



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



KARDİYOVASKÜLER HASTALIKLARININ
TEŞHİSİNE YÖNELİK MAKİNE ÖĞRENMESİ
ALGORİTMALARI İLE BİR KARAR DESTEK
SİSTEMİ TASARIMI

AMİR KARAJ

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği

Anabilim Dalı

Bilgisayar Mühendisliği Programı

DANIŞMAN

Dr. Öğr. Üyesi Zehra Aysun ALTIKARDEŞ

EŞ-DANIŞMAN

Prof. Dr. Ali Serdar FAK

İSTANBUL, 2023



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



KARDİYOYASKÜLER HASTALIKLARININ
TEŞHİSİNE YÖNELİK MAKİNE ÖĞRENMESİ
ALGORİTMALARI İLE BİR KARAR DESTEK
SİSTEMİ TASARIMI

AMİR KARAJ

523619921

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği

Anabilim Dalı

Bilgisayar Mühendisliği Programı

DANIŞMAN

Dr. Öğr. Üyesi Zehra Aysun ALTIKARDEŞ

EŞ-DANIŞMAN

Prof. Dr. Ali Serdar FAK

İSTANBUL, 2023

TEŐEKKÜR

Öncelikle sabrımdan, gayretimden, kararlılıđımdan ötürü kendime,

Tez çalışmam sürecinde daima yönlendiren, destek olan, yardımını esirgemeyen danışmanım Dr. Öğr. Üyesi Zehra Aysun Altıkardeş hocama ve eş danışmanım Prof. Dr. Ali Serdar Fak hocama,

Hipertansiyon ve Ateroskleroz Eğitim, Uygulama ve Araştırma Merkezi çalışanlarına, gerçekleştirilen bu ortak çalışmada destekleri için teşekkür ederim.

Maddi, manevi yardımlarını esirgemeyen ve dualarıyla yanımda olan aileme,

Moral veren, motive eden dostlarıma, arkadaşlarıma

Teşekkür ederim.

Ocak, 2023

Amir Karaj

İÇİNDEKİLER

ÖZET	I
ABSTRACT.....	III
KISALTMALAR	V
TABLO LİSTESİ.....	XII
1. GİRİŞ	1
1.1. AMAÇ, HEDEF VE UYGULAMA	2
1.1.1. Amaç.....	2
1.1.2. Hedef.....	2
1.2. LİTERATÜR TARAMASI.....	2
1.2.1. Tıpta Yapay Zekâ.....	5
1.3. UYGULAMANIN FARKLILIKLARI.....	6
2. MATERYAL VE YÖNTEM.....	8
2.1. YAPAY ZEKÂ (AI) VE MAKİNE ÖĞRENME (ML)	9
2.2. YAPAY ZEKÂ (AI).....	10
2.3. YAPAY ZEKANIN (AI) ÇALIŞMA MANTIĞI.....	11
2.4. MAKİNE ÖĞRENİMİ (ML).....	11
2.5. SAĞLIKTA MAKİNE ÖĞRENİMİ	11
2.6. VERİ SETİ.....	12
2.6.1. Veri seti analizi	14
2.7. YÖNTEM.....	18
2.7.1. Python	18
2.7.2. Scikit-learn.....	19
2.7.3. Plotly	19
2.7.4. Dash	20

2.7.5. PostgreSQL.....	20
2.7.6. Heroku.....	21
2.7.7. Makine öğrenmesi algoritmaları.....	21
3. BULGULAR.....	35
3.1. PERFORMANS ÖLÇÜTLERİ.....	35
3.1.1. Accuracy / Doğruluk / Kesinlik.....	36
3.1.2. Precision / Hassasiyet.....	37
3.1.3. Recall / Duyarlılık / Geri Çağırma / Sensitivity.....	37
3.1.4. F1-Score.....	37
3.1.5. Özgüllük / Specificity.....	38
3.1.6. Root Mean Square Error (RMSE) / Ortalama karesel hata karekökü.....	38
3.1.7. Balanced Accuracy / Dengeli Doğruluk.....	38
3.1.8. ROC ve PR Eğrileri.....	39
3.2. ANALİZ SONUÇLARI.....	41
3.2.1. Yapay Sinir Ağları (Artificial Neural Network ANN).....	41
3.2.2 Destek Vektör Makinesi (Support Vector Machine SVM).....	46
3.2.3. K-en Yakın Komşu (K-Nearest Neighbor K-NN).....	50
3.2.4 Karar ağacı (Decision Tree DT).....	54
3.2.5. Naive Bayes NB.....	56
3.2.6 CatBoost.....	59
3.3. KARŞILAŞTIRMA VE TARTIŞMA.....	61
3.4. MEDIC-AI ML DESTEKLİ KDS WEB UYGULAMASI.....	65
4. SONUÇLAR.....	74
KAYNAKLAR.....	78
ÖZGEÇİMİ.....	87

ÖZET

Kardiyovasküler Hastalıklarının Teşhisine Yönelik Makine Öğrenmesi Algoritmaları ile Karar Destek Sistemi Tasarımı.

Sağlık sektörü çok büyük ve hassas veriler içerir ve çok dikkatli bir şekilde ele alınması gerekmektedir. Büyük miktarda verinin erişilebilirliği ve kullanılabilirliği, belirli veri madenciliği teknikleri uygulandığında yararlı bilgiler sağlayabilecektir. Son yıllarda gelişen yapay zekâ ve makine öğrenme teknolojileri, tıp ve sağlık alanlarında hizmetlerini dönüştürmeye başlıyor. Şu anda sağlık alanlarında karar verme araçları yaygın olarak kullanılmaktadır. En doğru modelleri oluşturmak için birçok analiz ve sınıflandırma yöntemleri bulunmakta ve kullanılmaktadır. Hastalıkların tanı ve tedavisinde büyük bir rol oynamaktadır. Makine öğrenme algoritmaları birçok nadir hastalıklarda uygulanmaktadır. Tıbbi uygulamalar araştırılmasında makine öğrenme teknikleri büyük bir rol oynamaktadır.

Tezin amaç ve hedef doğrultusunda MEDIC-AI ML destekli web uygulaması geliştirildi. Geliştirilen web uygulamamız üzerinde farklı anonim veri setleri kullanılmıştır. Kullanılan anonim veri setleriyle, uygulama üzerine veri analizi, veri temizleme gibi özellikler kullanıcıya sunulmuştur. Bu web uygulamada kullanıcı tarafından, veri tabanından yüklenen verilerden, istenilecek herhangi bir veri tablosu seçip, ilk önce verinin analizini ve incelemesini web uygulaması üzerine yapılabilme şeklinde tasarlanmış ve geliştirilmiştir. Daha sonra kullanıcıya sunulan ML algoritmalarından, istenilen bir algoritma seçerek kullanılan veri setleri, tablo üzerine bir örneklem seçerek tahminleme sonuçlarını ve bu sonuçların grafiklerini ve raporlarını inceleyebilmektedir. Sunulan farklı ML algoritmalar üzerine deneme ve tahminleme sonuçları alabilir, tahminleme sonuçlarının karşılaştırılması ve grafikler üzerine incelenme ve analizini yapabilir, kullanıcıya sunulan farklı grafikler üzerine veri incelenmesi ve analizi yapabilmesi gibi özelliklerle, web üzerine ML destekli bir uygulama geliştirilmiştir.

