvenmek yerine her ağaçtan tahmin alır ve tahminlerin çoğunluk oylarına nihai çıktıyı tahmin eder. Özetle her bir ağaç tahmin oluşturur ve en çok oyu alan sınıf, modelin tahmini olur.

Rastgele orman, iki aşamalı olarak çalışır. İlk aşamada, n adet karar ağacını birleştirerek rastgele orman oluşturur. İkinci aşamada, ilk aşamada oluşturulan her ağaç tahminleme yapar. Ayrıntılı olarak çalışma mantığında öncelikle veri setinden rastgele veri noktaları seçilir ve seçilen bu veri noktaları ile bağlantılı n adet karar ağaçları oluşturulur. Ardından yeni veri noktaları için her karar ağacı tahminde bulunur ve yeni veri noktaları çoğunluk oyu kazanan kategoriye atanır (Azar vd., 2014). Rastgele orman algoritmasının çalışma mantığı Şekil 5.7' de gösterilmiştir.



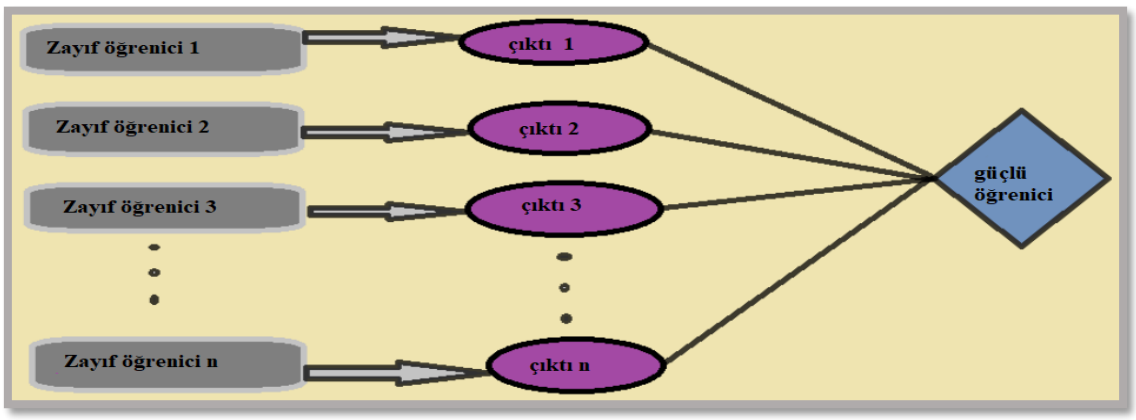
Şekil 5.7. Rastgele Orman algoritmasının çalışma mantığı (Anonim, 2021)

Sonuç olarak Rastgele orman algoritmasının sıklıkla tercih edilen bir algoritma olma nedeni kurulan model için yüksek doğruluk vermesi ve aşırı eğitim sorununu engellemesidir.

5.3.6. Artırma (Boosting)

Boosting algoritması, günümüzde kullanılan en popüler yapay öğrenme tekniklerinden birisi olmuştur. Gelişmiş öğrenme veya yükseltme yöntemi olarak da bilinen Boosting algoritması, sınıflandırma metodolojisindeki en önemli gelişmelerden birisidir ve yüksek performanslı tahmin kuralı oluşturmak için bir dizi zayıf sınıflandırıcıyı birleştirme tekniğidir (Wu ve Meng, 2016). Boosting algoritmasını anlamak için bu algoritmanın belirli bir modelden ziyade genel bir algoritma olduğunu bilmek gerekir. Bu algoritma, eğitim hatalarını en aza indirmek için bir dizi zayıf öğreniciyi güçlü bir öğrenicide birleştiren bir topluluk öğrenme yöntemidir (Freund vd., 2003). Boosting algoritmasının, birincil işlevi zayıf öğrenenleri güçlü öğrenenlere dönüştürmek olduğu için zayıf bir model belirlenir ve ardından onu iyileştirme çalışmaları yapılır. Tek bir model tarafından yapılan yüksek kaliteli tahmine odaklanan birçok yapay öğrenme modelinin aksine Boosting algoritmaları, her biri önceki modellerin zayıflıklarını telafi eden bir dizi zayıf modeli eğiterek tahmin gücünü iyileştirmeye çalışır.

Boosting algoritmasının genel çalışma mantığında, ilk olarak eğitim veri setinden bir model oluşturulur. Ardından, oluşturulan ilk modelde ki hataları düzeltmeye çalışan ve en aza indiren ikinci model oluşturulur (Burez ve Van Del Poel, 2009). Bu şekilde yinelemelere devam edilir ve tüm eğitim veri seti doğru tahmin edilene veya maksimum model sayısı eklenene kadar modeller eklenir. Her yinelemede, her bir sınıflandırıcıdan gelen zayıf kurallar, tek bir güçlü tahmin kuralı oluşturmak için birleştirilir. Boosting algoritmasının çalışma mantığı temel olarak Şekil 5.8’ de verilmiştir.



Şekil 5.8. Boosting algoritmasının temel düzeyde çalışma mantığı

Şekil 5.8'de, birden çok zayıf öğrenici nihai çıktı oluşturmak için birleştirilir. Böylelikle güçlü bir öğrenici oluşturulur.

Literatürde farklı Boosting algoritmaları mevcuttur. AdaBoost, Gradient Boosting Machines (GBM), Extreme Gradient Boosting (XGBoosting), Light Gradient Boosting Model (LightGBM) ve CatBoost sıklıkla kullanılan algoritmalardır.

5.3.6.1. Aşırı Gradyan Artırma (Extreme Gradient Boosting)

Yapay öğrenmede kullanılan en popüler Boosting algoritmalarından birisidir ve Gradyan Artırma (Gradient Boosting) algoritmasının çalışma prensibi ile çalışır (Gong, 2021). Karmaşık problemleri ve büyük veri setine sahip problemleri başarılı, hızlı ve etkili bir şekilde modelleme yapmaktadır. Diğer modellere kıyasla, aşırı eğitimi önlemek amacıyla daha düzenli model takviyesi sağlar ve böylelikle performansı artırır. Paralel ağaç yapısından oluşan hızlı bir algoritmadır.

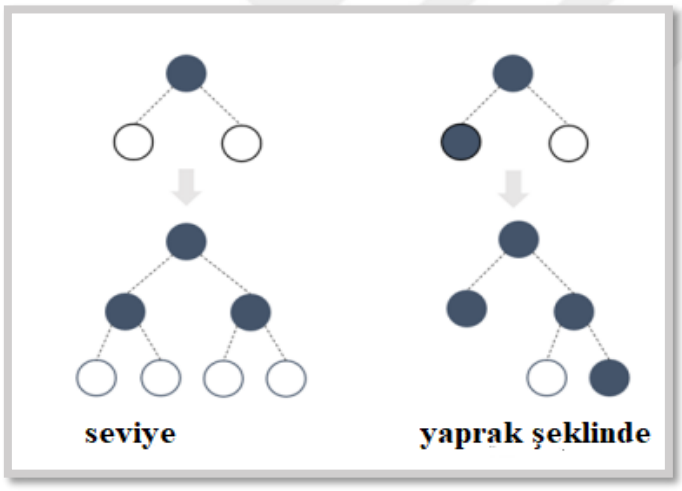
Bu algoritmada karar ağaçları sıralı olarak oluşturulur. Ağırlıklar XGBoost algoritmasında önemli bir rol oynar. Ağırlıklar, daha sonra sonuçları tahmin eden karar ağacına beslenen tüm bağımsız değişkenlere atanır. Ağacın yanlış tahmin ettiği değişkenlerin ağırlığı artırılır ve bu değişkenler daha sonra ikinci karar ağacına beslenir. Bu bireysel sınıflandırıcılar yani tahminciler daha sonra güçlü ve daha kesin bir model vermek için bir araya gelir.

XGBoost algoritması, GBM algoritmasından bazı yönlerden daha üstündür (Al Daoud, 2019). GBM algoritması, bir karar ağacı kurulduktan sonra bu ağacın yaptığı hatalar ile ikinci bir karar ağacının kurulması ilkesine dayanır ve bu yüzden ilk ağaç bitmeden ikincisinin hesaplanmasına başlanamaz. XGBoost algoritmasında ise bir karar ağacı kurulurken dallarını paralelde oluşturabilir. Diğer farklılıkları ise daha esnek olması, eksik değerleri kullanması ve düzenlileştirme özelliklerine sahip olmasıdır (Bentéjac vd., 2021).

5.3.6.2. Hafif Gradyan Artırma (Light Gradient Boosting Model)

Son yıllarda veri boyutunun artması ve çeşitlenmesi nedeniyle algoritmalar geliştirilerek varyantları geliştirilmiştir. LightGBM algoritması, GBM algoritmasına alternatif olarak geliştirilmiştir (Bentéjac vd., 2021). LightGBM, modelin verimliliğini artırmak ve bellek kullanımını azaltmak için karar ağaçlarına dayalı bir GBM çerçevesidir.

LightGBM algoritması, histogram tabanlı çalışır. Sürekli olan değerlere sahip değişkenleri kesikli hale getirip hesaplama hızını arttırır. Bu algoritma ile hem eğitim süresi azalır hem de kaynak kullanımı düşer (Ju vd., 2019). Diğer algoritmalar yatay olarak ağaçları büyütürken, bu algoritma dikey olarak yani yaprak şeklinde büyür. LightGBM algoritmasının ağaç oluşturma stratejisi Şekil 5.9' da gösterilmiştir. Büyümek için büyük kayıp olan yaprağı seçer.



Şekil 5.9. LightGBM algoritmasının ağaç oluşturma stratejisi (Ju vd., 2019)

5.4. Model Performans Değerlendirme Ölçütleri

Performans metrikleri, yapay öğrenme modeli geliştirme, seçme ve değerlendirme için önemli bir rol oynamaktadır ve bir yapay öğrenme modelinin performansını değerlendirmek, etkili bir yapay öğrenme modeli oluştururken önemli adımlardan birisidir (Gong, 2021). Modelin performansını veya kalitesini değerlendirmek için farklı metrikler kullanılmaktadır ve bu metrikler performans metrikleri veya değerlendirme metrikleri olarak bilinir. Yapay öğrenme modelleri için performans metriklerini takip etmek çok önemlidir.

Bu performans metrikleri, modelin verilen veriler için ne kadar iyi performans gösterdiğini anlamaya yardımcı olmaktadır. Bu sayede hiper parametreleri ayarlayarak modelin performansı iyileştirilebilir. Modelin nasıl çalıştığını anlamak, başka fikirler ile karşılaştırabilmek ve hedeften ne kadar uzakta olduğunu anlamak için performans değerlendirme ölçütleri önemli bir fayda sağlar.

Yapay öğrenme modelini değerlendirmek için hangi performans metriğinin kullanılacağı önemli bir ayrıntıdır. Metrik seçimi, yapay öğrenme algoritmalarının performansının nasıl ölçüldüğünü ve karşılaştırıldığını da etkilemektedir. Bu yüzden gözlem yöntemine bağlı olarak, modelin performansını değerlendirmek için farklı ölçüler vardır. Sorunun özelliklerine ve uygulama yollarına bağlı olarak en uygun önlemlerin seçimi yapılacaktır (Novaković vd., 2017). Sınıflandırma problemleri için sıklıkla kullanılan performans değerlendirme ölçütleri:

- Doğruluk (Accuracy)
- Karışıklık matrisi (Confusion Matrix)
- F-Skor (F-Score)
- Kesinlik (Precision)
- Duyarlılık (Recall)
- Eğrinin Altındaki Alan (Area under the curve)

5.4.1. Karışıklık Matrisi (Confusion Matrix)

Karışıklık Matrisi, çıktının iki veya daha fazla sınıf olabileceği yapay öğrenme sınıflandırma problemleri için bir performans ölçümüdür (Wu ve Meng, 2016). Tahmin ve gerçek değerlerin kombinasyonlarını içeren bir tablodur. Karışıklık Matrisi adından da anlaşılacağı üzere çıktı olarak bir matris vermekte ve gerçek değerlerin bilindiği bir dizi test verisi üzerindeki bir sınıflandırma modelinin performansını göstermek için sıklıkla kullanılan tablo olarak kabul edilmektedir. Karışıklık matrisinin Şekil 5.10'da gösterimi verilmiştir.

		Gerçek Değer	
		Doğru	**Yanlış**
Tahmin Edilen Değer	**Doğru**	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
	Yanlış	Yanlış Negatif (YN)	Doğru Negatif (DN)

Şekil 5.10. Karışıklık Matrisi (Başer vd., 2021)

Yukarıda bulunan Şekil 5.10'da performans ölçütlerini hesaplamak için önemli kavramlar kullanılır:

- Doğru pozitif (DP), tahmin edilen sonuç doğrudur ve gerçekte de doğrudur. Yani veri noktasının gerçek sınıfının doğru (1 olduğu durum) ve tahmin edilenin de doğru olduğu (1 olduğu durum) durumlardır.
- Doğru negatif (DN), tahmin edilen sonuç yanlıştır ve gerçekte de yanlıştır. Yani veri noktasının gerçek sınıfının yanlış (0 olduğu durum) ve tahmin edilenin de yanlış olduğu (0 olduğu durum) durumlardır.
- Yanlış pozitif (YP), tahmin edilen sonuç doğrudur fakat gerçekte yanlıştır. Yani veri noktasının gerçek sınıfının yanlış (0 olduğu durum) ve tahmin edilenin de doğru olduğu (1 olduğu durum) durumlardır.
- Yanlış negatif (YN), tahmin edilen sonuç yanlıştır fakat gerçekte doğrudur. Yani veri noktasının gerçek sınıfının doğru (1 olduğu durum) ve tahmin edilenin de yanlış olduğu (0 olduğu durum) durumlardır.

Karışıklık matrisinde bulunan bu değerler kullanılarak sınıflandırma probleminin performans ölçütleri hesaplanmaktadır.

5.4.2. Doğruluk (Accuracy)

Doğruluk, en sık kullanılan performans metriklerinden birisidir. Doğruluk, verilerdeki hedef değişken sınıfları neredeyse dengeli olduğunda iyi bir ölçüttür. Sınıflandırıcının ne sıklıkla doğru tahminde bulunduğunu ölçmektedir. Doğruluk metriği, doğru tahmin sayısının, toplam tahmin sayısına oranı şeklinde hesaplanmaktadır. Karışıklık

matrisinde bulunan değerler kullanılarak hesaplanmaktadır. Denklem 5.4' te doğruluk ölçütünün hesaplanması verilmiştir (Vafeiadis vd., 2015).

$$\text{Doğruluk} = \frac{DP + DN}{DP + YP + YN + DN} \tag{5.4}$$

5.4.3. Kesinlik (Precision)

Kesinlik, doğru tahmin edilenin kaçının gerçekte pozitif çıktığını açıklar. Sınıflandırıcı tarafından tahmin edilen pozitif sonuçların sayısına bölünen doğru pozitif sonuçların sayısıdır (Vafeiadis vd., 2015). Denklem 5.5' te kesinlik ölçütünün hesaplanması verilmiştir.

$$\text{Kesinlik} = \frac{DP}{DP + YP} \tag{5.5}$$

5.4.4. Duyarlılık (Recall)

Model tarafından döndürülen pozitiflerin sayısıdır. Doğru pozitiflerin toplam gerçek pozitif sayısına bölünmesi şeklinde hesaplanmaktadır. Yani modelin gerçek pozitif vakaların kaç tanesini doğru tahmin edebildiğini açıklar. Denklem 5.6' da hatırlama ölçütünün hesaplanması verilmiştir (Vafeiadis vd., 2015).

$$\text{Duyarlılık} = \frac{DP}{DP + YN} \tag{5.6}$$

5.4.5. F1 Skor (F1 Score)

F1 Skor, kesinlik (precision) ve duyarlılık (recall) arasındaki harmonik ortalamadır. Kesinlik ve hatırlama oranlarının ikisinin de iyi olduğu bir algoritma aranması durumunda bu iki değerin harmonik ortalaması olan F1 Skor kullanılmaktadır. Kesinlik ve duyarlılık eşit olduğu durumda F1 Skor maksimum olmaktadır. F1 Skor 0 ile 1 arasında değer almaktadır. Denklem 5.7'de bu ölçütün hesaplanması verilmiştir (Wu ve Meng, 2016).

$$\text{F1 Skor} = 2 * \frac{Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık} \tag{5.7}$$

5.4.6. AUC (Area Under the ROC Curve)

Eğri Altındaki Alan (AUC), model değerlendirmesi için en yaygın kullanılan metriklerden birisidir. İkili sınıflandırma problemi için kullanılmaktadır. Sınıflandırma modelinin performansını grafikler üzerinde görselleştirmek gerekmekte ve bu yüzden AUC-ROC eğrisi sıklıkla tercih edilmektedir. AUC, ROC (Receiver Operating Characteristic Curve) eğrisi ve koordinat ekseni tarafından çevrelenen alan olarak tanımlanmaktadır (Burez vd., 2009).

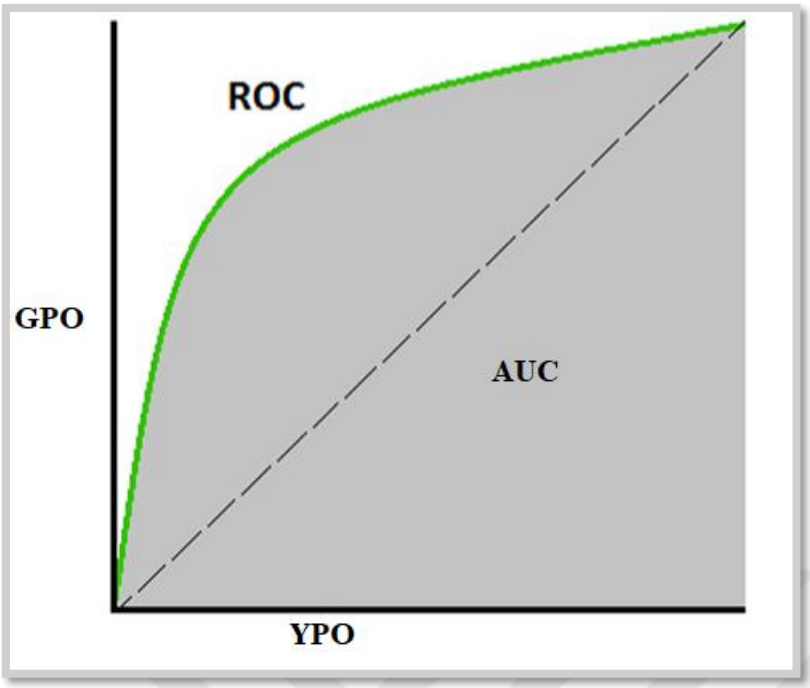
AUC, tüm ROC eğrisi altındaki iki boyutlu alanı hesapladığı için ROC olasılık eğrisidir. AUC ise ayrılabilirliğin derecesini veya ölçüsünü temsil etmektedir. (Narkhede, 2018). AUC, ROC eğrisinin altındaki alan, ROC eğrisinin sınıflandırma performansını çok iyi ölçemediği için genel bir nicel göstergesidir (Wu ve Meng, 2016).