Çalışma kapsamında sadece kardiyovasküler hastalıklarını araştırılmasına yönelik bir karar destek sistemi tasarlanmamıştır. Geliştirilen MEDIC-AI web uygulamamız, dosya yükleme özelliği ile veri tabanında yüklenecek anonim ya da herhangi bir veri seti ile farklı hastalıkların, kullanıcıya sunulan birden fazla (yedi (7)) makine öğrenme algoritma teknikleri ile tahminleme sonuçlarını ve analizlerini yapabilmektedir. Farklı

makine öğrenme teknikleri kullanarak elde olan veri setlerini farklı açılardan yaklaşması ve incelenmesi yapılmıştır. Elde edilen sonuçların değerlendirilmesi ve özetlenmesi yapılmıştır. Bu çalışmada regresyon algoritmaları yanı sıra kullanılan makine öğrenme algoritmaları, (K-Means) kümeleme ve sınıflandırma için Destek Vektör Makineleri (SVM), Yapay Sinir Ağları (ANN), K En Yakın Komşu (kNN), Karar Ağac (DT), Naive Bayes (NB) ve CatBoost algoritmalarıdır.

Bu çalışmada veri setinin analizi, temizlenmesi ve araştırılması geliştirilen uygulama üzerine yapılmaktadır. Uygulama Python, Flask ve JavaScript ile birleşiminden oluşan Dash Plotly programlama platformu ile geliştirilmiştir. Python programlama dilli makine öğrenme, derin öğrenme ve veri madenciliği tekniklerinde çok güçlü ve kullanımı rahat olan dillerden biri olmaktadır. Dash Plotly geliştirme platformu, sunduğu plotly grafiklerini web üzerine canlı olarak grafik çizimi için en ideal platformlardan biri olmaktadır.

Anahtar kelimeler: Makine öğrenmesi, hipertansiyon, kardiyovasküler hastalıkları, kümeleme algoritmaları, dash plotly, denetimsiz öğrenme, postgresql

ABSTRACT

Decision Support System Design with Machine Learning Algorithms for the Diagnosis of Cardiovascular Diseases.

The healthcare industry contains huge and sensitive data and needs to be handled very carefully. The accessibility and usability of large amounts of data can provide useful insights when certain data mining techniques are applied. Machine learning technologies in artificial intelligence that are developed recent years are beginning to transform their services in medicine and health. Currently, decision-making tools are widely used in healthcare fields. There are many analysis and classification methods available and used to create the most accurate models. It has an important role for diagnosis and treatment of diseases. Machine learning algorithms are applied on many difficult and rare diseases. Machine learning techniques play a major role in researching medical applications.

MEDIC-AI ML supported web application was developed in accordance with the goal and objective of this thesis. Various anonymous data sets are used in the web application we developed. The user is offered features like data analysis and data cleaning for the anonymous datasets used. This web application is developed in a way that user can select any desired data table from the anonymous data and load it from the database. After loading the data table user can analyze and examine the data on the application. Then, by selecting a sample on the table and by choosing a desired algorithm from the ML algorithms presented, user can examine and analyses the result of the anonymous data sets results, can see the graphics and reports of these results. The application is developed with features such as testing and estimation results on different ML algorithms offered to users, comparison of estimation results, analysis on graphics, and data examination and analysis on different graphics.

In the scope of the study, were not investigated and analyzed only cardiovascular diseases. MEDIC-AI web application with the file upload feature offered to the users which can upload to the database anonymously or any data set, can make prediction, take results, and make analyzes for different diseases with multiple (seven (7)) machine learning algorithm techniques. By using different machine learning techniques, the obtained data sets were approached and analyzed from different angles. Evaluation and summary of the obtained results were made. In this study,

besides regression algorithms, machine learning algorithms used, (K-Means) clustering and classification, Support Vector Machines (SVM), Artificial Neural Networks (ANN), K Nearest Neighbor (kNN), Decision Tree (DT), Naive Bayes (NB) and CatBoost are algorithms.

In this study, the analysis, cleaning, and research of the data set is done on application. The application is developed with Dash Plotly programming platform, a combination of Python, Flask and JavaScript programming languages. Python programming language is powerful and most comfortable languages in ML, DL, and data mining techniques. The Dash Plotly development platform is one of the most ideal platforms for drawing live graphics on the web with the plotly graphics it offers.

Keywords: Machine learning, hypertension, cardiovascular diseases, clustering algorithms, dash plotly, unsupervised learning, postgresql.

KISALTMALAR

ABPM	: Ambulatory Blood Pressure Monitor
ACC	: Accuracy
AI	: Artificial Intelligence
ANN	: Artificial Neural Network
API	: Application Programming Interface
BSD	: Berkley Source Distributor
DL	: Deep Learning
DT	: Decision Tree
FN	: False Negative
FP	: False Positive
HİPAM	: Hipertansiyon ve Ateroskleroz Eğitim, Uygulama ve Araştırma Merkezi
K-NN	: K Nearest Neighbors
KDS	: Karar Destek Sistemi
KVH	: Kardiyovasküler Hastalık
MDP	: Markov Decisin Progres
ML	: Machine Learning
MLE	: Maksimum Likelihood Estimation
MRI	: Magnetic Resonance İmaging
NB	: Naive Bayes
Ör.	: Örnek
P	: Precision
PPV	: Positive Prediction Value
PR	: Precision Recall
RMSE	: Root Mean Square Error
ROC	: Receiver Operating Characteristic
SQL	: Structured Query Language
SVM	: Suport Vector Machine

TN : True Negative
TNR : True Negative Rate
TP : True Positive
TPR : True Positive Rate



ŞEKİL

Şekil 1.1.yapay zekanın her alanında kullanıldığını gösterimi.....	1
Şekil 1.2. Yapay zekâ algoritmaları çalışma prensipleri.....	5
Şekil 1.3. Bir yapay zeka modelin geliştirilmesi.....	5
Şekil 2.1. Tez çalışmasında kullanılan donanımsal ve yazılımsal sistem mimarisi.....	9
Şekil 2.2. Yapay zekâ, makine öğrenme ve derin öğrenme ilişkisi.....	10
Şekil 2.3. Boş veri renklendirme gösterimi.....	13
Şekil 2.4. Uygulama üzerine silme işlemi.....	13
Şekil 2.5. Kaydetme işlemi yapıldığında veri tabanında oluşturulan yeni tablo.....	13
Şekil 2.6. Özellikler ve örneklem sayısı gösterimi.....	14
Şekil 2.7. Veri analizi için grafikler.....	15
Şekil 2.8. Seçilen özelliklerinin kullanıcıya sunulan scatterplot matrix grafiklerinin türleri.....	15
Şekil 2.9. Seçilen özelliklerinin scatterplot matrix grafiklerinin popup ile gösterimi...	16
Şekil 2.10. İki özelliklerini karşılaştırmak için kullanıcıya sunulan grafik türleri.....	16
Şekil 2.11. İki özelliklerini histogram grafiği ile karşılaştırılması.....	17
Şekil 2.12. İki özelliklerini bar grafiği ile karşılaştırılması.....	17
Şekil 2.13. Tez çalışmasında kullanılan makine öğrenmesi algoritmalar.....	22
Şekil 2.14. Uygulamamızda kullanılan algoritmalar.....	23
Şekil 2.15. Uygulamada kullanılan algoritmaların genel akış şeması.....	23
Şekil 2.16. Svm' nin sınır çizgisinin belirlenmesi.....	24
Şekil 2.17. Svm' nin sınır çizgilerinin paralel çizgisinin belirlenmesi [59].....	24
Şekil 2.18. Ann mimari yapısı.[62].....	25
Şekil 2.19. Ann çalışma prensibi [63].....	26
Şekil 2.20. K-nn algoritmanın uygulanması.....	27
Şekil 2.21. Knn algoritmasının k parametre seçimi.....	27
Şekil 2.22. İkili hedef değişken y'ye dayalı örnek karar ağacı.....	28

Şekil 2.23. Bayes teoremi örnek gösterimi.....	29
Şekil 2.24. Naive bayes sınıflandırması.....	30
Şekil 2.25. Catboost algoritmanın çalışma prensibi.....	30
Şekil 2.26. Uygulamamızda kullanılan regresyon algoritmalar.....	31
Şekil 2.27. Uygulamada linear regresyon gösterimi.....	32
Şekil 2.28. Logistic regresyon modeli genel gösterimi.....	32
Şekil 2.29. Uygulamada logistic regresyon modeli gösterimi.....	33
Şekil 2.30. K-means algoritmanın çalışma prensibi.....	34
Şekil 2.31. Uygulamada k-means kümeleme algoritma gösterimi.....	34
Şekil 3.1. Uygulamada karmaşıklık matris gösterimi.....	36
Şekil 3.2. Doğruluk formula gösterimi [83].....	36
Şekil 3.3. Hassasiyet veya pozitif tahmin değeri (ppv) [83].....	37
Şekil 3.4. Duyarlılık, çağırma, isabet oranı veya gerçek pozitif oran (tpr) [83].....	37
Şekil 3.5. F1-score hassasiyet ve duyarlılığın harmonik ortalamasıdır [83].....	37
Şekil 3.6. Özgüllük, seçicilik veya gerçek negatif oranı (tnr) [83].....	38
Şekil 3.7. Rmse denklem gösterimi.....	38
Şekil 3.8. Dengeli doğruluk formula gösterimi [83].....	39
Şekil 3.9. Roc eğrisi grafiksel gösterimi.....	40
Şekil 3.10. Pr eğrisi grafiksel gösterimi.....	40
Şekil 3.11. Uygulamada mlpclassifier kütüphane kullanımı.....	41
Şekil 3.12. Seçilen verinin ann tahmini sonuç.....	42
Şekil 3.13. Seçilen verinin ann algoritmanın karmaşıklık matrisi.....	43
Şekil 3.14. Seçilen verinin ann algoritmanın roc eğimi.....	43
Şekil 3.15. Seçilen verinin ann algoritmanın pr eğimi.....	43
Şekil 3.16. Seçilen verinin ann tahmini sonuç.....	44
Şekil 3.17. Seçilen verinin ann karmaşıklık matrisi.....	45
Şekil 3.18. Seçilen verinin ann roc eğimi.....	45

Şekil 3.19. Seçilen verinin ann pr eğimi.....	45
Şekil 3.20. Svm algoritmanın svc kütüphane kullanımı.....	46
Şekil 3.21. Seçilen verinin svm tahmini sonuç.....	46
Şekil 3.22. Seçilen verinin svm algoritmanın karmaşıklık matrisi.....	47
Şekil 3.23. Seçilen verinin svm algoritmanın roc eğimi.....	47
Şekil 3.24. Seçilen verinin svm algoritmanın pr eğimi.....	48
Şekil 3.25. Seçilen verinin svm algoritmanın tahmini sonuç.....	48
Şekil 3.26. Seçilen verinin svm algoritmanın karmaşıklık matrisi.....	49
Şekil 3.27. Seçilen verinin svm algoritmanın roc eğimi.....	49
Şekil 3.28. Seçilen verinin svm algoritmanın pr eğimi.....	49
Şekil 3.29. K-nn algoritmanın kneighborsclassifier kütüphane kullanımı.....	50
Şekil 3.30. Seçilen verinin k-nn algoritma tahmini sonuç.....	50
Şekil 3.31. Seçilen verinin k-nn algoritma karmaşıklık matrisi.....	51
Şekil 3.32. Seçilen verinin k-nn algoritma roc eğimi.....	51
Şekil 3.33. Seçilen verinin k-nn algoritma pr eğrisi.....	52
Şekil 3.34. Seçilen verinin k-nn algoritma tahmini sonuç.....	52
Şekil 3.35. Seçilen verinin k-nn algoritma karmaşıklık matrisi.....	53
Şekil 3.36. Seçilen verinin k-nn algoritma roc eğimi.....	53
Şekil 3.37. Seçilen verinin k-nn algoritma pr eğimi.....	53
Şekil 3.38. Dt algoritma decisiontreeclassifier kütüphane kullanımı.....	54
Şekil 3.39. Seçilen verinin dt algoritma tahmini sonuç.....	54
Şekil 3.40. Seçilen verinin dt algoritma karmaşıklık matrisi.....	55
Şekil 3.41. Seçilen verinin dt algoritma roc eğrisi.....	55
Şekil 3.42. Seçilen verinin dt algoritma pr eğrisi.....	56
Şekil 3.43. Naive bayes algoritma gaussiannb kütüphane kullanımı.....	56
Şekil 3.44. Seçilen verinin naive bayes algoritma tahmini sonuç.....	57
Şekil 3.45. Seçilen verinin naive bayes algoritma karmaşıklık matrisi.....	58