AUC hesabı için ROC eğrisinin çizilmesi gerekir. ROC, farklı eşik seviyelerinde bir sınıflandırma modelinin performansını gösteren bir grafiği temsil eder. Eğri, iki parametre arasında çizilir. Gerçek pozitif oran (GPO) ve yanlış pozitif oranı (YPO) kullanılarak eğri çizilir. Bu değerler Denklem 5.8 ve Denklem 5.9 gibi hesaplanır.

$$\text{Gerçek Pozitif Oranı} = \frac{DP}{DP + DN} \tag{5.8}$$

$$\text{Yanlış Pozitif Oran} = \frac{YN}{YN + YP} \tag{5.9}$$

Bir ROC eğrisinin herhangi bir noktasındaki değeri hesaplamak için AUC olarak bilinen verimli bir yöntem kullanılır. AUC, tüm eşiklerdeki performansı hesaplar ve toplu bir ölçüm sağlar. Şekil 5.11’ de AUC-ROC eğrisi gösterilmiştir.



Şekil 5.11. AUC-ROC eğrisi (Narkhede, 2018)

Şekil 5.11'de gösterilen grafikte x ekseni yanlış pozitif oranı, y ekseni ise gerçek pozitif oranı gösterir. AUC ise bu iki bu iki eksen kullanılarak çizilen ROC eğrisi altındaki alanı hesaplar. AUC, 0 ile 1 arasında değer alır (Burez ve Van Del Poel, 2009). Bu durum, %100 yanlış tahmine sahip bir modelin AUC değeri 0 iken, %100 doğru tahminlere sahip modellerin AUC değeri 1 olacaktır. Değer ne kadar büyük olursa, modelin performansı o kadar iyi olur.

5.5. Hiper Parametre Ayarlama (Hyperparameter Tuning)

Yapay öğrenme modellerinde ayarlama, bir modeli aşırı eğitmeden veya yüksek bir varyansa neden olmadan performansı maksimize etme sürecidir. Bu ayarlama süreci uygun hiper parametreler seçilerek gerçekleşir. Uygun bir hiper parametre seti seçmek, model doğruluğu için çok önemlidir (Bardenet vd., 2013). Hiper parametre ayarlama, modelleri en doğru sonuçları üretecek şekilde özelleştirmeye ve verilere ilişkin son derece değerli bilgiler sunmaya yardımcı olur. Hiper parametreler, modelin nasıl çalıştığını etkileyen harici kontroller olarak düşünülür. Bu değerler modelin dışındadır ve kullanıcı tarafından kontrol edilir. Bir algoritmanın nasıl eğitildiği nihai modelin yapısını etkileyebilir. Farklı algoritmalar farklı hiper parametrelerden oluşur (DeCastro-García vd., 2019). Her algoritma için başlangıç noktasını sağlayan ve algoritmanın varsaydığı hiper parametre seti vardır.

Ancak bu her veri seti için uygun olmayabilir. Algoritmalar için en iyi hiper parametreyi bulmak için belirli ayarlama işlemlerini manuel olarak yapmak gerekir. Uygun hiper parametreleri seçmek için birçok yöntem mevcuttur. GridSearchCV ve Rastgele Arama Yöntemi (Random Search CV) sıklıkla kullanılan yöntemlerdir (Yang ve Shami, 2020).

5.5.1. GridSearchCV

GridSearchCv yöntemi, en sık kullanılan geleneksel yöntemdir. Bu yöntem, hiper parametrik uzayın bir alt kümesinin manuel olarak tanımlanmasını ve belirtilen hiper parametre alt kümelerinin tüm kombinasyonlarının denenmesini içerir. En iyi performans gösteren hiper parametrik kombinasyon seçilir. En iyi sonucu verir (Yang ve Shami, 2020).

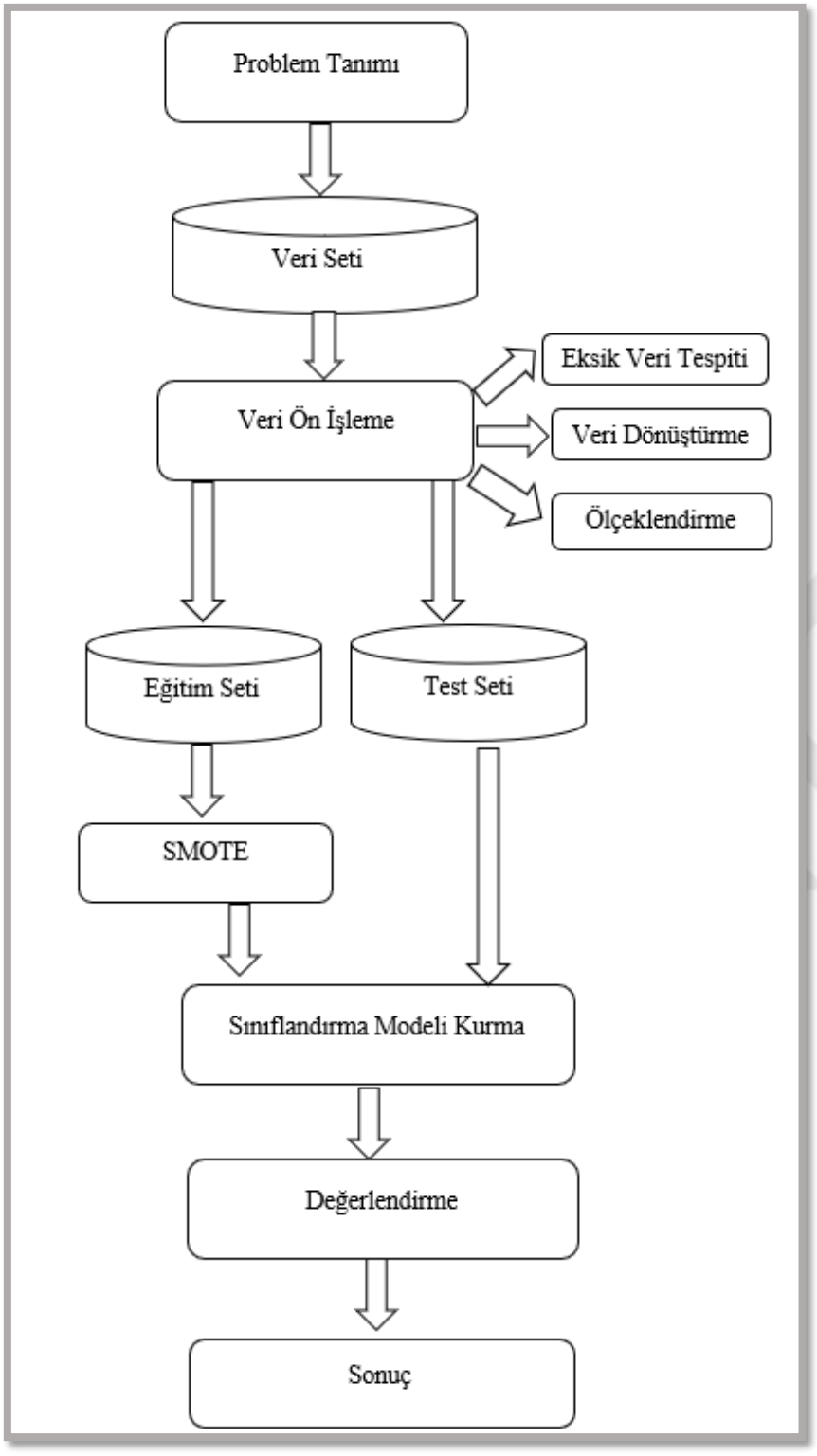
Rastgele Arama Yöntemi (Random Search CV) ise parametrelere ait dağılım bilgileri verilir bu aralıklara göre rastgele değerler seçilerek en iyi sonuca ulaşmaya çalışır. Ayrıca Rastgele Arama Yöntemi daha geniş parametre kümesi ile çalışmaktadır.

5.6. SMOTE (Synthetic Minority Over-sampling Technique)

Sentetik azınlık aşırı örnekleme tekniği (SMOTE), veride bulunan sınıf dengesizlik sorununu çözmek için en yaygın olarak kullanılan aşırı örnekleme yöntemlerinden birisidir (Wang vd., 2006). Gerçek dünya uygulamalarında, veri kümeleri genellikle dengesizdir ve bu yüzden dengesiz verilerin doğru bir şekilde nasıl sınıflandırılacağı önemli bir sorun haline gelmiştir. Dengesiz veri, sınıfların eşit dağılmadığı yani her sınıf için yaklaşık olarak aynı sayıda verinin olmadığı bir durumdur (Li vd., 2014). Örneğin, ikili sınıflandırma uygulaması için 800 verinin olduğu veri setinde, 70 veri azınlık sınıfa (sınıf-1) ve 730 verinin ise diğer sınıfa (sınıf-0) ait olmasıdır. Bu durumda veride bir dengesizlik söz konusudur. Dengesizlik durumunda algoritma sınıflandırma yaparken azınlık olan sınıfı dikkate almadan çoğunluğa göre bir sınıflandırma yapabilir. Bu hatanın önüne geçmek için, veri dengesizliğini önlemek ve azınlık sınıfı doğru sınıflandırmak gerekir. Bunun için SMOTE kullanılır. Böylece aşırı öğrenme engellenerek azınlık olan sınıftan sentetik örnekler oluşturulur. SMOTE, azınlık sınıfında bulunan örnekleri aşırı örnekler. Yapay öğrenme modeli eğitilmeden önce azınlık sınıfı örnekleri çoğaltılır ve böylece sınıf dağılımı dengelenir. Sonuç olarak SMOTE aşırı öğrenmenin önüne geçerek ve yüksek performanslı bir sınıflandırma olanağı sağlar.

6. BULGULAR VE TARTIŞMA

Bu çalışmada e-ticaret sektöründeki müşteri kaybı tahmini yapmak ve model geliştirmek amacıyla yapay öğrenme teknikleri kullanılmıştır. Uygulama için yapay öğrenme tercih edilme sebebi, model hangi değişkenleri veya özellikleri analiz edeceğini ve tahmin geliştirmek için hangilerini kullanacağını belirlemesi ve hızlı, güvenilir sonuçlar vermesidir. Çalışmada önemli bir e-ticaret şirketinin veri seti kullanılmıştır. Bu veri seti kullanılarak kayıp (churn) müşteri tahmini yapmak için sınıflandırma algoritmaları uygulanmış ve tahmin modelleri oluşturulmuştur. Model oluşturmak için kodlama dili olarak Python ve arayüz olarak ise Pycharm tercih edilmiştir. Veri setine ilk olarak ön işleme yapılmış ve veri setinde dengesizlik durumu olduğu için SMOTE metodu uygulanmıştır. Ardından tahmin modeli oluşturmak için veri setine yapay öğrenme algoritmaları uygulanmıştır. Tahmin modelleri birbirleri arasında performans ölçütüne göre kıyaslanmış ve en iyi değeri veren model en güvenilir model olarak seçilmiştir. Son adım ise en iyi model için değişken önem düzeyi belirlenmiştir. Yapay öğrenmede sınıflandırma algoritmaları olarak Lojistik Regresyon, K-En Yakın Komşu, Destek Vektör Makine, Karar Ağacı, Rastgele Orman, XGBoost ve LightGBM kullanılmıştır. Şekil 6.1'de bu çalışmanın genel çerçevesi verilmiştir.

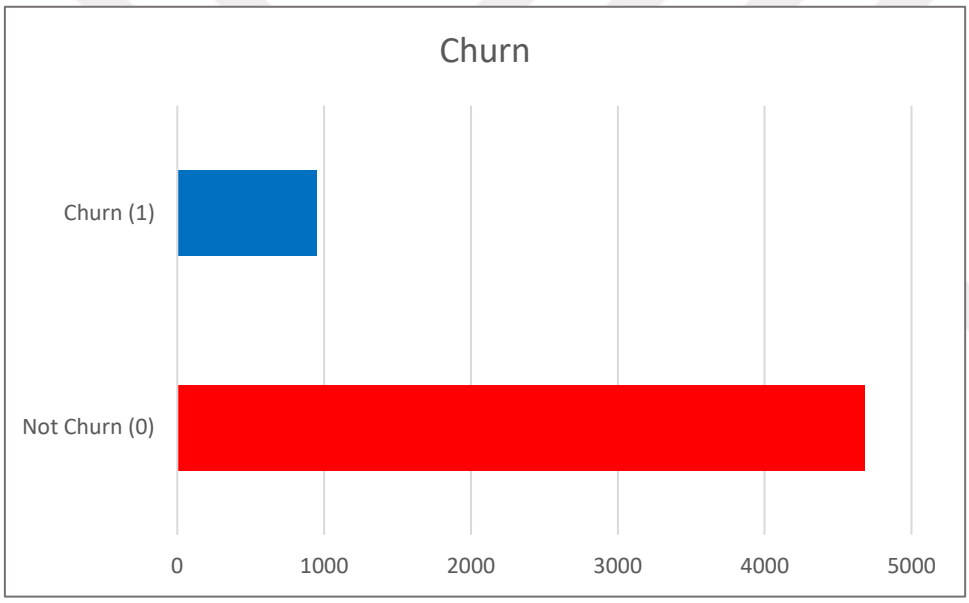


Şekil 6.1. Çalışmanın iş akış şeması

Yukarıda gösterilen iş akış şemasında bu çalışmada yapılacak adımlar genel hatları ile gösterilmiştir.

6.1. Veri Setinin Tanımı

Uygulamada kullanılan veri seti açık erişime sahip Kaggle adlı internet sitesinden alınmıştır. Veri seti, önde gelen bir e-ticaret firmasına aittir. Müşteriyi ve müşterinin e-ticaret firmasında ki alışveriş deneyimlerini içermektedir. Müşteri deneyimi sonucu e-ticaret firmasında ki alışverişine ya devam eder ya da devam etmez ve ayrılır. Bunun sonucu olarak veri setinde ikili sınıf değişkeni kayıp ve kayıp değil şeklinde tanımlanmıştır. Veri setinde kayıp müşteri sınıfı 'Churn' olarak adlandırılmıştır. Eğer müşteri kayıp ise 1 değerini alırken kayıp değil ise 0 değerini alır. Veri seti, 5630 müşterinin etkileşimini ve tercihini gösterir. Şekil 6.2' de Churn sınıfının veri setinde dağılımı gösterilmiştir.



Şekil 6.2. Müşteri kaybının veri setinde dağılımı

Yukarıda bulunan Şekil 6.2' de, veri setinde 948 müşteri kayıp, 4682 müşteri ise kayıp değildir. Yani 948 müşteri, artık e-ticaret firmasında müşteri olarak devam etmek istemezken 4682 müşteri ise sadık müşteri olmaya devam etmektedir. Veri setindeki dağılıma göre dengesizlik dikkat çekmektedir.

6.1.1. Öznitelik tanımı

Müşteriler çevrimiçi alışveriş yaparken birçok durumdan etkilenir. Müşteri olarak ya devam eder ya da devam etmezler. Bu sebeple müşteri kaybına neden olan birçok etken

vardır. Bu etkenler öznitelik olarak adlandırılır. Veri setinde, satırlar farklı müşterileri ve her bir sütun öznitelikleri temsil eder. Veri seti, 19 sütundan oluşmaktadır. Sütunların 18 tanesi müşterinin kaybını etkileyen farklı faktörlerdir. Bu faktörler bağımsız değişken (X) olarak adlandırılır. Sütunun 1 tanesi ise Churn sınıfıdır. Bu sütun ise hedef değişken (Y) olarak adlandırılır. Çizelge 6.1' de veri setinde bulunan öznitelikler tanımlanmıştır.

Çizelge 6.1. Veri setindeki öznitelik tanımı

	Öznitelik	**Açıklama**	**Tür**
1	Churn	Kayıp durumu	Sayı
2	Tenure	E-ticaret firmasında bulunma süresi	Sayı
3	PreferredLogindevice	Müşterinin siteye giriş için tercih ettiği cihaz	Karakter
4	CityTier	Şehir seviyesi (Şehir Katmanı)	Sayı
5	WarehouseToHome	Müşterinin depo ile evi arasındaki mesafe	Sayı
6	PreferredPaymentMode	Müşterinin tercih ettiği ödeme yöntemi	Karakter
7	Gender	Müşterinin cinsiyeti	Karakter
8	HourSpendOnApp	Mobil uygulama veya web sitesinde harcanan saat	Sayı
9	NumberOfDeviceRegistered	Kayıtlı cihaz sayısı	Sayı
10	PreferredOrderCat	Müşterinin geçen ay tercih ettiği sipariş kategorisi	Karakter
11	SatisfactionScore	Müşterinin hizmetten memnuniyet puanı	Sayı
12	MaritalStatus	Müşterinin medeni hali	Karakter
13	NumberOfAddress	Müşterinin kayıtlı Adres sayısı	Sayı

Çizelge 6.1. Veri setindeki öznitelik tanımı (devamı)

14	Complain	Müşterinin son ayda yaptığı şikâyet sayısı	Sayı
15	OrderAmountHikeFromlastYear	Geçen yıla göre sipariş yüzde artışı	Sayı
16	CouponUsed	Geçen ay kullanılan toplam kupon sayısı	Sayı
17	OrderCount	Geçen ay verilen toplam sipariş sayısı	Sayı
18	DaySinceLastOrder	Müşterinin son siparişinden bu yana geçen gün sayısı	Sayı
19	CashbackAmount	Geçen ay ortalama geri ödeme	Sayı

Öznitelik ismi, açıklaması ve öznitelik türü tabloda tanımlanmıştır. Veri seti için öznitelik türü ya sayısal ya da karakterdir. Özniteliklerden 5 tanesi karakter yani kategorik türüne sahipken, 14 tanesi sayısal yani nümerik türüne sahiptir.