Şekil 3.46. Seçilen verinin naive bayes algoritma roc eğrisi.....	58
Şekil 3.47. Seçilen verinin naive bayes algoritma pr eğrisi.....	58
Şekil 3.48. Catboost algoritma catboostclassifier kütüphane kullanımı.....	59
Şekil 3.49. Seçilen verinin catboost algoritma tahmini sonuç.....	59
Şekil 3.50. Seçilen verinin catboost algoritma karmaşıklık matrisi.....	60
Şekil 3.51. Seçilen verinin catboost algoritma roc eğrisi.....	60
Şekil 3.52. Seçilen verinin catboost algoritma pr eğrisi.....	61
Şekil 3.53. Weka uygulamasında histogram gösterimi.....	61
Şekil 3.54. MEDIC-AI uygulamasında histogram gösterimi.....	62
Şekil 3.55. WEKA uygulamasında k-means kümeleme gösterimi.....	62
Şekil 3.56. MEDIC-AI uygulamasında k-means kümeleme gösterimi.....	62
Şekil 3.57. WEKA uygulamasında Naive Bayes algoritmanın sonuçları.....	63
Şekil 3.58. MEDIC-AI uygulamasında Naive Bayes algoritmanın sınıflandırma raporu.....	63
Şekil 3.59. WEKA uygulamasında ANN algoritmanın sonuçları.....	63
Şekil 3.60. MEDIC-AI uygulamasında ANN algoritmanın sınıflandırma raporu.....	64
Şekil 3.61. Medıc-a1 web uygulamasının giriş ekranı.....	66
Şekil 3.62. Medıc-a1 uygulamasından giriş yaptıktan sonra ilk ekran.....	66
Şekil 3.63. Uygulamadan .csv formatında veri yüklemek.....	66
Şekil 3.64. Dosyanın başarıyla bir şekilde yüklediğini gösteren uyarı mesajı.....	67
Şekil 3.65. Dosya yükleme ve output target label oluşturma.....	67
Şekil 3.66. Veri tabanından tablo çekimi.....	68
Şekil 3.67. Veri tabanından seçilen tablonun gösterimi ve veri içeriği.....	68
Şekil 3.68. Farklı tabloların özellikler ve örneklem sayısı gösterimi.....	69
Şekil 3.69. Kaydetme esnasında oluşturulan yeni cancer_1 adındaki tablo.....	69
Şekil 3.70. Silme esnasında silinen kalphastaliklar_ tablonun silinmesi.....	70
Şekil 3.71 uygulama genel bakış veri analizi.....	70

Şekil 3.72. Boş veri renklendirme gösterimi.....	70
Şekil 3.73. Uygulamada sunduğumuz makine öğrenme algoritmaları.....	71
Şekil 3.74. Yeni gelen verini tahmini için seçimi.....	71
Şekil 3.75. Svm algoritmayla tahmini sonuç gösterimi.....	72
Şekil 3.76. Catboost algoritmanın tahmini sonuçları.....	72



TABLO LİSTESİ

Tablo 1.1. Dünya genelinde tıpta yapay zeka uygulamaları kullanan bazı şirketler [2].....	3
Tablo 2.1. Python programlama dili makine öğrenimi için sunduğu kütüphaneler [46].....	18
Tablo 2.2. Makine öğrenmesi algoritma türleri.....	21
Tablo 3.1. İkili sınıflandırma için bir karışıklık matrisi.....	35
Tablo 3.2. Seçilen verinin ANN algoritmanın sınıflandırma Raporu.....	42
Tablo 3.3. Seçilen verinin ANN algoritmanın sınıflandırma raporu.....	44
Tablo 3.4. Seçilen verinin SVM algoritmanın sınıflandırma raporu.....	47
Tablo 3.5. Seçilen verinin SVM algoritmanın sınıflandırma raporu.....	48
Tablo 3.6. Seçilen verinin K-NN algoritma sınıflandırma raporu.....	52
Tablo 3.7. Seçilen verinin K-NN algoritma sınıflandırma raporu.....	52
Tablo 3.8. Seçilen verinin DT algoritma sınıflandırma raporu.....	55
Tablo 3.9. Seçilen verinin Naive Bayes algoritma sınıflandırma raporu.....	57
Tablo 3.10. Seçilen verinin catboost algoritma sınıflandırma raporu.....	60
Tablo 3.11. Uygulamalar arasında karşılaştırma.....	65
Tablo 4.1. Hastaveri_ adındaki tablonun tahmini sonuçlarının ML algoritmalarının doğruluk oranları.....	74
Tablo 4.2. Cancer_ adındaki tablonun tahmini sonuçlarının ML algoritmaların doğruluk oranları.....	75
Tablo 4.3. Hastaveri_ tablosunda yapılan yeni verinin doğruluk oranları.....	75
Tablo 4.4. Cancer_ tablosunda yapılan yeni verinin doğruluk oranları.....	76

1. GİRİŞ

Büyük veri ve makine öğrenimi, eğlenceden ticarete ve sağlıktan modern yaşamın birçok yönü üzerinde etkili oluyor. Yapay zekâ (AI) uygulamasının teşhisten tedaviye kadar sağlık hizmetlerinin tüm alanlarında önemli gelişmeler sağlayabileceğine dair büyük bir iyimserlik var [1]. Yapay zekâ (AI), bankacılık ve finans, eğitim, e-ticaret ve sağlık gibi geniş bir alan yelpazesini geliştirmek ve ilerletmek için çeşitli biçim ve derecelerde kullanılmıştır. Teknoloji endüstrisinde yapay zekâ, birçok yeni iş yeniliği için önemli bir kolaylaştırıcı olmuştur [2].

Sağlık konusunda dünya genelinde, hastalık tespiti konusunda çokça araştırmalar ve sistemler yapılmaktadır. Yapılan çoğu karar verme destek sistemleri (yapay zekâ veya makine öğrenme ile destekli), gerek sağlık personellerine gerek ise hastalara yardımcı olmayı amaçlanmaktadır. Yapay zekâ algoritmalarının çeşitli görevlerde, örneğin tıbbi görüntüleri analiz etmede veya elektronik tıbbi kayıtlardan (EMR'ler) semptomları ve biyobelirteçleri, hastalığın karakterizasyonu ve prognozu ile ilişkilendirmede, insanlarla eşit veya daha iyi performans gösterdiğine dair birçok örnekler var [3]. Sağlık ekosistemi, yeni nesil sağlık teknolojisinde yapay zekâ destekli araçların önemini fark ediyor. Yapay zekanın sağlık hizmeti operasyonu ve sunumundaki herhangi bir sürece iyileştirmeler getireceğine inanılıyor. Örneğin, AI'nın sağlık sistemine getirebileceği maliyet tasarrufları, AI uygulamalarının uygulanması için önemli bir itici güçtür [4,5].



Şekil 1.1. Yapay zekanın her alanında kullanıldığını gösterimi.

1.1. Amaç, Hedef ve Uygulama

1.1.1. Amaç

Dünya genelinde ölümcül hastalıklardan biri olan ve yaygın olarak görülen rahatsızlıklardan başında gelen hipertansiyon hastalığı ve onun tetikleyebilecek kardiyovasküler hastalıkları, makine öğrenme, derin öğrenme ve veri madenciliği tekniklerini kullanarak bu hastalıkların tespitini ve analizini yapmak için üzerinde farklı veri setleri kullanılabilir bir web uygulaması geliştirmektir. Bu anonim veri setleri üzerine farklı ML algoritmaları deneyerek, çıkan sonuçlarını ML algoritmaları arasında ve anonim veri setleri arasında karşılaştırılması yapılmıştır. Kullanılan anonim veri setlerinin, yapılan uygulama üzerine incelenmesi ve analiz yapılmasını, farklı denemelerin yapılması planlanan bir web üzerine ML destekli karar destek sistemi yapılmıştır.

1.1.2. Hedef

Bu çalışmada web uygulaması üzerinde farklı anonim veri setleri kullanılmıştır. Bu veri setleri üzerine, uygulama üzerine veri analizi, veri temizleme gibi kullanıcıya sunmak, sunulan farklı ML algoritmaları üzerine deneme ve tahminleme sonuçları alabilmek, tahminleme sonuçlarını karşılaştırılması ve grafikler üzerine incelenme ve analiz yapılabilmesi, sunulan farklı grafikler üzerine veri inceleme ve analiz yapılması gibi özelliklerle web üzerine ML destekli bir karar destek sistemi yapılmıştır. Bu web uygulamasında kullanıcı tarafından, seçeneklerden, istenilecek herhangi bir veri seti seçip, ilk önce verinin analizini ve incelemesini web uygulaması üzerine yaparak, daha sonra istenilen bir algoritma seçerek kullanılan anonim veri setleri üzerine ki hastalıkların tahminleme sonuçları alabilecektir ve bu sonuçların grafiklerini ve raporlarını inceleyebilecektir. Bunun doğrultusunda, MEDIC-AI, ML destekli karar destek web uygulaması yapılmıştır.

1.2. Literatür Taraması

Bu tez çalışması ve bu teze bağlı geliştirilen MEDIC-AI ML destekli uygulama, farklı hastalıklar üzerine araştırılmış ve yapılmıştır. Dünya genelinde yapılan araştırmalar ve yazılan makalelere bakılarak, sağlık alanında çok sayıda makine öğrenmesi, büyük veri teknikleri, derin öğrenme gibi uygulamalar yapılmıştır ve halen devam etmektedir.

Yapılan birçok uygulamalar uzman doktorlar tarafından kullanılmaktadır. Tablo 1’de bunlara ilişkin örnekler sunulmuştur.

Tablo 1.1. Dünya genelinde tıpta yapay zekâ uygulamaları kullanan bazı şirketler [2].

Şirket Adı	Amaç	Web Sayfası
AiCure (New York City) Patient-oriented	Hastalar, hastalık ve tedavi arasındaki bağlantıyı daha iyi anlamak için video, ses ve davranışsal veriler kullanır.	www.aicure.com
Aidence (Amsterdam, The Netherlands) Clinician-oriented	Radyologlar için yapay zekâ: akciğer kanserinin tedavisi için teşhisin iyileştirilmesi.	www.aidence.com
Aiva Health Administrative and Operational-oriented	Sesle çalışan ilk bakım asistanı: iletişim için hastaları doğru doktorla bağlantı kurar.	aivahealth.com
Bot MD Clinician-oriente	Bot asistanı: klinik soruları yanıtlar, dikte edilen vaka notlarını yazıya döker, görüntüleri ve dosyaları otomatik olarak düzenler.	botmd.io/en/
Suki Clinician-oriented	Doktorlar için sesli dijital asistan.	suki.ai
Insitro Patient-oriented	Hastalar için ilaç keşfiyle ilişkili süreci ve maliyeti azaltmak için hesaplamalı genomik ile gelişmiş makine öğrenimini kullanır.	insitro.com/

Yazılan makaleler açısından örnek vermek gerekirse;

1 “Nadir hastalıklarda makine öğreniminin kullanımı: bir kapsam incelemesi” adlı makalesinden bakılırsa

“ Hangi nadir ve ölümcül hastalıklarda makine öğrenme teknikleri uygulandığını, hangi algoritmaların, hangi verilerinin kullanıldığını veya hangi tıbbi uygulamaları (ör. Tanı alma) araştırılmıştır., prognoz veya tedavi) incelenmiştir [6].

2 “Çok etiketli sınıflandırma kullanılarak ayakta kan basıncı izleme analizi” adlı makalesinden bakılırsa:

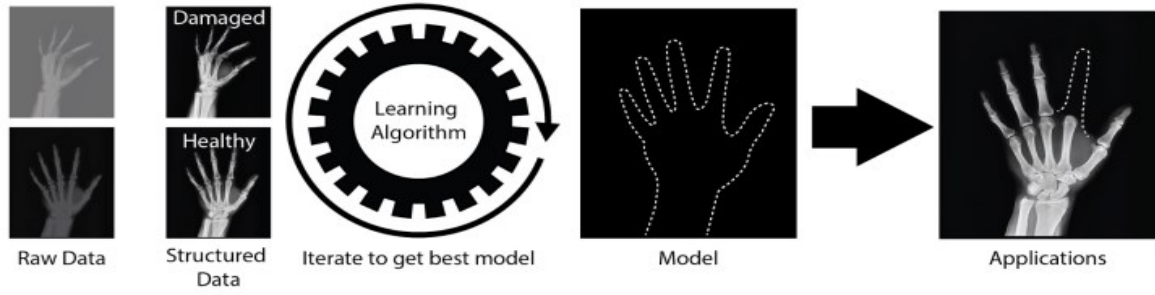
“ Bu çalışmada aynı anda birden fazla etiketle(sınıf) ilişkilendirildiği çoklu-etiket(multi-label) sınıflandırma yöntemleri kullanılmıştır. Bu ilişkilendirmeyi kullanarak ABPM verilerinin analizini iyileştirmeye çalışılmıştır. Bu çalışmada yedi farklı algoritma kullanılarak yeni çoklu-etiket(multi-label) ABPM veri kümesinde deneysel olarak karşılaştırma yapılmıştır [7].

3 “Diyabetli yetişkinler için görüntüleri diyabetik retinopati ve makula ödemi” olarak sınıflandırmak için 128.175 retinal fundus görüntüsü kullanan bir DCNN geliştirdi ve eğitti.

Böyle bir yapay zekâ modelinin varlığının çeşitli avantajları vardır, örneğin [8]:

- Diyabetik retinopatinin otomatik olarak derecelendirilmesi, birçok hastayı daha kısa sürede teşhis etmede verimliliğin artmasını sağlar;
- İkinci görüş göz doktorları olarak hizmet vermektedir;
- Modelin görüntüleri granüler düzeyde inceleme yeteneği nedeniyle diyabetik retinopatinin erken evrelerde saptanması.
- Erişim engellerini azaltan geniş kapsamlı tarama programları.