6.2. Kodlama ve Kütüphaneler

Bu uygulama için Python yazılım dili kullanılmıştır. Kodlama için arayüz olarak programlama alanında kullanılan çok platformlu bir IDE (entegre geliştirme ortamı) olan Pycharm tercih edilmiştir.

Kodlamalar yapabilmek için kütüphanelere ihtiyaç duyulmuştur. Bu yüzden çalışma için birçok kütüphane içe aktarılmıştır.

- İlk olarak veri setine ön işleme yapabilmek ve veri analizi için uygun hale getirebilmek için “Pandas” kütüphanesi kullanılmıştır.
- Çok boyutlu diziler ve matrislerle çalışma yapabilmek ve matematiksel hesaplamalar için “NumPy” kütüphanesi içe aktarılmıştır.
- Veri setini ve sonuçlarını görselleştirmek için “Matplotlib”, “Seaborn” ve “Missingno” kütüphanesi kullanılmıştır.
- Çok sayıda yapay öğrenme algoritması içerdiği için modelleme yapabilmek ve değerlendirmek için “Scikit-learn” kütüphanesi içe aktarılmıştır.

6.3. Veri Ön İşleme (Data Preprocessing)

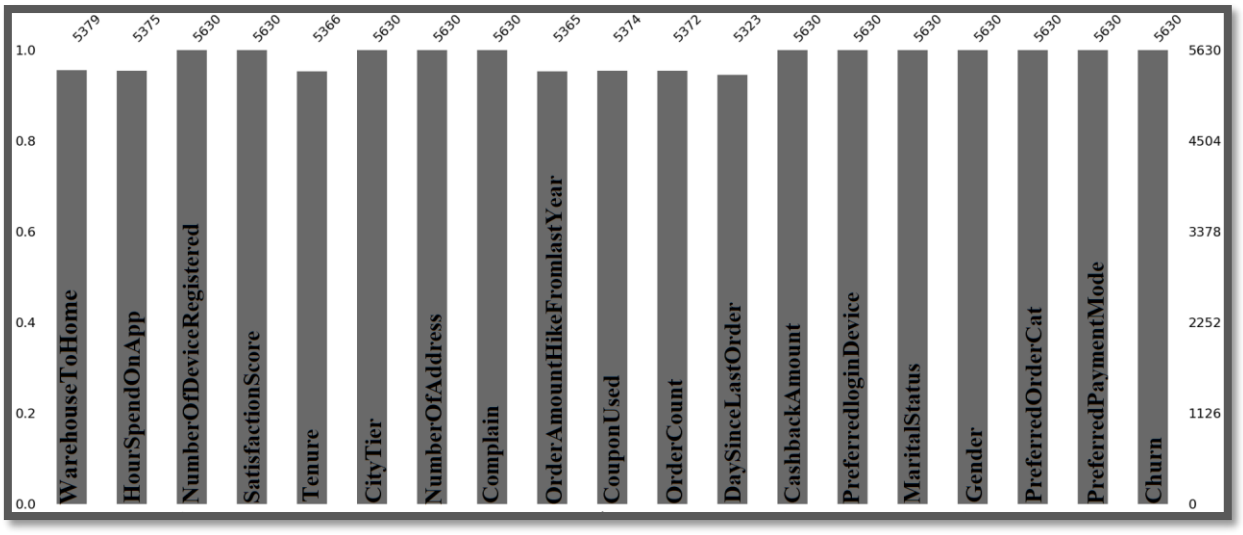
Veri setini analiz edebilmek ve analize uygun hale getirebilmek için, kullanılan veri seti veri ön işleme adımından geçmesi gerekir. Veri setinin doğru yorumlanması ve model kurulabilmesi için veri ön işleme çok önemli bir adımdır. Bu yüzden sonraki adımlar için ilk aşamada veri seti bu işlem ile temiz ve kullanılabilir hale getirilmiştir.

İlk olarak çeşitli işlemleri rahat bir şekilde gerçekleştirebilmek için kütüphaneleri içe aktarma (import) işlemi yapılmıştır. Ardından veri seti üzerinde çalışabilmek için veri seti dosyası içe aktarılmıştır. Sırasıyla veri ön işleme adımları uygulanmıştır.

Veri ön işlemeye başlamadan önce her bir müşteriyi ifade ID sütunu veri setinden çıkarılmıştır. Nedeni modelleme yaparken yapay öğrenmeye hiçbir katkısı olmamasıdır.

6.3.1. Eksik veri analizi (Missing value analysis)

Veri setinin eksik veriye sahip olup olmadığı kontrol edilmiştir. Kontrol sonucu 5630 veriden, toplam 1856 verinin eksik olduğu tespit edilmiştir. Eksik verilerin, hangi özniteliklerde olduğunu net görmek amacıyla Python'da "Missingno" kütüphanesi kullanılmış ve özniteliklerin ne kadarının dolu olduğunu görmek için grafik çizdirilmiştir. Şekil 6.3' te tüm özniteliklerin veri sayıları çizdirilmiştir.



Şekil 6.3. Her bir öznitelik sütunu için dolu veri sayısı

Şekil 6.3 ‘te bulunan grafiğe göre “WarehouseToHome” öznitelik sütununda 251 tane veri, “HourSpendOnApp” öznitelik sütununda 255 tane veri, “Tenure” öznitelik sütununda 264 tane veri, “OrderAmountHikeFromlastYear” öznitelik sütununda 265 tane veri, “CouponUsed” öznitelik sütununda 256 tane veri, “OrderCount” öznitelik sütununda 258 tane veri, “DaySinceLastOrder” öznitelik sütununda 307 tane verinin eksik olduğu tespit edilmiştir.

Yaklaşık veri setinin %33’ü eksik veriye sahiptir. Bu çalışmada veri setinde büyük miktarda veri eksikliği olduğu için eksik olan veri satırlarını silmek yerine eksik olan verileri türetme işlemi yapılmıştır. Böylelikle veri setinin kalitesi düşürülmemiştir. Eksik verinin olduğu ilgili öznitelik sütunun ortalaması alınır ve ilgili öznitelik sütununda bulunan eksik verilerin olduğu her yere bu hesaplanmış olan ortalama değeri koyulur. Bu işlem kayıp verilerin olduğu her öznitelik sütunu için ayrı olarak yapılır. Sonuç olarak, eksik olan veriler türetilmiş olur. Bu işlemlerin hepsi Python’da bulunan “SimpleImputer” sınıfı (class) ile yapılmıştır. Bu modül sayesinde veri setinde bulunan her eksik veri bulunmuş ve boş olan yerler ortalama değer ile doldurulmuştur. Böylelikle veri setinde eksik veriler tamamlanmıştır.

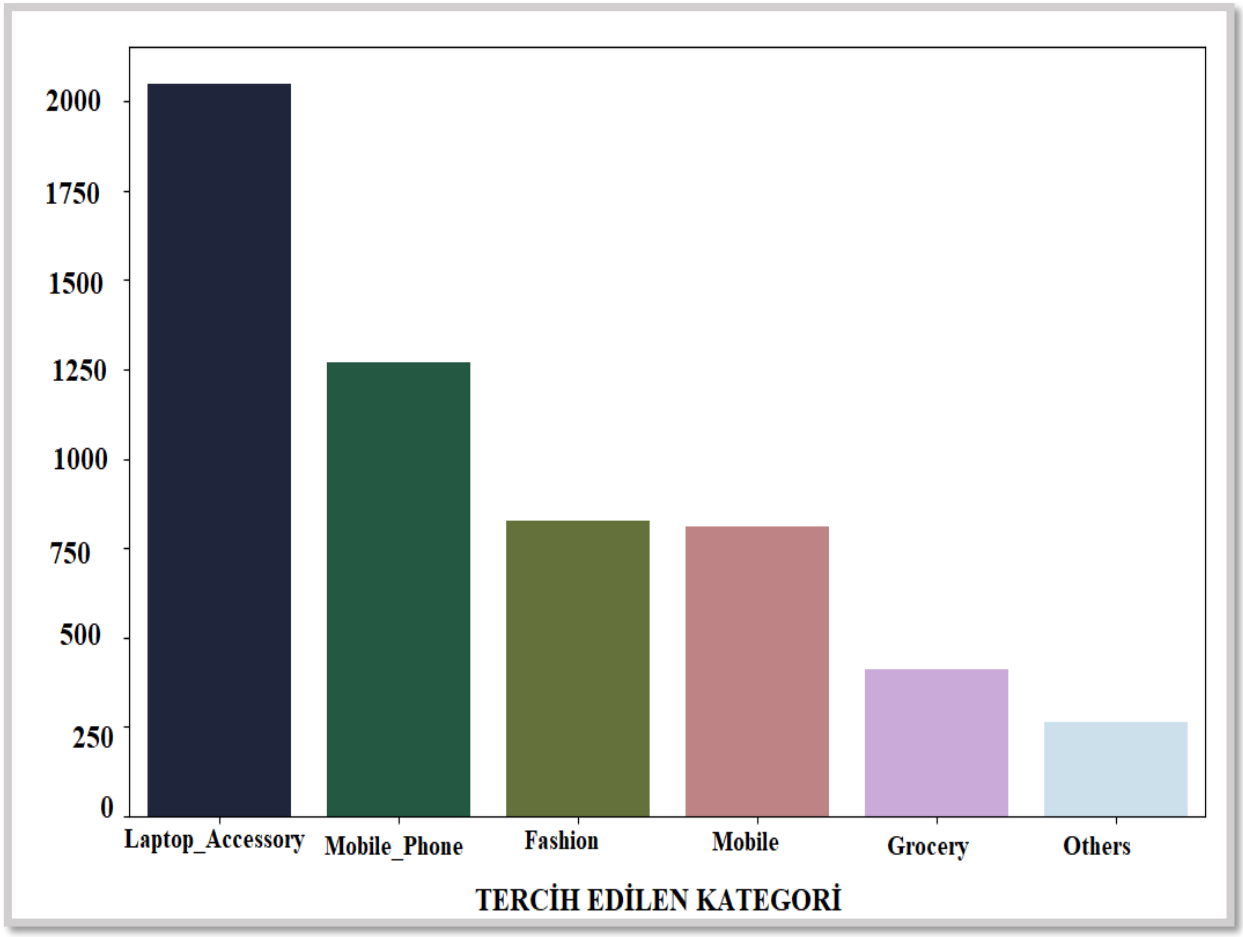
6.3.2. Değişken dönüştürme

Veri ön işleme esnasında kategorik veriler ile ilgilenilmesi gerekir. Kategorik veriler, sayısal değere sahip olmayan yani karakter özelliğine sahip verilerdir. Yapay öğrenme için modelleme yaparken bu tip veriler sıkıntıya neden olmaktadır. Nedeni yapay öğrenme, matematik ve istatistik temeli üzerine kurulmuş yöntemler bütünü olmasıdır. Sonuç olarak, kategorik verileri sayısal hale getirmek gerekir. Bu uygulamada kullanılan veri setinde kategorik özelliğe sahip öznitelikler bulunmaktadır. Veri setinde bulunan “PreferredLogindevice”, “PreferredPaymentMode”, “Gender”, “PreferredOrderCat”, “MaritalStatus” öznitelikleri kategorik özelliğe sahiptir.

6.3.2.1. Veri setinde bulanan kategorik verileri tanımlama

Kategorik özelliğe sahip öznitelikleri daha iyi anlayabilmek amacıyla Python’da “Seaborn” kütüphanesi kullanılarak her biri için grafik çizdirilmiştir. Çizdirilen grafiklerde

her bir özniteliğe ait alt değişkenler ve veri setinde ki dağılımları gösterilmiştir. Şekil 6.4' te "PreferredOrderCat" özniteliğine ait grafik verilmiştir.

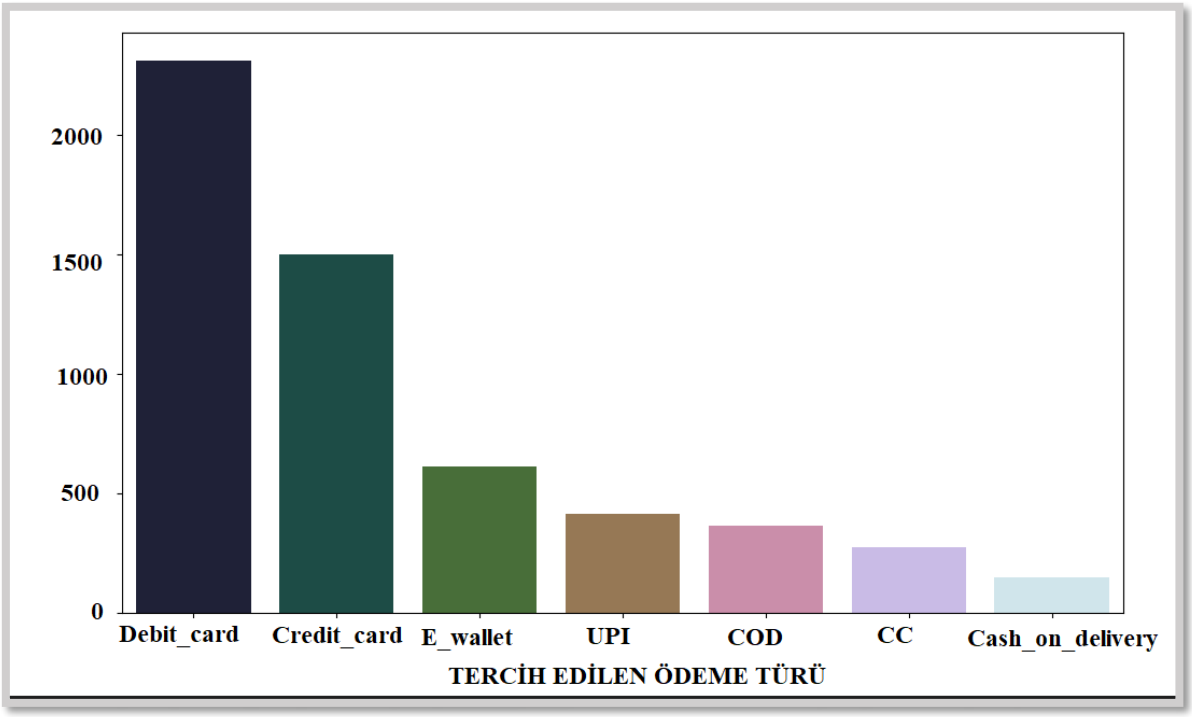


Şekil 6.4. "PreferredOrderCat" özniteliğine ait grafik

Yukarıda verilen grafiğe bakıldığında, "PreferredOrderCat" özniteliği 6 adet değişkenden oluşmaktadır.

- "Laptop_accesory" değişkeni laptop aksesuarı kategorisini ifade etmektedir.
- "Mobile_Phone" değişkeni cep telefonu kategorisini ifade etmektedir.
- "Fashion" değişkeni moda ile ilgili kategoriyi ifade etmektedir.
- "Mobile" değişkeni mobil ile ilgili kategoriyi ifade etmektedir.
- "Grocery" değişkeni market alışveriş ile ilgili kategoriyi ifade etmektedir.
- "Others" değişkeni ise kalan diğer kategorileri ifade etmektedir.

Müşterinin satın almak için çok tercih ettiği kategorinin “Laptop_Accessory” kategorisi olduğu görülmektedir. Şekil 6.5’ te “PreferredPaymentMode” özniteliğine ait grafik verilmiştir.

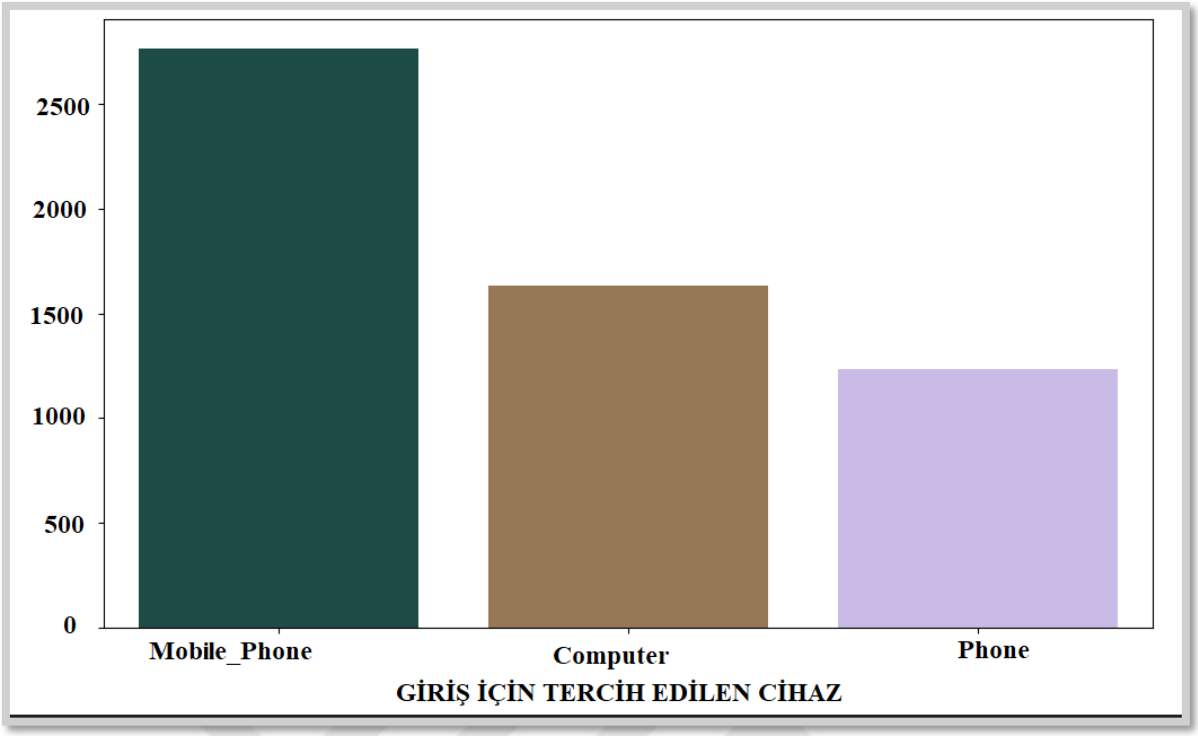


Şekil 6.5. “PreferredPaymentMode” özniteliğine ait grafik

Yukarıda verilen grafiğe bakıldığında, “PreferredPaymentMode” özniteliği 7 adet değişkenden oluşmaktadır.