Başarılı bir yapay zeka algoritması oluşturmak için her veri noktasının algoritma tarafından tanınabilen bir etikete veya ek açıklamaya sahip olduğu verilerle beslenir [9].



Şekil 1.2. Yapay Zekâ algoritmaları çalışma prensipleri.

1.2.1. Tıpta Yapay Zekâ

Yapay zekâ (AI) ve alt disiplinler makine öğrenime ve derin öğrenime gibi, tıpta giderek artan bir ilgi görmektedirler [10]. Araştırmacılar, verileri analiz edebilmek ve birleştirmek için teknolojik gelişmelerden yararlanarak yapay zekânın tıp ve sağlık hizmetinin uygulanma şeklini değiştireceğini belirtmektedirler. Araştırmacılara göre bu vaatler, devrim yaratmak için hastalıkların prognozu ve daha doğru klinik tahmin modellerinin (algoritmalar, araçlar veya kurallar) geliştirilmesi ve uygulanmasıdır [11]. AI, insan sezgisini, karar vermeyi ve nesne tanımayı simüle etmeyi amaçlayan algoritmalar geliştirmek için bilgisayar bilimi, istatistik, psikoloji, sinirbilim, malzeme bilimi, makine mühendisliği ve bilgisayar donanımı tasarımını birleştiren, hızla gelişen disiplinler arası bir alandır [12-13]. Sağlık hizmetlerinde yapay olarak akıllı sistemler büyük miktarda veri ile başlar, bu veriler üzerinde bilgi elde etmek için makine öğrenme algoritmaları kullanılır, bu bilgiler daha sonra tıbbi sistemdeki iyi tanımlanmış bir sorunu çözmek için faydalı bir çıktı üretmek için kullanılır [14]. Şekil 1.3'te bu süreçteki yolculuk görülmektedir.



Şekil 1.3. Bir yapay zekâ modelin geliştirilmesi.

Yapay zekâ (AI), sağlık hizmeti sistemine hızla hakimdir. İnsanların tıbbi uygulamada rutin işleri/görevleri yürüttüğü manuel sağlık sistemini, hastaların ve tıbbi kaynakların yönetimine kadar otomatik hale getirir. Geliştiriciler, görevleri yerine getirmek için yapay zekâ sistemleri oluşturduğunda, sağlık hizmetlerini dijitalleştirmenin teknik zorlukları yeni sorunlar doğurur [15]. AI, hasta bakımında büyük bir gelişme ve bakım maliyetlerinde azalma bir potansiyel sunar. Son zamanlarında nüfus artışı, sağlık hizmetleri alma talebi teşvik etmesi beklenmektedir [16]. Yapay zekâ, erken teşhis için çok önemli ve gereklidir. Kanser gibi hastalıkların daha doğru, güvenilir ve hızlı bir şekilde tespit edilmesi için çeşitli algoritmaları ve metotları kullanılmaktadır. Bağımsız öğrenme sistemi ilişkiyi tespit eder ve bir teşhis önerir [17]. KVH yükünün hafifletilmesine yardımcı olmak için uygun maliyetli ve ölçeklenebilir çözümlere ihtiyaç vardır ve yapay zekâ (AI) kritik bir rol oynayabilir. Tıpta yapay zekanın çoğu uygulamaları, veri türlerinin, sayısal tabanlı (kalp artışı ya da kan basıncı) ya da görüntü tabanlı (bir MRI taramaları veya biyopsi doku örneklerinin görüntüleri gibi). Algoritmalar daha sonra verilerden olasılıkları veya sınıflandırmaları öğrenir veya üretir [18]. Tıbbi uygulamalarda, bir teşhis görevindeki bir algoritmanın performansı, klinik kabiliyetini ve değerini belirlemek için bir hekiminkiyle karşılaştırılır [19]. Sağlık hizmetlerinde, teknoloji ve yapay zekanın (AI) uygulamaları arz ve talep zorluklarından bazılarını çözmeye potansiyeline sahiptir. Bilgi işlem gücü ve veri güvenliği, yapay zekâ ile güçlendirilmiş sağlık sistemleri aracılığıyla sağlık hizmeti sunum modellerini temelden dönüştürmek için sağlık hizmetleri ve teknoloji arasında bir yakınlaşma anını müjdeliyor.[20].

1.3. Uygulamanın Farklılıkları

Yapılan MEDIC-AI web uygulamamız genel amaçlı olarak yapılmıştır. Öyleki kullanıcı uygulamamız herhangi bir veri seti yüklendiğinde, veri görüntüleme ve inceleme, veri temizleme, satır veya sütün silme ve ekleme, verinin grafiksel gösterme, tablo üzerinde seçilen herhangi bir veri satırını sunulan ML algoritmalarıyla tahminleme ve bu tahminlemelerin raporlarını ve grafiksel görüntülerini inceleyebilir ve analiz yapabilmektedir. Ayrıca kullanıcıya da kümeleme ve regresyon algoritmalarını kullanarak, veri seti içerisinde bulunan özelliklerini karşılaştırma ve sunulan grafikler yapısı üzerine inceleme yapmaktadır. Yapılan MEDIC-AI web uygulaması deneme sürümümüzde şu ana kadar altı (6) tahminleme algoritmaları, yedi (7) regresyon

algoritmaları, veri yükleme, tablo silme, tablo özelliklerini gösterme, canlı olarak kullanıcıya sunulan grafiksel türlerini seçerek, grafiksel gösterimi yapabilir hale getirilmiştir.

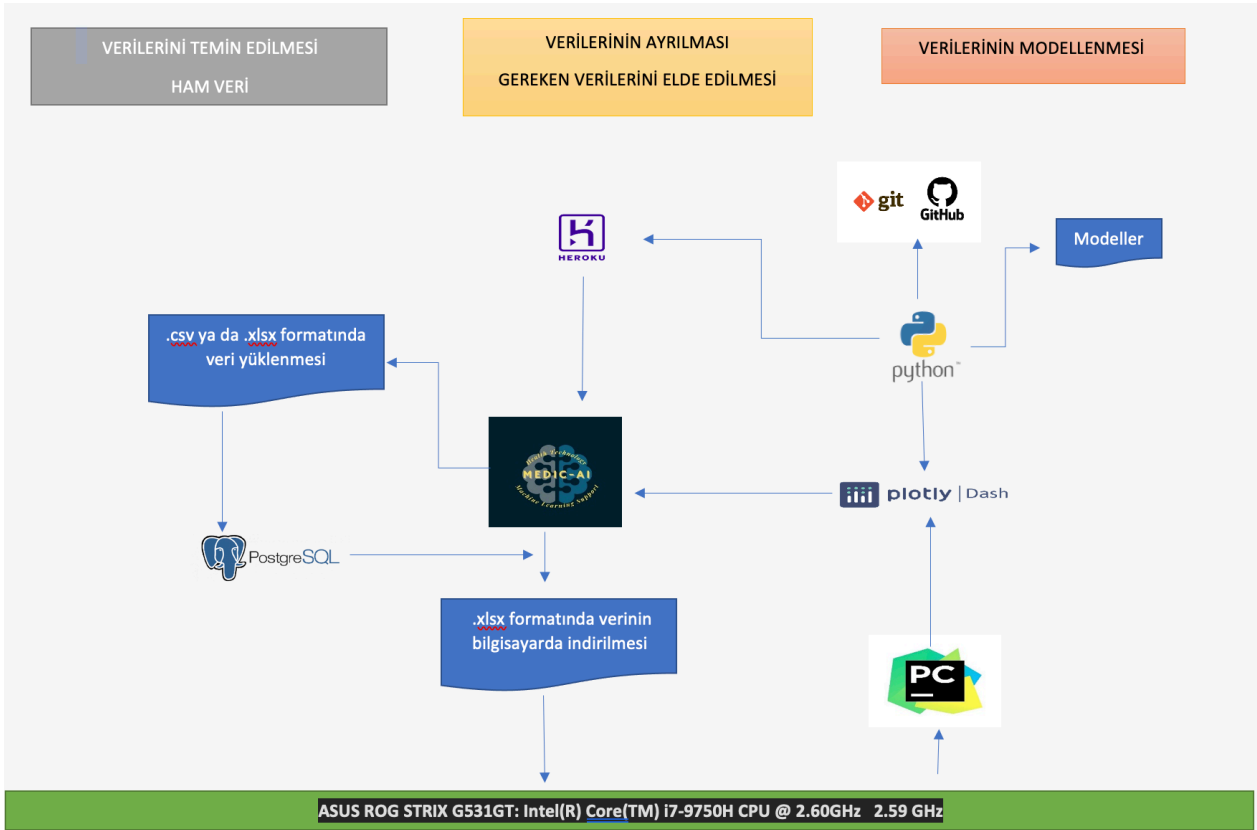
Deneme sürümümüzde sadece hastalık anonimse verileriyle çalışılmış. Onun için şu anda ki deneme versiyonunda uygulamada sadece belli başlı algoritmalar, verilen varsayılan değerlerle çalışmaktadır.

Farklı olarak bu web uygulamada:

- .csv ve .xlsx formatında veri yükleyebilir ve yüklenen veri tabanında yüklenen ismiyle tablo olarak kaydedilir
- İstediği tabloyu veri tabanından çekip, verilerini rahatlıkla tablo üzerine gösterilebilir,
- Tablo üzerine yapılan değişiklikler kaydedilebilir ve farklı bir tablo olarak ana tablo veriyi bozmadan veri tabanında kaydolabilir.
- Yeni kaydedilen tabloyu indirebilir veya istendiğinde silebilir.
- Tablo üzerine yeni veri girişleri yapılabilir.
- Yeni girilen verinin tahminlemesi yapılmaktadır.
- Tahminleme sonuçlarını görebilir ve inceleyebilir.

2. MATERYAL VE YÖNTEM

Sağlık alanında yeni bir çalışma ve karar verme açısından doktor kullanıcılara yardımcı olabileceğini bir uygulama daha kazanmak adına MEDIC-AI ML destekli web uygulaması geliştirilmiştir. Bu çalışmanın esasını, veri analizine dayalı birden fazla makine öğrenme algoritmaları kullanarak tahminleme modelleri oluşturmak ve bu modellerin MEDIC-AI web uygulamasını üzerine yeni gelen bir verinin işlenmesini, analiz edebilmesini ve sunulan 7 (yedi) makine öğrenme algoritmalarından birini seçerek sonucuna ulaşabilmesini oluşturur. Uygulama geliştirme sürecinde, gerekli olan çalışma verilerin elde edilmesini, algoritmalarını oluşturulması ve geliştirilmesidir. Uygulama üzerine verinin içeriğini incelenmesinin, veri analizini yapılmasıdır. MEDIC-AI uygulamasını geliştirmek için yazımsal ve donanımsal materyalleri kullanmak oldukça önemlidir. Şekil 2.1.'de gösterildiği gibi donanım olarak **ASUS ROG STRIX G531GT: Intel Core i7-9750H CPU @ 2.60GHz 2.59 GHz** kullanıldı. MEDIC-AI uygulaması esas olarak **Python 3.10** ve **Flask 2.2** diller üzerine geliştirildi. **Dash by Plotly** programlama ortamı ve IDE olarak **PyCharm Community Edition** kullanıldı. Uygulama **heroku** bulut platformunda alındı.



Şekil 2.1. Tez çalışmasında kullanılan donanımsal ve yazılımsal sistem mimarisi.

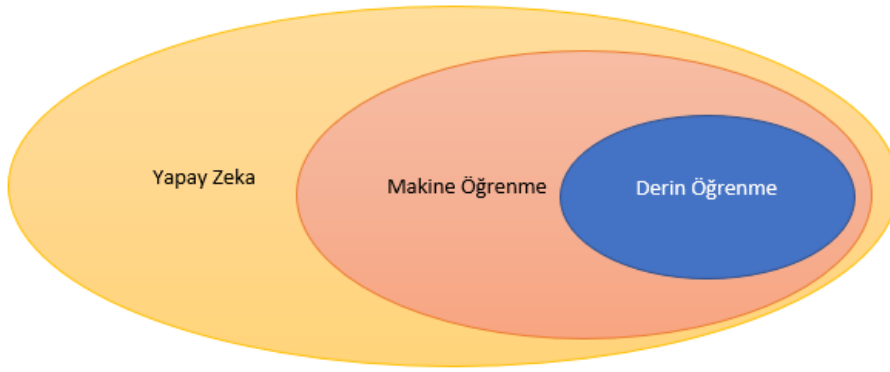
2.1. Yapay Zekâ (AI) ve Makine Öğrenme (ML)

Son zamanlardaki başarılar sayesinde, yapay zekânın (AI), makine öğrenmenin (ML) ve özellikle derin öğrenmenin (DL), birçok araştırma ve endüstriyel alanda en önemli alanlar haline geliyor. Dünya katlanarak büyüyor ve dünya genelinde toplanan verilerin boyutu da öyle. Veriler daha anlamlı ve bağlamsal olarak alakalı hale geliyor, makine öğrenimi (ML) için, özellikle derin öğrenme (DL) ve yapay zekâ (AI) için yeni temeller atıyor ve onları araştırma laboratuvarlarından üretime taşıyor. Yapay zekâ ve makine öğrenimi çok ilişkilidir [21]. Akıllı sistemler, özellikle akıllı programları yapma mühendisliğidir. İnsanların zekasını anlayabilmek için bilgisayarlar programlarını kullanma görevine benzetilmektedir. [22].