- “Debit_card” değişkeni banka kartı ödeme türünü ifade etmektedir.
- “Credit_card” değişkeni kredi kartı ile ödeme türünü ifade etmektedir.
- “E-wallet” değişkeni e-cüzden ile ödeme türünü ifade etmektedir.
- “UPI”, “COD” ve “CC” değişkenleri ile ödeme diğer özel ödeme türlerini ifade etmektedir.
- “Cash_on_delivery” değişkeni teslimatta ödeme türünü ifade etmektedir.

Müşterinin çok tercih ettiği ödeme türü “Debit_Card” olduğu görülmektedir. Şekil 6.6’ da “PreferredLoginDevice” özniteliğine ait grafik verilmiştir.

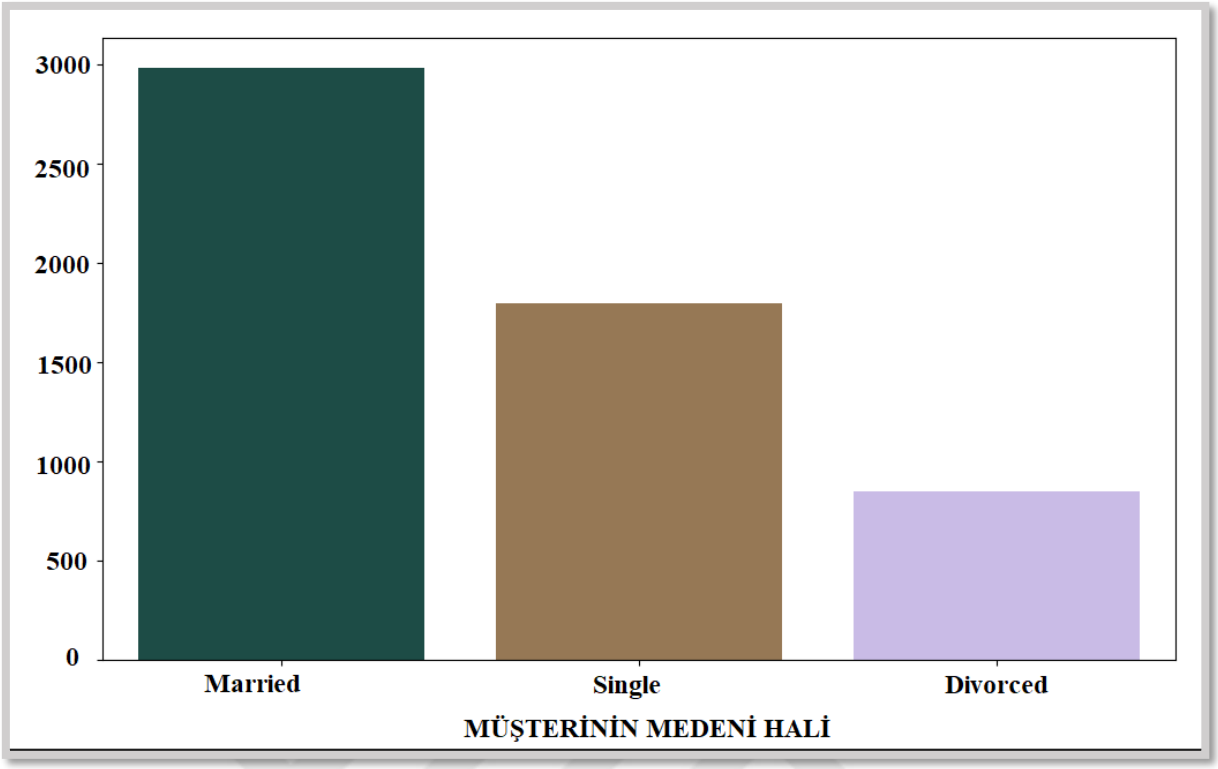


Şekil 6.6. "PreferredLoginDevice" özniteliğine ait grafik

Yukarıda verilen grafiğe bakıldığında, PreferredLoginDevice" özniteliği 3 adet değişken oluşmaktadır.

- "Mobile_Phone" değişkeni giriş için cep telefonu cihazını ifade etmektedir.
- "Computer" değişkeni giriş için bilgisayar cihazını ifade etmektedir.
- "Phone" değişkeni giriş için telefon cihazını ifade etmektedir.

Müşterinin e-ticaret platformuna giriş için en çok tercih ettiği cihazın "Mobile_Phone" olduğu görülmektedir. Şekil 6.7' de "MaritalStatus" özniteliğine ait grafik verilmiştir.



Şekil 6.7. “MaritalStatus” özniteliğine ait grafik

Yukarıda verilen grafiğe bakıldığında, “MaritalStatus” özniteliği 3 adet değişkenden oluşmaktadır.

- “Married” değişkeni evli kişileri ifade etmektedir.
- “Single” değişkeni bekar kişileri ifade etmektedir.
- “Divorced” değişkeni boşanmış kişileri ifade etmektedir.

“Married” yani evlenmiş müşterilerin daha çok e-ticaret platformunu tercih ettiği görülmüştür. Şekil 6.8’ de “Gender” özniteliğine ait grafik verilmiştir.

Şekil 6.8. "Gender" özniteliğine ait grafik

Grafikte görüldüğü gibi "Gender" özniteliği 2 adet değişkenden oluşmaktadır.

- "Male" değişkeni erkek cinsiyetine sahip kişileri ifade etmektedir.
- "Female" değişkeni kadın cinsiyetine sahip kişileri ifade etmektedir.

"Male" yani erkek müşterilerin daha çok e-ticaret sitesini tercih ettiği görülmüştür.

6.3.2.2. Veri setinde bulunan kategorik verileri dönüştürme

Veri setinde bulunan kategorik öznitelikler sayısal hale getirilmiştir. Bunun için Python'da "OneHotEncoder" sınıfı kullanılarak veri dönüşümü sağlanmıştır. Bu modül, 0 ve 1 değer ataması yapar ve 0 ve 1 olarak dönüştürülen gruplar, vektör dizeleri halinde yeni değişkenler oluştururlar. "Gender" özniteliği ikili (binary) değişkene sahiptir ve değişkenler 0 ve 1 değerine etiketlenmiştir. Diğer kategorik öznitelikler ise ikiden fazla değişkene sahiptir. Bu yüzden kukla değişken (dummy variable) ilkesi devreye girmiştir. Her bir kategorik değişken kadar kukla değişken yani kolon oluşturulur. Bu kolonlardan bir tanesi

silinir. Bu durumun nedeni çoklu bağlanım (multicollinearity) sorununun oluşmasını engellemektir. Yani k adet kukla değişken varsa (k-1) adet kolon kullanılır. Uygulama için;

- "MaritalStatus" ve "PreferredLoginDevice" öznitelikleri 3 adet değişkene sahiptir. Yani 3 adet kukla değişken oluşturulur fakat 2 (3-1) adet kolon kullanılır.
- "PreferredPaymentMode" özniteliği 7 adet değişkene sahiptir. Yani 7 adet kukla değişken oluşturulur fakat 6 (7-1) adet kolon kullanılır.
- Yine aynı şekilde, "PreferredOrderCat" özniteliği 6 adet değişkene sahiptir. Yani 6 adet kukla değişken oluşturulur fakat 5 (6-1) adet kolon kullanılır.

Müşteri hangi değere sahipse ona 1, diğer kolonlara 0 değeri verilir. Nihai olarak değişkenler kolonlara ayrılmıştır. Örneğin "MaritalStatus" özniteliği için, MaritalStatus_Married, MaritalStatus_Single, MaritalStatus_Divorced olarak 3 adet kukla değişkeni değişkeni oluşmuş fakat veri setine kukla değişken ilkesi nedeniyle 2 kolon eklenmiştir. Diğer kategorik değişkenler de aynı mantıkla kolonlara ayrılmış veri setine yeni kolon olarak eklenmiştir.

Sonuç olarak veri setinde bulunan bütün kategorik özelliğe sahip öznitelikler sayısal değere dönüştürülmüş ve bütün kategorik değişkenler ya 0 ya da 1 değerine sahip olmuştur. Veri seti dönüştürme işlemi yapmadan önce veri seti toplam 19 sütuna sahipken, dönüştürme işleminde sonra toplam 30 sütundan oluşmuştur. Veri dönüştürme işlemi sayesinde veri seti yapay öğrenme modellemesi için daha uygun hale getirilmiştir.

6.3.3. Özellik ölçeklendirme (Feature scalling)

Veri setinde bulunan sayısal değere sahip özniteliklerin farklı değerlerini en doğru şekilde analiz etmeye olanak sağlar. Özellik ölçeklendirmenin amacı farklı özniteliklerin tüm değerlerini aynı aralıkta olmasını sağlamaktır. Sayısal öznitelikler uç değerlere sahip olabilir. Bu nedenle veri setinde sayısal değere sahip öznitelikler için özellik ölçeklendirme uygulanmıştır. Bu değerlere standartlaştırma işlemi yapılmıştır. Böylelikle tüm sayısal değerler aynı aralıkta sayısal değere sahip olmuştur. Standartlaştırma işlemi Python'da "StandardScaler" sınıfı aracılığıyla uygulanmıştır. Bu Python sınıfı, Denklem 6.1'de verilen

matematiksel eşitliği temel alarak standartlaştırma işlemi yapar. Standart hale getirilecek özniteliğin her bir değerinden, ilgili özniteliğin ortalaması çıkarılır ve daha sonra standart sapmasına bölünür.

$$x' = \frac{x_i - \mu_i}{\sigma_i} \tag{6.1.}$$

x'=Standart hale getirilecek değer
x_i= İlgili özniteliğin değerleri (Girdi)
μ_i= İlgili öznitelik sütununun ortalaması
σ_i=İlgili özniteliğin standart sapması

Kategorik verilere standartlaştırma işlemi yapılmamıştır. Nedeni, kategorik verilere önceki adımda dönüştürme işlemi yapılmıştır ve tüm değerler ya 0 ya da 1 değerini almıştır. Bu durumda değerler aynı aralıkta olduğu için tekrardan standartlaştırma işlemi yapmaya gerek yoktur. Aynı şekilde bağımlı değişken olan "Churn" sütunu da ya 0 ya da 1 değerini aldığı için standartlaştırma işlemi uygulanmamıştır.

Sonuç olarak değişkenlerin bilgi ve varyansları bozulmadan, değerleri değiştirilip belli formata sokulmuştur. Böylelikle kurulacak modelin performansı arttırılmıştır.

6.3.4. Veri setini ayırma (Split the data set)

Veri setinin %70'i eğitim seti, kalan kısım yani %30'luk kısmı test seti olarak ayrılmıştır. Ayırma işleminden sonra eğitim verisi 3941 veriden oluşurken test seti 1689 veriden oluşmuştur. Eğitim seti, test setine göre daha çok veriden oluşmaktadır. Nedeni veri setindeki korelasyonu öğrenmek ve anlamak için, kurulacak olan yapay öğrenme modellerine daha fazla olanak sağlamasıdır. Ayırma işlemi Python'da "train_test_split" fonksiyonu kullanılarak yapılmıştır. Eğitim seti, makine öğrenmesi modellerini eğitmek için kullanılır. Başka bir deyişle modelin eğitilmesi için, verilerin tanınması ve tahminlemenin yapılması bu eğitim verileri üzerinden gerçekleşmektedir. Test seti ise eğitim seti kullanılarak oluşturulan modelin doğru çalışıp çalışmadığını test etmek için kullanılmıştır. Yeni gözlemler üzerinde modelin performansının ölçüldüğü veri setidir.

6.4. Veri Setine SMOTE Yöntemi Uygulama

Veri seti ayrılmadan önce veri setinde 948 müşteri kayıp (Churn), 4682 müşteri ise kayıp değildir. Başka bir deyişle 948 müşteri 1 sınıfına aitken, 4682 müşteri ise 0 sınıfına ait olduğu görülmüştür. Kaybedilen müşterilerin sayısının, mevcut müşterileri büyük ölçüde aştığı tespit edilmiştir. Veri setinde 1 sınıfı azınlık durumdayken, 0 sınıfı çoğunluk durumundadır. Bu yüzden, veri setinde dengesiz (imbalance) bir durum olduğu görülmüştür. Bu durumda yapay öğrenme modeli sınıflandırma yaparken azınlık olan sınıfı dikkate almadan çoğunluğa göre bir sınıflandırma yapabilir. Böylece yapay öğrenme algoritmaları tahminleme yaparken çoğunluk kümesini tercih eder ve doğru sonuçlar vermez.

Veri seti ön işleme yapıldıktan ve veri seti eğitim ve test seti olarak ayrıldıktan sonra modeli eğitmek için eğitim seti kullanılmıştır. Eğitim setinde 3267 veri 0 sınıfına aitken 674 veri 1 sınıfına ait olduğu görülmüştür. Bu sınıf dengesizliğini önlemek amacıyla Python'da "SMOTE" sınıfı kullanılarak SMOTE yöntemi uygulanmıştır. Bu Python sınıfı, azınlık sınıf olan 1 sınıfından sentetik örnekler oluşturmuştur. 1 sınıfında bulunan verileri aşırı örneklemiştir. Modeli eğitmeden önce 1 sınıfı verilerini çoğaltmıştır. Bunu en yakın komşularını kullanarak yapmıştır. Böylelikle sınıf dağılımı dengelenmiştir. Çizelge 6.2' de eğitim veri setinin SMOTE uygulamadan önce ve sonraki hali gösterilmiştir.

Çizelge 6.2. SMOTE yöntemi uygulama öncesi ve sonrası

	Ayrılmayan Müşteri (0 sınıfı)	Ayrılan Müşteri (1 sınıfı)
SMOTE uygulama öncesi	3267	674
SMOTE uygulama sonrası	3267	3267

Çizelge 6.2' de görüldüğü üzere, SMOTE yöntemi uygulaması sonrası azınlık sınıfı olan 1 sınıfı için sentetik olarak veriler oluşturulmuştur. Modeli eğitmek için eğitim seti bu şekilde kullanılmıştır. Böylelikle yanlış sınıflandırmanın ve aşırı öğrenmenin (overfitting) önüne geçilmiştir.

6.5. Modelleme

E-ticaret sektöründeki müşteri kaybı tahmini yapmak ve model geliştirmek amacıyla yapay öğrenmeye ait sınıflandırma algoritmaları kullanılmıştır. Bu algoritmalar:

- Lojistik Regresyon
- K-En Yakın Komşu
- Destek Vektör Makine
- Karar Ağacı
- Rastgele Orman
- XGBoost
- LightGBM

Performans kriteri olarak hem literatürde hem de gerçek yaşam uygulamalarında kullanılan performans metrikleri kullanılmıştır. Bu çalışma için tahmin modellerini kıyaslamak amacıyla doğruluk (accuracy) ve AUC değerine göre değerlendirme yapılmıştır.

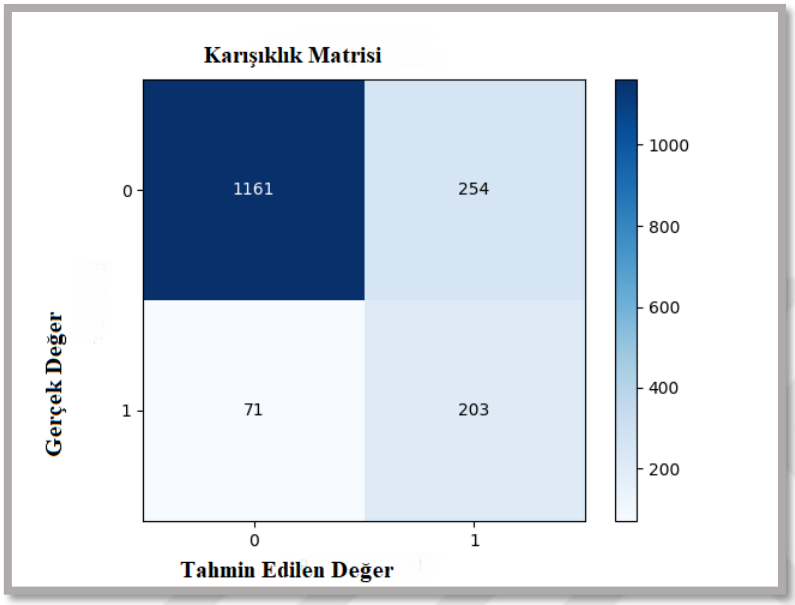
6.5.1. Lojistik Regresyon modelleme

Uygulama için ilk oluşturulan yapay öğrenme modelidir. Lojistik Regresyon modeli kurulmadan önce hiper parametre ayarlaması yapılmıştır. Her ne kadar modelin önceden belirlenmiş parametre değeri mevcut ise de daha iyi bir model elde etmek amacıyla bu ayarlama işlemi yapılmıştır. Model için, maksimum iterasyon sayısını ayarlayan "max_iter" parametresi belirlenmiştir. GridSearchCV yöntemi uygulanarak model için en iyi parametre buldurulmuştur. Bu yöntemi uygulamak için Python'da "GridSearchCV" sınıfı kullanılmıştır. Bu uygulamada "max_iter" parametresi için şu değerlerin deneme kümesi oluşturulmuştur: {100, 1000,10000,100000}. En iyi sonucu veren parametre değeri 100000 değeridir.

- Lojistik Regresyon hiper parametre ayarından sonra (max_iter = '100000') olarak ayarlanmıştır.

Hiper parametre ayarlaması yapıldıktan sonra Lojistik Regresyon modeli kurulmuştur. Model önce eğitilmiş daha sonra test edilmiştir.

Modeli değerlendirmek için performans kriteri olarak doğruluk (accuracy) ve AUC değerine göre değerlendirme yapılmıştır. Öncelikle Python'da "confusion_matrix" sınıfı kullanılarak karışıklık matrisi çizdirilmiştir. Şekil 6.9 'da Lojistik Regresyon modeline ait karışıklık matrisi gösterilmiştir.



Şekil 6.9. Lojistik Regresyon modeli için karışıklık matrisi

Karışıklık matrisine göre, müşteri ayrılmışsa (Churn) 1 sınıf değerini alır ve bu durum doğru kabul edilirken, ayrılmamışsa 0 sınıf değerini alır ve bu durum yanlış kabul edilir.

- Veri setinin gerçek sınıfının doğru (sınıf değeri 1) ve tahmin edilenin de doğru olduğu (sınıf değeri 1) 203 müşteri görülmüştür. 203 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmış müşterilerdir. Bu durumda doğru pozitif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf değeri 0) ve tahmin edilenin de yanlış olduğu (sınıf değeri 0) 1161 müşteri görülmüştür. 1161 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmamış müşterilerdir. Bu durumda doğru negatif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf değeri 0) ve tahmin edilenin ise doğru olduğu (sınıf değeri 1) 254 müşteri görülmüştür. 254 müşteri gerçek

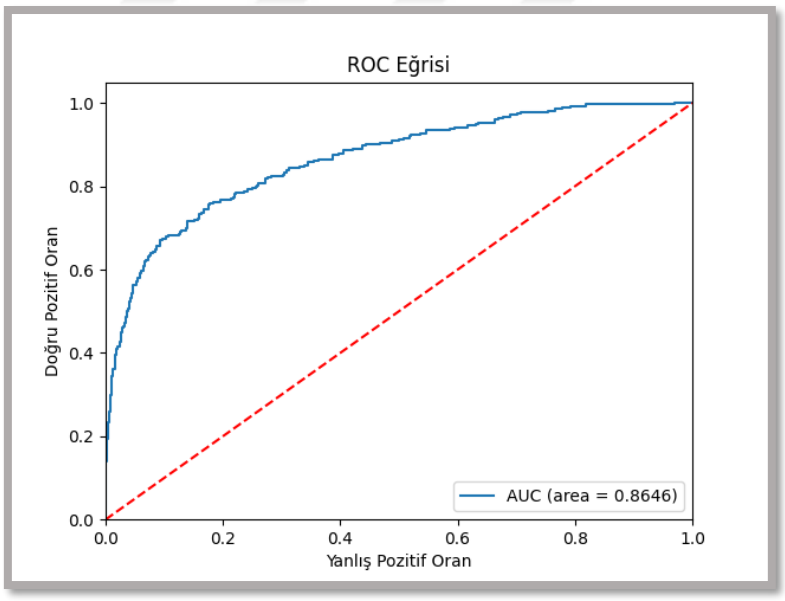
veri setinde ayrılmamış fakat tahmin veri setinde ayrılmış müşterilerdir. Bu durumda yanlış pozitif olur.

- Veri setinin gerçek sınıfının doğru (sınıf değeri 1) ve tahmin edilenin de yanlış olduğu (sınıf değeri 0) 71 müşteri görülmüştür. 71 müşteri gerçek veri setinde ayrılmış fakat tahmin veri setinde ayrılmamış müşteridir. Bu durumda yanlış negatif olur.

Karışıklık matrisi kullanılarak Denklem 6.2' de doğruluk değeri hesaplanmıştır.

$$\text{Doğruluk (Accuracy)} = \frac{203 + 1161}{203 + 1161 + 254 + 71} = 0.8076 = \%80{,}76 \qquad (6.2)$$

Çıkan sonuca göre Lojistik Regresyon modeli %80,76 doğruluk göstermiştir. Tatmin edici bir doğruluk yüzdesi görülmüştür. Ardından Python'da "Matplotlib" kütüphanesi kullanılarak ROC eğrisi çizdirilmiş ve bu eğrinin altındaki alanı veren AUC değeri hesaplatılmıştır. Şekil 6.10' da ROC-AUC eğrisi verilmiştir.



Şekil 6.10. Lojistik Regresyon modeli için ROC-AUC eğrisi

Grafiğe bakıldığında iki sınıfın (sınıf = 0, sınıf = 1) doğruluk oranlarının yüksek olduğu görülmektedir. Aynı zamanda sınıfların geçtiği eğrinin altında yer alan bölüm AUC

bölgesi yüksek yer kapladığı için AUC değeri 0.8646'dır. AUC değeri 1'e yaklaştığı için model başarısının tatmin edici olduğu söylenebilir.

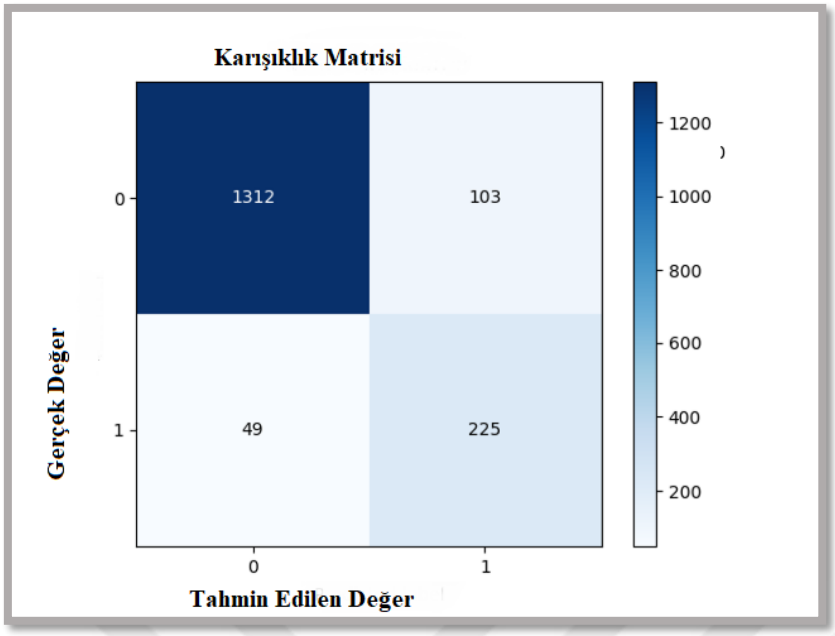
6.5.2. Destek Vektör Makine (DVM) modelleme

Uygulama için kurulan diğer yapay öğrenme modelidir. DVM modeli kurulmadan önce hiper parametre ayarlaması yapılmıştır. Yine modelin başta varsaydığı parametre değerleri vardır fakat en iyi modeli kurmak için bu ayarlama işlemi yapılmıştır. Model için, önemli bir parametre olan ve ele alınan problem için kullanılan veri seti türüne göre değişiklik gösteren, çekirdek türü ve sayısı ayarlaması yapan "kernel" parametresi ayarlanmıştır. GridSearchCV yöntemi uygulanarak model için en iyi parametre buldurulmuştur. Bu yöntemi uygulamak için Python'da "GridSearchCV" sınıfı kullanılmıştır. Bu uygulamada "kernel" parametresi için şu değerlerin denemesi yapılmıştır: {linear, rbf}. En iyi sonucu veren parametre rbf çekirdek türüdür. En iyi sonucun rbf çekirdek (kernel) türünün olması, veri setinin doğrusal olmayan bir yapıda olduğunu göstermiştir.

- DVM için hiper parametre ayarlanmış hali (kernel = 'rbf') şeklindedir.

Hiper parametre ayarlaması yapıldıktan sonra DVM modeli kurulmuştur. Model önce eğitilmiş daha sonra test edilmiştir.

Modeli değerlendirmek için performans kriteri olarak doğruluk (accuracy) ve AUC değerine göre değerlendirme yapılmıştır. İlk olarak Python' da karışıklık matrisi çizdirilmiştir. Şekil 6.11' de DVM modeline ait karışıklık matrisi gösterilmiştir.



Şekil 6.11. DVM modeli için karışıklık matrisi

Karışıklık matrisine göre, müşteri ayrılmışsa (Churn) 1 sınıf değerini alırken ve bu durum doğru kabul edilirken, ayrılmamışsa 0 sınıf değerini alırken ve bu durum yanlış kabul edilirdi.

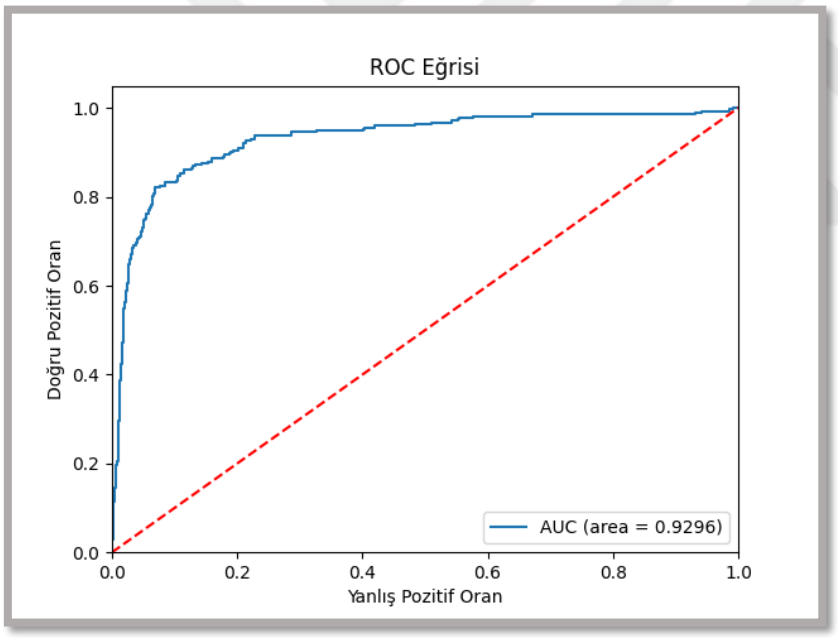
- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin de doğru olduğu (sınıf = 1) 225 müşteri görülmüştür. 225 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmış müşterilerdir. Bu durumda doğru pozitif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin ise yanlış olduğu (sınıf = 0) 1312 müşteri görülmüştür. 1312 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmamış müşterilerdir. Bu durumda doğru negatif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin de doğru olduğu (sınıf = 1) 103 müşteri görülmüştür. 103 müşteri gerçek veri setinde ayrılmamış fakat tahmin veri setinde ayrılmış müşterilerdir. Bu durumda yanlış pozitif olur.
- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin ise yanlış olduğu (sınıf = 0) 49 müşteri görülmüştür. 49 müşteri gerçek veri setinde

ayrılmış fakat tahmin veri setinde ayrılmamış müşteridir. Bu durumda yanlış negatif olur.

Karışıklık matrisi kullanılarak DVM modeli için doğruluk değeri hesaplanmıştır. Denklem 6.3' te hesaplaması verilmiştir.

$$\text{Doğruluk (Accuracy)} = \frac{225 + 1312}{225 + 1312 + 103 + 49} = 0.9100 = \%91{,}00 \qquad (6.3)$$

Çıkan sonuca göre DVM modeli %91,00 doğruluk göstermiştir. Model kayıp müşteri tahmininde başarılı bir sonuç göstermiştir. ROC eğrisi çizdirilmiş ve bu eğrinin altındaki alanı veren AUC değeri hesaplatılmıştır. Şekil 6.12' de ROC-AUC eğrisi verilmiştir.



Şekil 6.12. DVM modeli için ROC-AUC eğrisi

Grafikte görüldüğü gibi başarılı bir sonuç vermiştir. ROC eğrisinin altında yer alan bölüm AUC bölgesi yüksek yer kapladığı için, AUC değerinin 0.9296 olduğu görülmektedir. AUC değeri 1'e yaklaştığı için modelin başarılı olduğu söylenebilir.

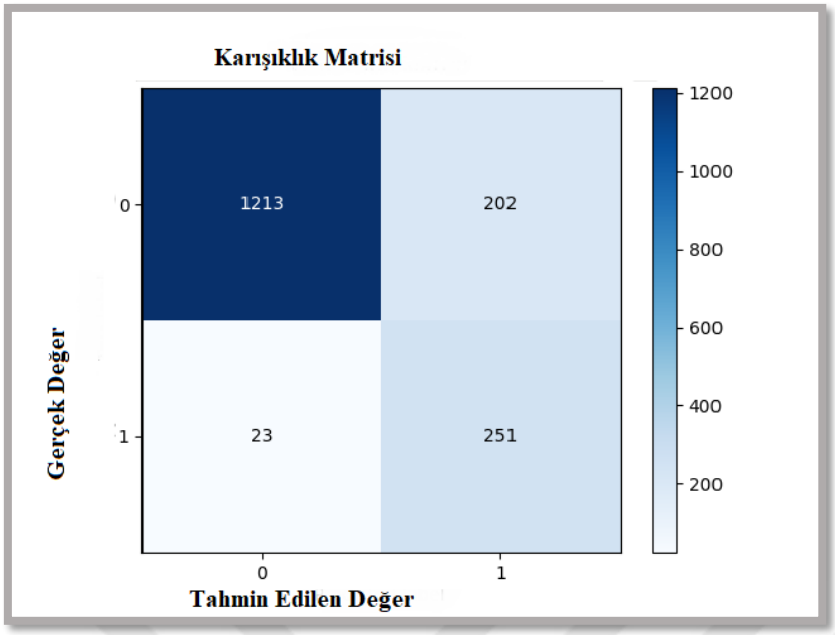
6.5.3. K-En Yakın Komşu (K-NN) modelleme

K-NN, bu uygulama için kullanılan diğer sınıflandırma algoritmasıdır. K-NN modeli kurulmadan önce hiper parametre ayarlanmıştır. Modelin başta varsaydığı parametre değerleri vardır ancak bazı parametrelerin ayarlaması yapılmasına ihtiyaç duyulmuştur. K-NN modelinin çalışma prensibi için önemli bir parametre olan en yakın komşu sayısını ayarlayan "n_neighbors" parametresi ayarlanmıştır. GridSearchCV yöntemi uygulanarak model için en iyi parametre buldurulmuştur. Bu uygulamada "n_neighbors" parametresi için şu aralıkta bulunan değerlerin denemesi yapılmıştır: (1,50). Değerlerin aralığı arttırılırsa model iyi sonuç vermeyeceği için 1 ile 50 arasında en iyi parametre değeri aranmıştır. GridSearchCV fonksiyonu 10 kat çapraz doğrulama (cv=10) ile parametrenin en iyi değerini aramıştır. En iyi sonucu veren parametre değeri 5'dir. Yani sınıflandırma işlemini, en yakınında bulunan 5 komşusuna göre yapmıştır. Ardından değerler arasındaki uzaklığı hesaplamıştır. En sık kullanılan uzaklık metriği "Minkowski Uzaklık" fonksiyonu tercih edilmiştir.

- K-NN hiper parametre ayarlanmış (n_neighbors = '5', metric= 'Minkowski') halidir.

Hiper parametre ayarlaması yapıldıktan sonra K-NN modeli kurulmuştur. Model önce eğitilmiş daha sonra test edilmiştir.

Modeli değerlendirmek için performans kriteri olarak doğruluk (accuracy) ve AUC değerine göre değerlendirme yapılmıştır. İlk olarak Python'da karışıklık matrisi çizdirilmiştir. Şekil 6.13' te K-NN modeline ait karışıklık matrisi gösterilmiştir.



Şekil 6.13. K-NN modeli için karışıklık matrisi

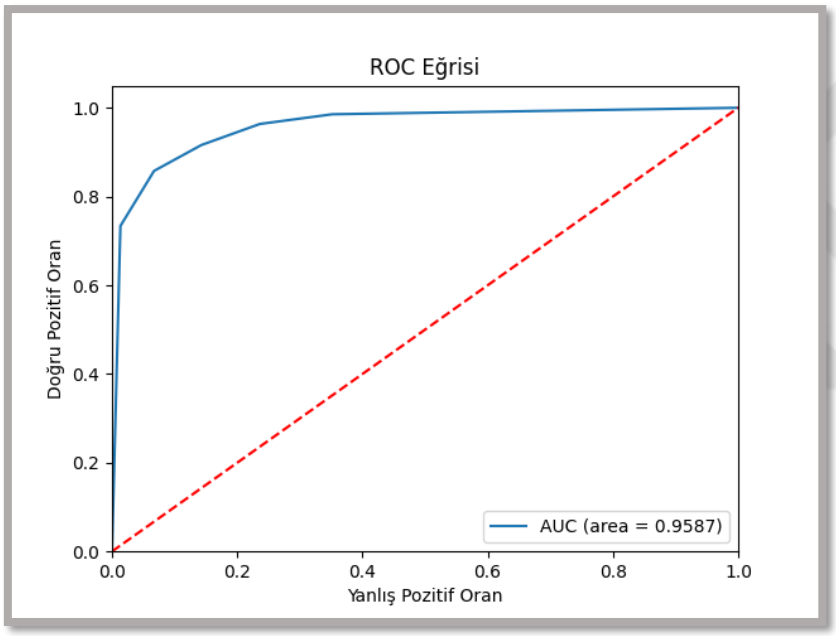
Karışıklık matrisine göre;

- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin de doğru olduğu (sınıf = 1) 251 müşteri görülmüştür. 251 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmış müşterilerdir. Bu durumda doğru pozitif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin de yanlış olduğu (sınıf = 0) 1213 müşteri görülmüştür. 1213 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmamış müşterilerdir. Bu durumda doğru negatif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin ise doğru olduğu (sınıf = 1) 202 müşteri görülmüştür. 202 müşteri gerçek veri setinde ayrılmamış fakat tahmin veri setinde ayrılmış müşterilerdir. Bu durumda yanlış pozitif olur.
- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin ise yanlış olduğu (sınıf = 0) 23 müşteri görülmüştür. 23 müşteri gerçek veri setinde ayrılmış fakat tahmin veri setinde ayrılmamış müşteridir. Bu durumda yanlış negatif olur.

Karışıklık matrisi kullanılarak K-NN modeli için doğruluk değeri hesaplanmıştır. Doğruluk değeri Denklem 6.4' te verilmiştir.

$$\text{Doğruluk (Accuracy)} = \frac{251 + 1213}{251 + 1213 + 202 + 23} = 0.8667 = \%86{,}67 \qquad (6.4)$$

Çıkan sonuca göre K-NN modeli %86,67 doğruluk göstermiştir. ROC eğrisi çizdirilmiş ve bu eğrinin altındaki alanı veren AUC değeri hesaplatılmıştır. Şekil 6.14' te ROC-AUC eğrisi verilmiştir.



Şekil 6.14. K-NN modeli için ROC-AUC eğrisi

Grafikte görüldüğü gibi diğer kurulan üç modele göre başarılı bir sonuç vermiştir. ROC eğrisinin altında yer alan bölüm AUC bölgesi yüksek yer kaplamıştır. AUC, 0.9587 değerine sahiptir. AUC değeri 1'e oldukça yaklaşmıştır.

6.5.4. Karar Ağaçları modelleme

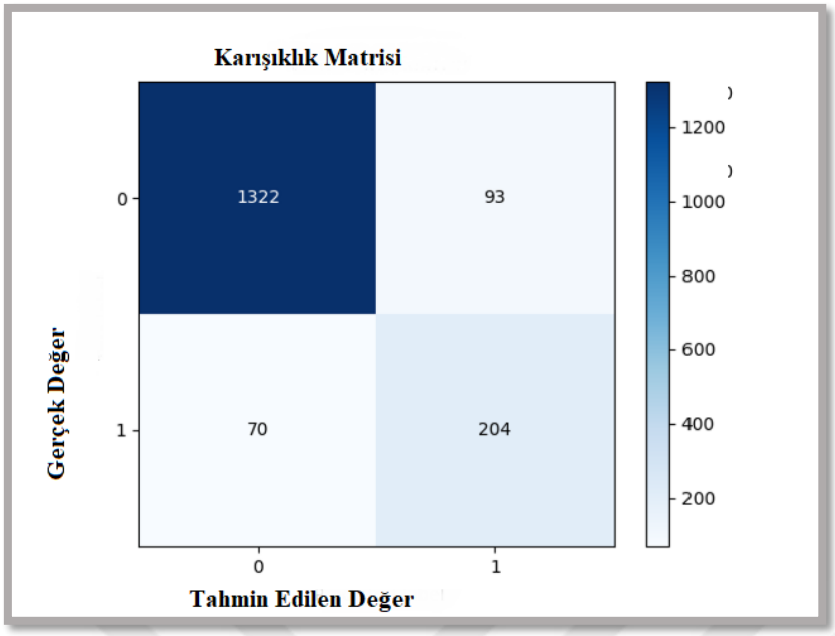
Ağaç yapısı şeklinde sınıflandırma yapan Karar Ağaçları modeli kurulmuştur. Model kurulmadan önce diğer modellerde olduğu gibi hiper parametre ayarı yapılmıştır. Karar Ağaçları modelinin çalışma prensibi için önemli bir parametre olan "max_depth"

parametresi ayarlanmıştır. Bu parametre, kök düğümden yaprak düğüme giden yol uzunluğunu ifade etmekte ve bir ağacın ne kadar derinleşeceğine karar vermektedir. GridSearchCV yöntemi uygulanarak model için en iyi parametre buldurulmuştur. Bu uygulamada "max_depth" parametresi için şu aralıkta bulunan değerlerin denemesi yapılmıştır: (2,5,10,18). Değerlerin aralığı model için kullanılan veri setine göre verilmiştir. Eğer çok büyük değerler kullanılırsa ağaç o kadar fazla dallandığı için veri setini iyi öğrenir fakat genelleme özelliğini kaybeder. Bu parametre sayesinde dallanma kontrol altına alınmış olur. GridSearchCV fonksiyonu 10 kat çapraz doğrulama (cv=10) ile parametrenin en iyi değerini aramıştır. Bu fonksiyon parametre için deneme yapılan değer sayısı ve belirlenen çapraz doğrulama sayısının çarpımı kadar modeli dener ve en iyi sonucu buldurur. Yani 4*10 yaparak 40 modeli denemiş ve en iyi sonucu bulmuştur. En iyi sonucu veren parametre değeri 10'dur. Ağaç yapısının homojen olması istendiği için homojenliği hesaplamak için entropi kullanılmıştır. Diğer parametre değerlerine ayarlama yapılmamıştır. Nedeni en iyi değeri varsayılan değerler vermesidir.

- Karar ağacı için hiper parametre ayarından sonra (max_depth= '10', criterion= 'entropy') değerleri kullanılmıştır.

Hiper parametre ayarlaması yapıldıktan sonra Karar Ağaçları modeli kurulmuştur. Model önce eğitilmiş daha sonra test edilmiştir.

Modeli değerlendirmek için performans kriteri olarak doğruluk (accuracy) ve AUC değerine göre değerlendirme yapılmıştır. İlk olarak karışıklık matrisi Python'da çizdirilmiştir. Şekil 6.15'te Karar Ağaçları modeline ait karışıklık matrisi gösterilmiştir.



Şekil 6.15. Karar Ağaçları model için karışıklık matrisi

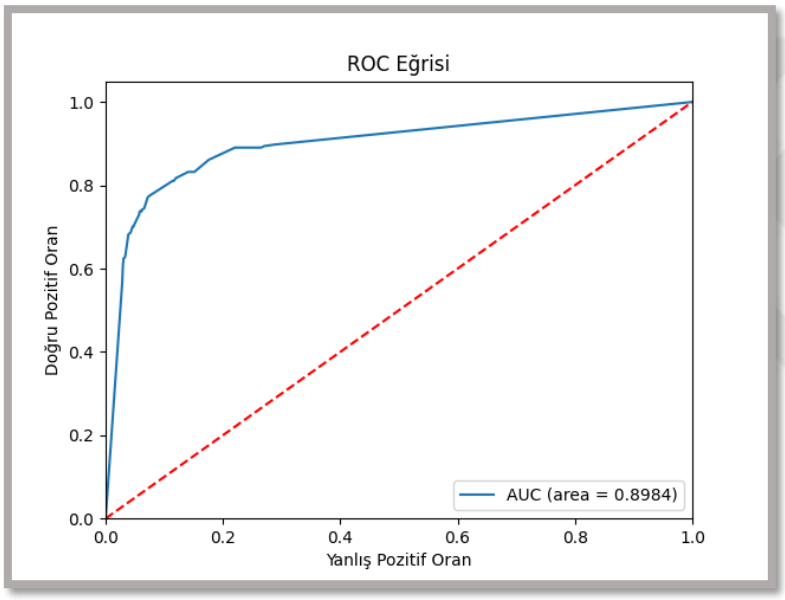
Karışıklık matrisine göre;

- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin de doğru olduğu (sınıf = 1) 204 müşteri görülmüştür. 204 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmış müşterilerdir. Bu durumda doğru pozitif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin ise yanlış olduğu (sınıf = 0) 1322 müşteri görülmüştür. 1322 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmamış müşterilerdir. Bu durumda doğru negatif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin ise doğru olduğu (sınıf = 1) 93 müşteri görülmüştür. 93 müşteri gerçek veri setinde ayrılmamış fakat tahmin veri setinde ayrılmış müşterilerdir. Bu durumda yanlış pozitif olur.
- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin de yanlış olduğu (sınıf = 0) 70 müşteri görülmüştür. 70 müşteri gerçek veri setinde ayrılmış fakat tahmin veri setinde ayrılmamış müşteridir. Bu durumda yanlış negatif olur.

Karışıklık matrisi kullanılarak Karar Ağaçları modeli için doğruluk değeri hesaplanmıştır. Denklem 6.5' te hesaplanmıştır.

$$\text{Doğruluk (Accuracy)} = \frac{204 + 1322}{251 + 1213 + 202 + 23} = 0.9035 = \%90{,}35 \qquad (6.5)$$

Çıkan sonuca göre Karar Ağaçları modeli %90,35 doğruluk göstermiştir. ROC eğrisi çizdirilmiş ve bu eğrinin altındaki alanı veren AUC değeri hesaplatılmıştır. Şekil 6.16'da ROC-AUC eğrisi verilmiştir.



Şekil 6.16. Karar Ağaçları modeli için ROC-AUC eğrisi

Grafiğe bakıldığında, AUC değeri 0.8984 olarak hesaplanmıştır. AUC değeri 1'e yaklaşmıştır.

6.5.5. Rastgele Orman modelleme

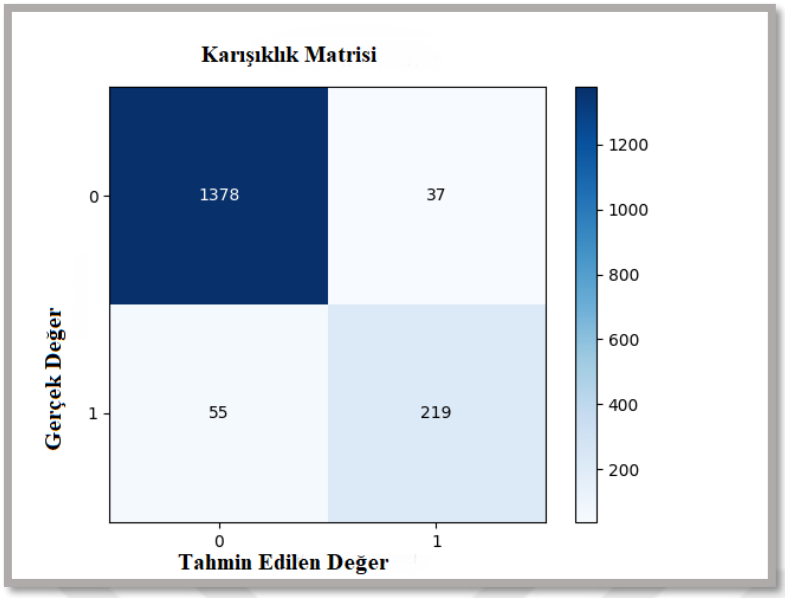
Yapay öğrenme için en yaygın kullanılan algoritmadır. Bu algoritma ile Rastgele Orman modeli kurulmuştur. Modeli kurmadan önce hiper parametre ayarı yapılmıştır. Rastgele Orman modelinin çalışma prensibi için önemli bir parametre olan birkaç parametrenin ayarlaması yapılmıştır. İlk olarak kullanılacak ağaç sayısını veren "n_estimator" parametresinin ayarlaması için şu değerler denenmiştir: (100,200,500,1000).

Ardından bölünme bulunurken göz önünde bulundurulması gereken maksimum değişken sayısını veren “max_feaures” parametresinin ayarlaması için şu değerler denenmiştir: (3,5,7,8,10,18). Diğer parametre ise dallanmayı kontrol eden “min_samples_split” parametresinin ayarlaması için şu değerler denenmiştir: (2,5,10,18). GridSearchCV yöntemi uygulanarak model için en iyi parametre buldurulmuştur. Değerlerin aralığı model için kullanılan veri setine göre verilmiştir. Aşırı öğrenmemin önüne geçmek için uygun değerler denenmiştir. GridSearchCV fonksiyonu 10 kat çapraz doğrulama (cv=10) ile parametrenin en iyi değerini aramıştır. Bu fonksiyon parametre için deneme yapılan değer sayısı ve belirlenen çapraz doğrulama sayısının çarpımı kadar modeli dener ve en iyi sonucu buldurur. Yani 4*6*4*10 yaparak 960 modeli denemiş ve en iyi sonucu bulmuştur. Tüm değerlerin kombinasyonunu deneyerek Rastgele Orman modeli en iyi sonuçları veren parametre değerlerini bulmuştur. Deneme sonucunda “n_estimator” parametresi için en iyi parametre değeri 1000, “max_feaures” için en iyi parametre değeri 18, “min_samples_split” parametresi için en iyi değer 2’dir. Diğer parametre değerlerine ayarlama yapılmamıştır. Diğer parametrelerin varsayılan değerleri modele daha iyi katkı sağladığı için değiştirilmemiştir.

- Rastgele Orman için (max_features=‘18’, n_estimator=‘1000’, min_samples_split= ‘2’) hiper parametre değerleri kullanılmıştır.

Hiper parametre ayarlaması yapıldıktan sonra Rastgele Orman modeli kurulmuştur. Model önce eğitilmiş daha sonra test edilmiştir.

Karışıklık matrisi Python’da çizdirilmiştir. Şekil 6.17’de Rastgele Orman modeline ait karışıklık matrisi gösterilmiştir.



Şekil 6.17. Rastgele Orman modeli için karışıklık matrisi

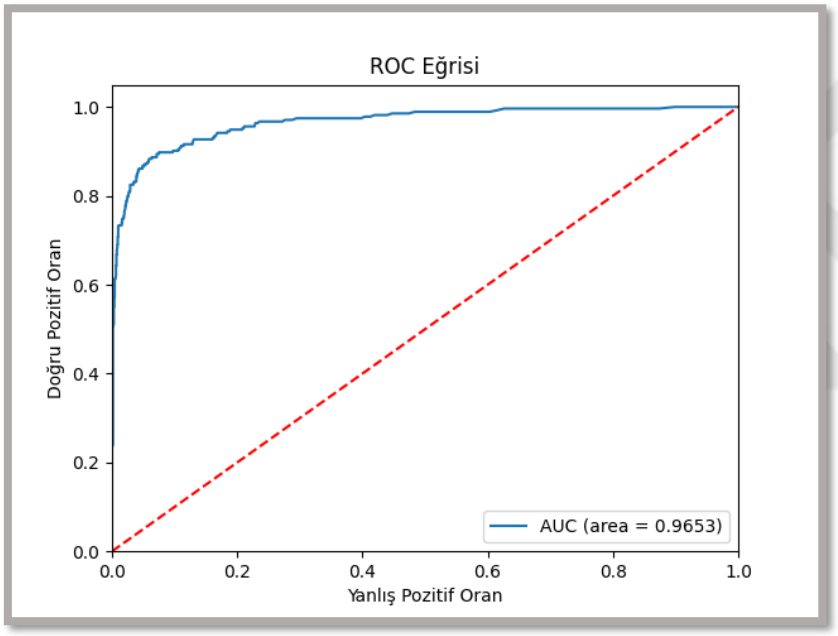
Karışıklık matrisine göre;

- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin de doğru olduğu (sınıf = 1) 219 müşteri görülmüştür. 219 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmış müşterilerdir. Bu durumda doğru pozitif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin de yanlış olduğu (sınıf = 0) 1378 müşteri görülmüştür. 1378 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmamış müşterilerdir. Bu durumda doğru negatif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin ise doğru olduğu (sınıf = 1) 37 müşteri görülmüştür. 37 müşteri gerçek veri setinde ayrılmamış fakat tahmin veri setinde ayrılmış müşterilerdir. Bu durumda yanlış pozitif olur.
- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin ise yanlış olduğu (sınıf = 0) 55 müşteri görülmüştür. 55 müşteri gerçek veri setinde ayrılmış fakat tahmin veri setinde ayrılmamış müşteridir. Bu durumda yanlış negatif olur.

Karışıklık matrisi kullanılarak Karar Ağaçları modeli için doğruluk değeri hesaplanmıştır. Denklem 6.6'da gösterilmiştir.

$$\text{Doğruluk (Accuracy)} = \frac{219 + 1378}{219 + 1378 + 37 + 55} = 0.9455 = \%94{,}55 \qquad (6.6)$$

Çıkan sonuca göre Rastgele Orman modeli %94,55 doğruluk göstermiştir. ROC eğrisi çizdirilmiş ve bu eğrinin altındaki alanı veren AUC değeri hesaplatılmıştır. Şekil 6.18' de ROC-AUC eğrisi verilmiştir.



Şekil 6.18. Rastgele Orman modeli için ROC-AUC eğrisi

Grafiğe bakıldığında, AUC değeri 0.9653 olarak hesaplanmıştır. AUC değeri 1'e oldukça yaklaşmıştır. Bu durum model için olumlu bir sonuçtur.

6.5.6. XGBoost modelleme

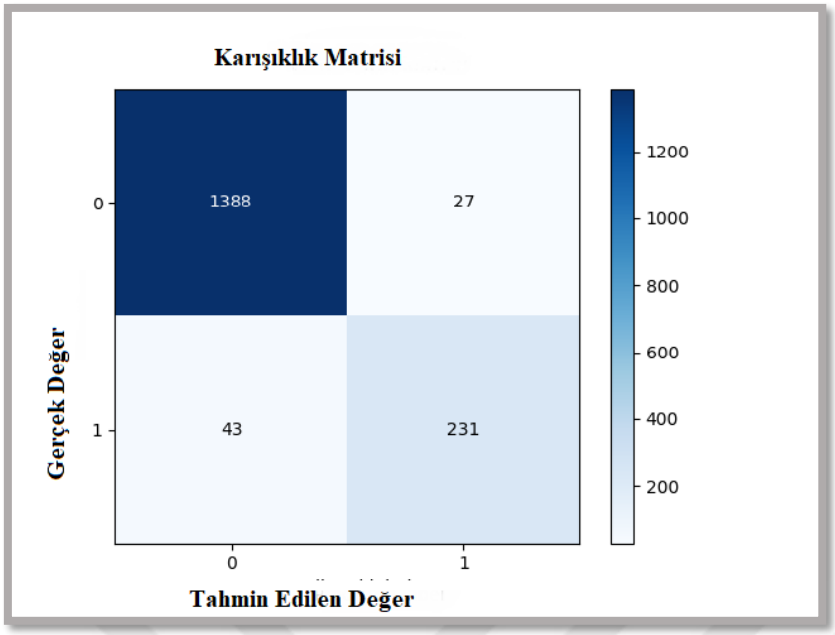
Günümüzde yaygın olarak kullanılan güncel bir algoritma olan XGBoost algoritması kullanılarak modelleme yapılmıştır. Model kurulmadan hiper parametre ayarı yapılmıştır. XGBoost modelinin birkaç parametresinin ayarlaması yapılmıştır. İlk olarak kullanılacak ağaç sayısını veren "n_estimator" parametresinin ayarlaması için şu değerler denenmiştir:

(100,200,500,1000). Ardından dallanmayı kontrol eden "max_depth" parametresinin ayarlaması için şu değerler denenmiştir: (3,5,7,8,10,18). Veri setinden rastgele olarak seçilen alt örneklem oranı olan "subsample" parametresinin ayarlaması için şu değerler denenmiştir: (0.1,0.2,0.3,0.4,0.5,0.6.0.7.0.8,1). Bu denenen değerler 0 ile 1 arasında olmalıdır. Modele yeni eklenen ağaçların ağırlığını kontrol eden "learning_rate" parametresinin ayarlaması için şu değerler denenmiştir: (0.1,0.01,0.001). GridSearchCV yöntemi uygulanarak model için en iyi parametre buldurulmuştur. Değerlerin aralığı model için kullanılan veri setine göre verilmiştir. Aşırı öğrenmemin önüne geçmek için uygun değerler denenmiştir. GridSearchCV fonksiyonu 10 kat çapraz doğrulama (cv=10) ile parametrenin en iyi değerini aramıştır. Bu fonksiyon parametre için deneme yapılan değer sayısı ve belirlenen çapraz doğrulama sayısının çarpımı kadar modeli dener ve en iyi sonucu buldurur. Yani 4*6*9*3*10 yaparak 6480 modeli denemiş ve en iyi sonucu bulmuştur. Denenecek model sayısı arttığı için hiper parametre ayar süresi uzamıştır. Tüm değerlerin kombinasyonunu deneyerek XGBoost modeli en iyi sonuçları veren parametre değerlerini bulmuştur. Deneme sonucunda "n_estimator" parametresi için en iyi parametre değeri 500, "max_features" için en iyi parametre değeri 18, "subsample" parametresi için en iyi değer 0.8 ve "learning_rate" parametresi için en iyi değer 0.1'dir. Diğer parametre değerlerinde değişiklik yapılmamıştır ve varsayılan değerler kullanılmıştır.

- XGBoost için (n_estimator= '500', learning_rate= '0,1', max_features= '18', subsample= '0,8') hiper parametre değerleri kullanılmıştır.

Hiper parametre ayarlaması yapıldıktan sonra XGBoost modeli kurulmuştur. Model önce eğitilmiş daha sonra test edilmiştir.

Modeli değerlendirmek için performans kriteri olarak doğruluk (accuracy) ve AUC değerine göre değerlendirme yapılmıştır. İlk olarak karışıklık matrisi Python' da çizdirilmiştir. Şekil 6.19' da XGBoost modeline ait karışıklık matrisi gösterilmiştir.



Şekil 6.19. XGBoost model için karışıklık matrisi

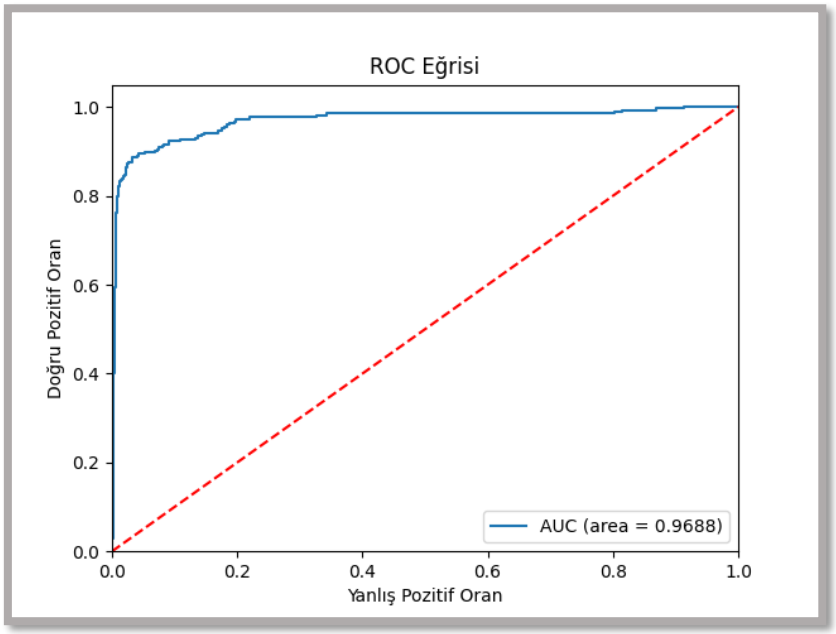
Karışıklık matrisine göre;

- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin de doğru olduğu (sınıf = 1) 231 müşteri görülmüştür. 231 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmış müşterilerdir. Bu durumda doğru pozitif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin de yanlış olduğu (sınıf = 0) 1388 müşteri görülmüştür. 1388 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmamış müşterilerdir. Bu durumda doğru negatif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin ise doğru olduğu (sınıf = 1) 27 müşteri görülmüştür. 27 müşteri gerçek veri setinde ayrılmamış fakat tahmin veri setinde ayrılmış müşterilerdir. Bu durumda yanlış pozitif olur.
- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin ise yanlış olduğu (sınıf = 0) 43 müşteri görülmüştür. 43 müşteri gerçek veri setinde ayrılmış fakat tahmin veri setinde ayrılmamış müşteridir. Bu durumda yanlış negatif olur.

Karışıklık matrisi kullanılarak XGBoost modeli için doğruluk değeri hesaplanmıştır. Denklem 6.7' de hesaplanmıştır.

$$\text{Doğruluk (Accuracy)} = \frac{231 + 1388}{219 + 1378 + 37 + 55} = 0.9586 = \%95{,}86 \qquad (6.7)$$

Çıkan sonuca göre XGBoost modeli %95,86 doğruluk göstermiştir. ROC eğrisi çizdirilmiş ve bu eğrinin altındaki alanı veren AUC değeri hesaplatılmıştır. Şekil 6.20' de ROC-AUC eğrisi verilmiştir.



Şekil 6.20. XGBoost modeli için ROC-AUC eğrisi

Grafiğe bakıldığında, AUC değeri 0.9688 olarak hesaplanmıştır. AUC değeri 1'e oldukça yaklaşmıştır. Bu durum XGBoost modelinin iyi bir model olduğunu göstermektedir.

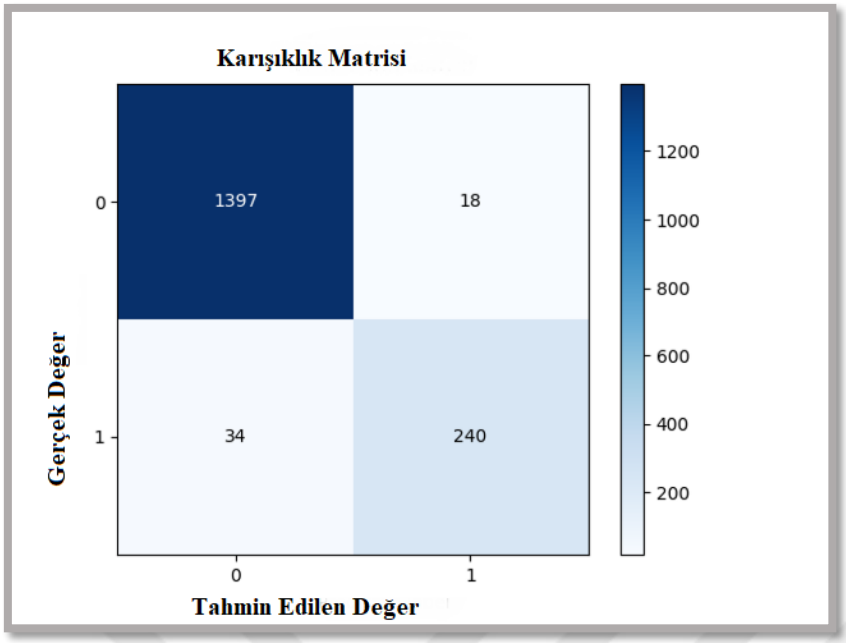
6.5.7. LightGBM modelleme

Günümüzde hızlı, güvenilir ve yaygın olarak kullanılan LightGBM modellemesi yapılmıştır. Uygulama için kurulan son modeldir. Modelleme yapmadan önce hiper parametre ayarlaması yapılmıştır. Kullanılacak ağaç sayısını veren "n_estimator" parametresinin ayarlaması için şu değerler denenmiştir: (100,200,500,1000). Ağacın

dallanmasını kontrol eden “max_depth” parametresinin ayarlaması için şu değerler denenmiştir: (3,5,7,8,10,18). Modele yeni eklenen ağaçların ağırlığını kontrol eden “learning_rate” parametresinin ayarlaması için şu değerler denenmiştir: (0.1,0.01,0.001). GridSearchCV yöntemi uygulanarak model için en iyi parametre buldurulmuştur. Değerlerin aralığı model için kullanılan veri setine göre verilmiştir. Aşırı öğrenmemin önüne geçmek için uygun değerler denenmiştir. GridSearchCV fonksiyonu 10 kat çapraz doğrulama (cv=10) ile parametrenin en iyi değerini aramıştır. Bu fonksiyon parametre için deneme yapılan değer sayısı ve belirlenen çapraz doğrulama sayısının çarpımı kadar modeli dener ve en iyi sonucu buldurur. Yani 4*6*3*10 yaparak 720 modeli denemiş ve en iyi sonuca ulaşmıştır. Tüm değerlerin kombinasyonunu deneyerek LightGBM modeli için en iyi sonuçları veren parametre değerlerini bulmuştur. Deneme sonucunda “n_estimator” parametresi için en iyi parametre değeri 1000, “max_feaures” için en iyi parametre değeri 18 ve “learning_rate” parametresi için en iyi değer 0.1’dir. Diğer parametre değerlerinde değişiklik yapılmamıştır ve varsayılan değerler kullanılmıştır. Hiperparametre ayarlaması yapıldıktan sonra LightGBM modeli kurulmuştur. Model önce eğitilmiş daha sonra test edilmiştir.

- LightGBM için (n_estimator= ‘1000’, learning_rate= ‘0,1’, max_features= ‘18’,) hiper parametre değerleri kullanılmıştır.

Modeli değerlendirmek için performans kriteri olarak doğruluk (accuracy) ve AUC değerine göre değerlendirme yapılmıştır. İlk olarak karışıklık matrisi Python’ da çizdirilmiştir. Şekil 6.21’ de LightGBM modeline ait karışıklık matrisi gösterilmiştir.



Şekil 6.21. LightGBM model için karışıklık matrisi

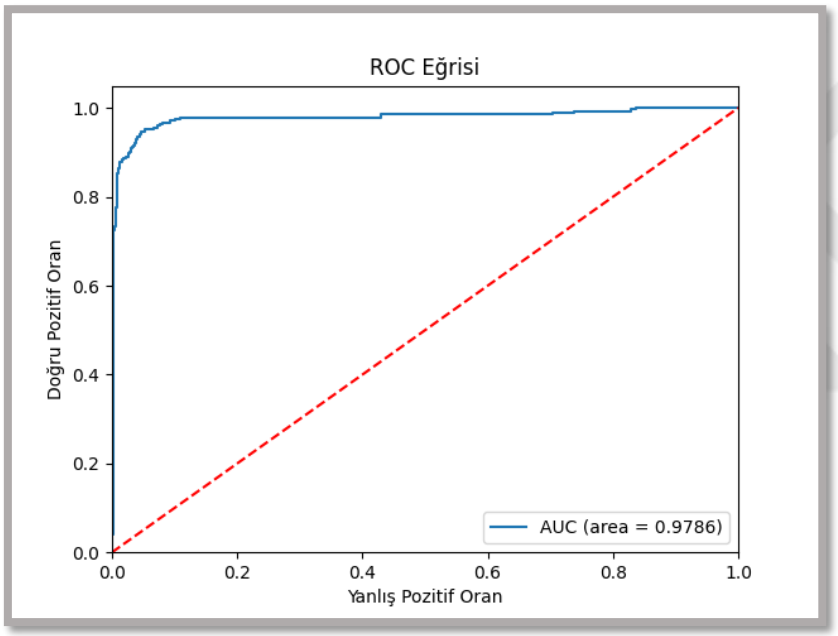
Karışıklık matrisine göre;

- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin de doğru olduğu (sınıf = 1) 240 müşteri görülmüştür. 240 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmış müşterilerdir. Bu durumda doğru pozitif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin de yanlış olduğu (sınıf = 0) 1397 müşteri görülmüştür. 1397 müşteri hem gerçek veri setinde hem de tahmin veri setinde de ayrılmamış müşterilerdir. Bu durumda doğru negatif olur.
- Veri setinin gerçek sınıfının yanlış (sınıf = 0) ve tahmin edilenin ise doğru olduğu (sınıf = 1) 18 müşteri görülmüştür. 18 müşteri gerçek veri setinde ayrılmamış fakat tahmin veri setinde ayrılmış müşterilerdir. Bu durumda yanlış pozitif olur.
- Veri setinin gerçek sınıfının doğru (sınıf = 1) ve tahmin edilenin ise yanlış olduğu (sınıf = 0) 34 müşteri görülmüştür. 34 müşteri gerçek veri setinde ayrılmış fakat tahmin veri setinde ayrılmamış müşteridir. Bu durumda yanlış negatif olur.

Karışıklık matrisi kullanılarak LightGBM modeli için doğruluk değeri hesaplanmıştır. Denklem 6.8'de hesaplama verilmiştir.

$$\text{Doğruluk (Accuracy)} = \frac{240 + 1397}{219 + 1378 + 37 + 55} = 0.9692 = \%96{,}92 \qquad (6.8)$$

Çıkan sonuca göre LightGBM modeli %96,92 doğruluk göstermiştir. ROC eğrisi çizdirilmiş ve bu eğrinin altındaki alanı veren AUC değeri hesaplatılmıştır. Şekil 6.22'de ROC-AUC eğrisi verilmiştir.



Şekil 6.22. LightGBM modeli için ROC-AUC eğrisi

Grafiğe bakıldığında, AUC değeri 0.9786 olarak hesaplanmıştır. AUC değeri 1'e oldukça yaklaşmıştır. ROC eğrisi ise y eksenine oldukça yaklaşmıştır. Bu durumda modelin başarılı bir sonuç verdiğini göstermiştir.

6.6. Modellerin Karşılaştırılması

Modelleme için toplam 7 tane algoritma kullanılmıştır. Tahmin modellerini değerlendirmek ve kıyaslamak için kullanılan performans kriteri, doğruluk ve AUC değerleri Çizelge 6.3' te verilmiştir.

Çizelge 6.3. Modellerin doğruluk ve AUC değerleri

Modeller	**Doğruluk (Accuracy)**	**AUC**
Lojistik Regresyon	0,8076	0,8665
Destek Vektör Makine	0,9100	0,9296
K-En Yakın Komşu	0,8667	0,9587
Karar Ağaçları	0,9035	0,8984
Rastgele Orman	0,9455	0,9653
XGBoost	0,9586	0,9688
LightGBM	**0,9692**	**0,9786**

Yukarıda bulunan modellerin sonuç tablosuna bakıldığında, performans kriteri olarak kullanılan doğruluk ve AUC değerlerine göre en iyi sonucu veren model LightGBM modelidir. Veri seti kullanılarak müşteri kaybı tahminini en iyi yapan modeldir. Bu modelden sonra XGBoost modeli en iyi sonucu vermiştir. LightGBM modeline yakın sonuçları vermiştir ancak LightGBM modeli, XGBoost modeline göre daha hızlı, güncel ve güvenilir olduğu için daha iyi sonuç vermiştir. Lojistik Regresyon modeli, diğer modellere göre performans olarak daha düşüktür. Nedeni bu model parametrik bir sınıflandırma algoritmasıdır ve diğer modellere göre daha basit yapıdadır.

6.7. Öznitelik Önem Düzeyi Belirleme

Öznitelik önem düzeyi hesaplamak ve analiz etmek çalışmanın bir diğer amacıdır. En iyi model olan LightGBM için Python'da öznitelik önem düzeyleri hesaplatılmıştır. Model için en önemli olan öznitelikten daha az önemli olan özniteliğe doğru sıralama yapılmıştır ve Çizelge 6.4' te verilmiştir.

Çizelge 6.4. LightGBM modeli için öznitelik önem düzeyleri

Öznitelikler	**Önem Düzeyleri**
WarehouseToHome	0,2975
CashbackAmount	0,2936
Complain	0,2486
Tenure	0,2147
DaySinceLastOrder	0,1903
NumberOfAddress	0,1749
SatisfactionScore	0,1713
NumberOfDeviceRegistered	0,0919
OrderCount	0,0833
CityTier	0,0774
OrderAmountHikeFromlastYear	0,0761
CouponUsed	0,0719
Gender_Male	0,0694
HourSpendOnApp	0,0685
MaritalStatus_Single	0,0671
PreferredPaymentMode_Debit_Card	0,0536
PreferredLoginDevice_Phone	0,0472
PreferedOrderCat_Laptop_Accessory	0,0468
PreferredLoginDevice_Mobile_Phone	0,0459
PreferredPaymentMode_Credit_Card	0,0326
PreferredPaymentMode_COD	0,0292
MaritalStatus_Married	0,0279
PreferredPaymentMode_E_wallet	0,0252
PreferedOrderCat_Mobile_Phone	0,0194
PreferredPaymentMode_UPI	0,0184
PreferredPaymentMode_Cash_on_Delivery	0,0135
PreferedOrderCat_Mobile	0,0112
PreferedOrderCat_Others	0,0055
PreferedOrderCat_Grocery	0,0013

Çizelge 6.4 ‘e bakıldığında hangi özniteliğin çıktı üzerinde ne kadar etkisi olduğu görülmektedir.

- “WarehouseToHome” özniteliğinin yani müşterinin depo ile evi arasındaki mesafenin çıktı üzerinde ki öneminin diğer özniteliklerden fazla olduğu görülmüştür. Müşterinin depo ile evi arasındaki mesafenin fazla olması çıktı üzerinde olumsuz bir etki oluşturmaktadır. Mesafe arttıkça firmanın müşteri kaybetme olasılığının artabileceği anlamına gelmektedir.
- Ardından “CashbackAmount” yani müşteriye geri ödenen ortalama miktarın çıktı üzerinde etkisi fazladır. Geri ödenen miktarın artması müşterinin iadesinin fazla olduğunu göstermektedir. Yani müşterinin memnuniyetinin az olduğunu ifade etmektedir. Geri ödenen miktar arttıkça firmanın müşteri kaybetme olasılığı artacak demektir.
- “Complain” yani müşterinin son ayda yaptığı şikayet sayısı diğer bir önemli etkendir. Müşterinin şikayet sayısı artması müşterinin firmadan memnun kalmadığını göstermektedir. Şikayet sayısının artması müşterinin yıpranmasına neden olacak ve firma ile müşteri ilişkilerini bitireceği anlamına gelmektedir.
- “Tenure” yani müşterinin firmada bulunduğu süre çıktı üzerinde etkisi olmuştur. Müşterinin firmada bulunduğu sürenin artması müşterinin memnun olduğunu ve ne kadar sadık bir müşteri olduğunu göstermektedir.

Model üzerindeki etkisi fazla olan öznitelikler e-ticaret firmalar için çok şey ifade etmektedir. E-ticaret firmaları bu öznitelikleri belirleyerek durumu kendileri için olumlu yönde kullanmalıdır. Çıktı üzerinde ki etkisi fazla olan öznitelikler dikkate alınıp olumsuz durumlar ele alınarak iyileştirme çalışmaları yapılabilir. Böylece hem müşteriyi elde tutmak hem de yeni müşteri kazanmak için çeşitli kampanyalar, promosyonlar, reklamlar yapabilir. Böylelikle firma müşteri kaybını minimize etme imkanı bulmuş olacaktır.

7. SONUÇ VE ÖNERİLER

Bu çalışmada e-ticaret sektöründeki müşteri kaybı tahminini yapmak amacıyla yapay öğrenme algoritmaları kullanılarak tahmin modelleri kurulmuş ve bu tahmin modelleri, performans kriterlerine göre kıyaslanıp hangi modelin daha uygulanabilir olduğu tartışılmıştır. Çalışma için kullanılan veri seti açık erişimli olan internet sitesinden alınmıştır. Bu çalışma e-ticaret sektöründeki müşteri kaybı üzerindeki etkenleri bir araya getiren veri setini keşfetmek ve tanımlamakla başlamıştır. İlk olarak veri setindeki öznitelikler tanımlanmıştır. Veri setini analize ve modellemeye uygun hale getirebilmek amacıyla veri ön işleme yapılmıştır. Veri ön işleminin ilk adımı olarak eksik veriler tespit edilmiş ve bu eksik verileri silmenin veri setini olumsuz etkileyeceğine karar verildiği için eksik olan verilerin ait olduğu özniteliğin ortalama değeri ile doldurulmuştur. İkinci adım olarak ise yapay öğrenme modelleri için sorun teşkil eden kategorik veriler, veri setinde tespit edilmiş ve sayısal değere dönüştürülmüştür. Son adım olarak veri setindeki sayısal değerler aynı değer aralığında tanımlanması için verilere standartlaştırma işlemi uygulanmıştır. Tahmin için kurulacak makine öğrenmesi modelinin eğitilmesi ve test edilmesi için veri seti eğitim ve test seti olarak ayrılmıştır. Veri setinde veri dengesizliği (imbalance) olduğu tespit edilmiştir. Sınıf dengesizliğini ve aşırı öğrenmeyi engellemek amacıyla SMOTE yöntemi uygulanmıştır. Böylelikle veri seti dengeli hale getirilmiştir. Müşteri kaybı tahmini yapmak için veri setinin son hali kullanılarak yapay öğrenme modelleri kurulmuştur. Yapay öğrenmeye ait sınıflandırma algoritmaları olarak Lojistik Regresyon, K-En Yakın Komşu, Destek Vektör Makine, Karar Ağacı, Rastgele Orman, XGBoost ve LightGBM kullanılmıştır. Literatürde çoğunlukla klasik olan yapay öğrenme algoritmaları tercih edilmiştir. Bu çalışmada günümüzde güncel ve popüler olan ve daha iyi sonuçlar vermesi için geliştirilen LightGBM sınıflandırma algoritması da kullanılmıştır. En iyi modellerin kurulması amacıyla her model için hiper parametre ayarı yapılmıştır. Böylelikle modeller en iyi parametre değerleri ile kurulmuş ve modellerin kalitesi artırılmıştır. Modelleri kıyaslamak ve hangi modelin daha uygulanabilir olduğuna karar vermek için performans kriteri olarak doğruluk (accuracy) ve AUC değerleri kullanılmıştır. Doğruluk oranının tek başına karar verici bir kriter olarak kullanılması istenmediği için güvenilir sonuçlar veren AUC değeri de performans kriteri olarak kullanılmıştır. Bunlar literatür ve gerçek yaşam uygulamaları için de en çok kullanılan performans kriterleridir. Hem doğruluk hem de AUC

değerine göre modeller kıyaslanmıştır. En iyi sonucu veren model LightGBM modeli olarak tespit edilmiştir. Bu modelin müşteri kaybı tahmini için daha güvenilir ve iyi sonuçlar verdiği görülmüştür. Sonuç olarak bu çalışma, LightGBM modelinin en iyi sonuç vermesi ile LightGBM algoritmasını geliştirenlerin iddia ettiği gibi hem tahmin gücünün hem de öğrenim hızının iyi olduğunu desteklemiştir. Bu çalışma sadece en iyi tahmin modelini bulup çalışmayı sonlandırmamıştır. En iyi tahmin modeline karar verildikten sonra öznitelik önem düzeylerine göre de analiz yapılmıştır. LightGBM modeli için öznitelik önem düzeyi incelendiğinde "WarehouseToHome" özniteliğinin yani müşterinin depo ile evi arasındaki mesafenin müşteri kaybı tahmini üzerinde ki öneminin diğer özniteliklerden fazla olduğu görülmüştür. Ardından "CashbackAmount" yani müşteriye geri ödenen ortalama miktar, "Complain" yani müşterinin son ayda yaptığı şikayet sayısı ve "Tenure" yani müşterinin firmada bulunduğu süresi gibi öznitelikler de diğer öznitelik kadar çıktı üzerinde etkisi olduğu tespit edilmiştir. Bu çalışmada en iyi model araştırması yanında, öznitelikler üzerinde yapılan önem düzeyi analizi, e-ticaret firmalarının kendi müşteri kaybı tahmini araştırmalarında hangi faktör ya da faktörlerin/özniteliklerin daha önemli etkiye sahip olduklarını anlamalarında da yol gösterici nitelikte olmuştur. Sonuç olarak, önerilen model kullanılarak ve daha önemli öznitelikler göz önüne alınarak firmaların müşteri kayıp oranlarını en düşük seviyeye indirmelerine olanak sağlanmıştır.

Çalışmada model araştırması 19 özniteliğe sahip 5630 adetlik veri seti üzerinden yapılmıştır. Gelecekteki çalışmalar veri setindeki öznitelik sayısını artırarak çeşitlendirebilir. Ayrıca veri seti büyüklüğünü artırarak çalışmayı genişletebilir. Diğer bir öneri olarak rakip e-ticaret firmalarının verileri kullanılıp kıyaslanarak hangi e-ticaret firmasının sektöründe daha başarılı olduğuna karar verilebilir. Yani başka veri setleri ile model gelişimi arttırılabilir.

KAYNAKLAR DİZİNİ

Abioye, S. O., Oyedele, L. O., Akanbi, L., Ajayi, A., Delgado, J. M. D., Bilal, M., ... & Ahmed, A. (2021). Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges. Journal of Building Engineering, 44, 103299. https://doi.org/10.1016/j.jobe.2021.103299

Al Daoud, E. (2019). Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset. International Journal of Computer and Information Engineering, 13(1), 6-10.

Amari, S. I., & Wu, S. (1999). Improving support vector machine classifiers by modifying kernel functions. Neural Networks, 12(6), 783-789.

Azar, A. T., Elshazly, H. I., Hassanien, A. E., & Elkorany, A. M. (2014). A random forest classifier for lymph diseases. Computer methods and programs in biomedicine, 113(2), 465-473. http://dx.doi.org/10.1016/j.cmpb.2013.11.004

Bagul, N., Berad, P., Surana, P., & Khachane, C. (2021). Retail customer churn analysis using rfm model and k-means clustering. Int J Eng Res Technol (IJERT), 10(3), 349-354.

Bardenet, R., Brendel, M., Kégl, B., & Sebag, M. (2013, May). Collaborative hyperparameter tuning. In International conference on machine learning (pp. 199-207). PMLR.

Başer, B. Ö., Yangın, M., & Sarıdaş, E. S. (2021). Makine öğrenmesi teknikleriyle diyabet hastalığının sınıflandırılması. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 25(1), 112-120. https://doi.org/10.19113/sdufenbed.842460

Batista, G. E. A. P. A., & Silva, D. F. (2009, August). How k-nearest neighbor parameters affect its performance. In Argentine symposium on artificial intelligence (pp. 1-12).

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54(3), 1937-1967.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195-202. https://doi.org/10.1038/nature23474

Bittencourt, H. R., de Oliveira Moraes, D. A., & Haertel, V. (2007, July). A binary decision tree classifier implementing logistic regression as a feature selection and classification method and its comparison with maximum likelihood. In 2007 IEEE international geoscience and remote sensing symposium (pp. 1755-1758). IEEE.

KAYNAKLAR DİZİNİ (devam)

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. Expert Systems with Applications, 36(3), 4626-4636. https://doi.org/10.1016/j.eswa.2008.05.027

CFI Team. (2021), *Random Forest*, Erişim: https://corporatefinanceinstitute.com/resources/knowledge/other/random-forest/ Erişim tarihi: 28.09.2022

Chinnamgari, S. K. (2019). R Machine Learning Projects: Implement supervised, unsupervised, and reinforcement learning techniques using R 3.5. Packt Publishing Ltd.

Chomboon, K., Chujai, P., Teerarassamee, P., Kerdprasop, K., & Kerdprasop, N. (2015, March). An empirical study of distance metrics for k-nearest neighbor algorithm. In Proceedings of the 3rd international conference on industrial application engineering (pp. 280-285).

Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In Machine learning techniques for multimedia (pp. 21-49). Springer, Berlin, Heidelberg.

Çakırdoğan, H.A. (2021). Makine öğrenmesi ile e-ticarette müşteri kaybı tahmini (Tez No. 675863) [Yüksek Lisans Tezi, Millî Savunma Üniversitesi Hezârfen Havacılık ve Uzay Teknolojileri Enstitüsü]. YÖK Ulusal Tez Merkezi. https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp

DeCastro-García, N., Muñoz Castañeda, Á. L., Escudero García, D., & Carriegos, M. V. (2019). Effect of the sampling of a dataset in the hyperparameter optimization phase over the efficiency of a machine learning algorithm. Complexity, 2019. https://doi.org/10.1155/2019/6278908

Dick, S. (2019). Artificial Intelligence. Harvard Data Science Review, 1(1). https://doi.org/10.1162/99608f92.92fe150c

E-ticaret Bilgi Platformu (ETBİS). (2021). E-ticaret hacminin sektör itibarıyla bir önceki yılın aynı dönemine göre değişimi. Erişim: https://www.eticaret.gov

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. Journal of machine learning research, 4(2003), 933-969.

Friedl, M. A., & Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. Remote sensing of environment, 61(3), 399-409. https://doi.org/10.1016/S0034-4257(97)00049-7

Gedik, Y. (2021). E-Ticaret: Teorik bir çerçeve. Ankara Üniversitesi Sosyal Bilimler Dergisi, 12(1), 184-198.

KAYNAKLAR DİZİNİ (devam)

Gong, M. (2021). A novel performance measure for machine learning classification. International Journal of Managing Information Technology (IJMIT), 13(1). https://doi.org/10.5121/ijmit.2021.13101

Guo, F., & Qin, H. L. (2015, November). The Analysis of Customer Churns in e-Commerce Based on Decision Tree. In 2015 International Conference on Computer Science and Applications (CSA), pp. 199-203. IEEE. Beijing, China https://doi.org/10.1109/CSA.2015.74

Hamet, P., & Tremblay, J. (2017). Artificial intelligence in medicine. Metabolism, 69, S36-S40. http://dx.doi.org/10.1016/j.metabol.2017.01.011

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. Electronic Markets, 31(3), 685-695. https://doi.org/10.1007/s12525-021-00475-2

Jiang, L., Cai, Z., Wang, D., & Jiang, S. (2007, August). Survey of improving k-nearest-neighbor for classification. In Fourth international conference on fuzzy systems and knowledge discovery (FSKD 2007) (Vol. 1, pp. 679-683). IEEE. Wuhan, China. https://doi.org/10.1109/FSKD.2007.552

Ju, K. (2022). The Influence of Electronic Commerce on International Trade and Development Strategy. Asian Business Research, 7(3), 33. https://doi.org/10.20849/abr.v7i3.1123

Ju, Y., Sun, G., Chen, Q., Zhang, M., Zhu, H., & Rehman, M. U. (2019). A model combining convolutional neural network and LightGBM algorithm for ultra-short-term wind power forecasting. *Ieee Access*, *7*, 28309-28318.

Kang, M., & Jameson, N. J. (2018). Machine Learning: Fundamentals. Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things, 85-109. https://doi.org/10.1002/9781119515326.ch4

Kaynar, O., Tuna, M. F., Görmez, Y., & Deveci, M. A. (2017). Makine öğrenmesi yöntemleriyle müşteri kaybı analizi. Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi, 18(1), 1-14.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), 3-24.

Li, K., Zhang, W., Lu, Q., & Fang, X. (2014, October). An improved SMOTE imbalanced data classification method based on support degree. In 2014 international conference on identification, information and knowledge in the internet of things (pp. 34-38). IEEE.

KAYNAKLAR DİZİNİ (devam)

Li, X., & Li, Z. (2019). A Hybrid Prediction Model for E-Commerce Customer Churn Based on Logistic Regression and Extreme Gradient Boosting Algorithm. Ingénierie des Systèmes d'Information, 24(5). 525-530. https://doi.org/10.18280/isi.240510

Lorena, A. C., De Carvalho, A. C., & Gama, J. M. (2008). A review on the combination of binary classifiers in multiclass problems. Artificial Intelligence Review, 30(1), 19-37. http://dx.doi.org/ 10.1007/s10462-009-9114-9

McCarthy, J. (2004). What is artificial intelligence. URL: http://www-formal. stanford. edu/jmc/whatisai. html.

Murthy, C. S. V. (2007). E-Commerce-Concepts, Models And Strategies. Himalaya Publishing

Narkhede, S. (2018). Understanding auc-roc curve. Towards Data Science, 26(1), 220-227.

Novaković, J. D., Veljović, A., Ilić, S. S., Papić, Ž., & Milica, T. (2017). Evaluation of classification models in machine learning. Theory and Applications of Mathematics & Computer Science, 7(1), 39-46.

Priyanka, & Kumar, D. (2020). Decision tree classifier: A detailed survey. International Journal of Information and Decision Sciences, 12(3), 246-269.

Raeisi, S., & Sajedi, H. (2020, October). E-Commerce Customer Churn Prediction By Gradient Boosted Trees. In 2020 10th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 055-059. IEEE. Tehran, Iran. https://doi.org/10.1109/ICCKE50421.2020.9303661.

Ray, S. (2019, February). A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE.

Shinde, P. P., & Shah, S. (2018, August). A review of machine learning and deep learning applications. In 2018 Fourth international conference on computing communication control and automation (ICCUBEA) (pp. 1-6). IEEE.

Triguero, I., García-Gil, D., Maillo, J., Luengo, J., García, S., & Herrera, F. (2019). Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(2), e1289.

KAYNAKLAR DİZİNİ (devam)

Tsangaratos, P., & Ilia, I. (2016). Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size. Catena, 145, 164-179.

Vafeiadis, T., Diamantaras, K. I., Sarigiannidis, G., & Chatzisavvas, K. C. (2015). A comparison of machine learning techniques for customer churn prediction. Simulation Modelling Practice and Theory, 55, 1-9.http://dx.doi.org/10.1016/j.simpat.2015.03.003

Vladimir, Z. (1996). Electronic commerce: structures and issues. International journal of electronic commerce, 1(1), 3-23. https://doi.org/10.1080/10864415.1996.11518273

Wang, J., Xu, M., Wang, H., & Zhang, J. (2006, November). Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding. In 2006 8th international Conference on Signal Processing (Vol. 3). IEEE.

Wang, L., & Liu, S. (2020, July). Research on E-commerce Customer Relationship Management Based on Data Analysis. In 2020 The 11th International Conference on E-business, Management and Economics, pp. 20-26. Beijing, China. https://doi.org/10.1145/3414752.3414776

Wu, X., & Meng, S. (2016, June). E-commerce customer churn prediction based on improved SMOTE and AdaBoost. In 2016 13th International conference on service systems and service management (ICSSSM), pp. 1-5. IEEE. Shanghai, China. https://doi.org/ 10.1109/ICSSSM.2016.7538581.

Xiahou, X., & Harada, Y. (2022). B2C E-Commerce Customer Churn Prediction Based on K-Means and SVM. Journal of Theoretical and Applied Electronic Commerce Research, 17(2), 458-475. https://doi.org/10.3390/jtaer17020024

Xie, Y., Li, X., Ngai, E. W. T., & Ying, W. (2009). Customer churn prediction using improved balanced random forests. Expert Systems with Applications, 36(3), 5445-5449. https://doi.org/10.1016/j.eswa.2008.06.121

Yanfang, Q., & Chen, L. (2017, December). Research on E-commerce user churn prediction based on logistic regression. In 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 87-91. IEEE. Chengdu, China. https://doi.org/10.1109/ITNEC.2017.8284914.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 415, 295-316. https://doi.org/10.1016/j.neucom.2020.07.061

KAYNAKLAR DİZİNİ (devam)

Yu, X., Guo, S., Guo, J., & Huang, X. (2011). An extended support vector machine forecasting framework for customer churn in e-commerce. Expert Systems with Applications, 38(3), 1425-1430. https://doi.org/10.1016/j.eswa.2010.07.049

Z, Y. (2018). Research on E-commerce customer churn prediction based on improved value model and XG-boost algorithm. Management Science and Engineering, 12(3), 51-56. http://dx.doi.org/10.3968/10816

Zhao, Y., Li, B., Li, X., Liu, W., & Ren, S. (2005, July). Customer churn prediction using improved one-class support vector machine. In International conference on advanced data mining and applications (pp. 300-306). Springer, Berlin, Heidelberg.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1), 1-130.