Yapay Zekâ (AI) soyut olarak akıl yürütme ve genelleme yapma, bulmacaları çözme, hedeflere nasıl ulaşılacağını planlama, dünyada dolaşma, nesnelere ve sesleri tanıma, konuşma, tercüme etme, sosyal veya ticari işlemler gerçekleştirme, yaratıcı çalışma gibi birden fazla görevi içerir. Eğer çok akıllı bir program yazabilirseniz, örneğin insan benzeri davranışları varsa, yapay zekâ olabilir [23].

Ancak verilerden otomatik olarak öğrenmediği sürece ML değildir:

ML, “deneyimle otomatik olarak gelişen bilgisayar programlarının nasıl oluşturulacağı sorusuyla ilgilenen” bilimdir.



Şekil 2.2. Yapay Zekânın (AI) Makine Öğrenme (ML) ile Derin Öğrenme İlişkisi.

2.2. Yapay Zekâ (AI)

Yapay zekâ (AI), insan zekasının, insan gibi düşünmeye ve davranışlarını taklit etmeye programlanmış makinelerde simülasyonunu ifade eder. [24].

Son yıllarda yapay zekâ (AI) ile ilgili çok fazla çalışmalar ve proje yapılmaktadır.

Yapay zekâ çalışmasında, bilgisayar modelleri düşünceye karşı bir eyleme dayalı olarak farklılaştıran, potansiyelli bir yapay zekâ hedefini veya tanımını araştırırlar [36]:

- İnsani yaklaşım:
 - İnsanlar gibi düşünen modeller
 - İnsanlar gibi hareket eden modeller

- Gerekli yaklaşım:
 - Rasyonel düşünen modeller
 - Rasyonel davranan modeller

2.3. Yapay Zekanın (AI) çalışma mantığı

Yapay Zekâ ile geliştirilmiş sistemler, büyük oranda eğitim verisi alarak, verileri analiz ederek gelecekteki durumlar hakkında tahminlerde bulunmak için kullanarak çalışır [25].

Yapay zekâ programlaması üç beceriye yoğunlaşmaktadır:

- Akıl yürütmek, öğrenmek ve kendini düzeltme.
- Akıl yürütmek: İstenilen sonuca ulaşabilmek için doğru bir algoritma seçmeye yoğunlaşır.
- Öğrenmek: Verileri toplama ve verinin nasıl eyleme geçirilebilir bilgilere dönüştürüleceğine ile ilgili kurallar oluşturmaya çalışır.
- Kendini düzeltmek. Algoritmalar ile sürekli olarak ince ayarlar yaparak en doğru sonuçları vermelerini sağlayabilmektir.

2.4. Makine Öğrenimi (ML)

Makine Öğrenmesi, AI'nin alt kümesidir. Akıllı programlarının insanlar tarafından hiç bir yardım edilmeksizin yeni verilerden otomatik öğrenebileceği ve yeni gelen verilere uyum sağlayabileceği kavramını ifade eder [26]. ML bir algoritmanın özelliklerini ve sonuçlarını ilk eğitim veri kümesinden, özelliklerin sonuçlarla nasıl ilişkili olduğunu ve sonuçların nasıl tahmin edildiğini öğrenir. Yeni giriş verilerde, ML modeli, yalnızca özelliklerle birlikte sunulduğunda sonuçları tahmin edebilmektedir [27].

2.5. Sağlıkta Makine Öğrenimi

Makine öğrenimi (ML) ile yerleşik yeni otonom teknolojilerin sağlıkta önemli ölçüde etkileyeceğini tahmin ediliyor. ML, mümkün olduğunca doğru tahminler yapmaya

odaklanırken, geleneksel istatistiksel modeller değişkenler arasındaki ilişkileri çıkarmayı amaçlar [28].

Sağlık verilerinin bolluğu ve artan hesaplama gücü nedeniyle, makine öğrenimi (ML), sağlık araştırmacılarını, klinik olarak güvenilir tahminler yapmak için veriye dayalı algoritmalar geliştirme konusunda bir keşif sürecine dahil ediyor. Birçok şirketler hem doktorları destekleyen araçları hem de onlardan bağımsız olarak çalışmak üzere tasarlanmış algoritmaları izleyerek makine öğrenimini tıbbi karar alma sürecine dahil etmek için çalışıyorlar.

ML, sağlık hizmetlerinde sonuçları iyileştirmek ve maliyetleri azaltmak için etkili araçlar sağlama potansiyeline sahiptir ve klinik topluluk bu keşiflerin geliştirilmesinde ve değerlendirilmesinde yer almalıdır. Büyük verileri analiz etmek için makine öğrenimi araçları, gelecek nesil doktorlar için temel bir gereklilik olacağını söylenmektedir.

Sağlık hizmetlerinde makine öğrenimi araçlarını etkin bir şekilde kullanmak için, klinik uygulaması ve sağlık hizmeti sunumunda etik gibi bazı sınırlamaların ele alınması ve kilit konuların dikkate alınması gereklidir. ML uygulanması, gerçek klinik sorunlarla ilgili, verilerin ön işlemlerini, model eğitimi ve sistemin iyileştirilmesini gerektiren benzersiz zorluklar da sunmaktadır [29-30].

2.6. Veri Seti

Bu çalışma için farklı anonim veri setleri kullanılmıştır. Yapılan MEDIC-AI uygulaması üzerine veri temizlenmesi yapılmaktadır. Kullanıcı web uygulamasından, PostgreSQL veri tabanından .csv veya .xlsx formatında yüklenen veriler, tablo halinde görüntülenebilir, veri olmayan satırlardan, kullanıcı tarafından daha rahat görünmek için üzerine renklendirmesi yapılmıştır. Yine de kullanıcı tarafından satır veya sütün silinebilir veya eklenebilir. Yüklenmiş olduğu veriyi kullanıcı tarafından, ana veri setini bozulmadan, PostgreSQL veri tabanından kayıt ettiği sürece, o tablo adından, ardından rakamlar ekleyerek (Ör: hastaveri_, hastaveri_1, hastaveri_2.....) istenildiği gibi büyütebilir veya küçültebilir. Sonrasında da üzerine işlem yapılan tabloda excel formatından bilgisayarında indirilebilir hale geliştirilmiştir. Eğer kullanıcı üzerine işlem yaptığı tabloyu silmek isterse, uygulamada tablo adını seçerek bu işlemi

gerçekleştirebilmektedir. Yine de uygulama üzerinde kullanıcı yüklenen veri setinin canlı olarak ne kadar özellikler ve örneklemeler olduğunu da görebilmektedir.

Şekil 2.3. Boş veri renklendirme gösterimi.

Şekil 2.4. Uygulama üzerine silme işlemi.

Şekil 2.5. Kaydetme işlemi yapıldığında veri tabanında oluşturulan yeni tablo.



Şekil 2.6. Özellikler ve Örneklem sayısı gösterimi.

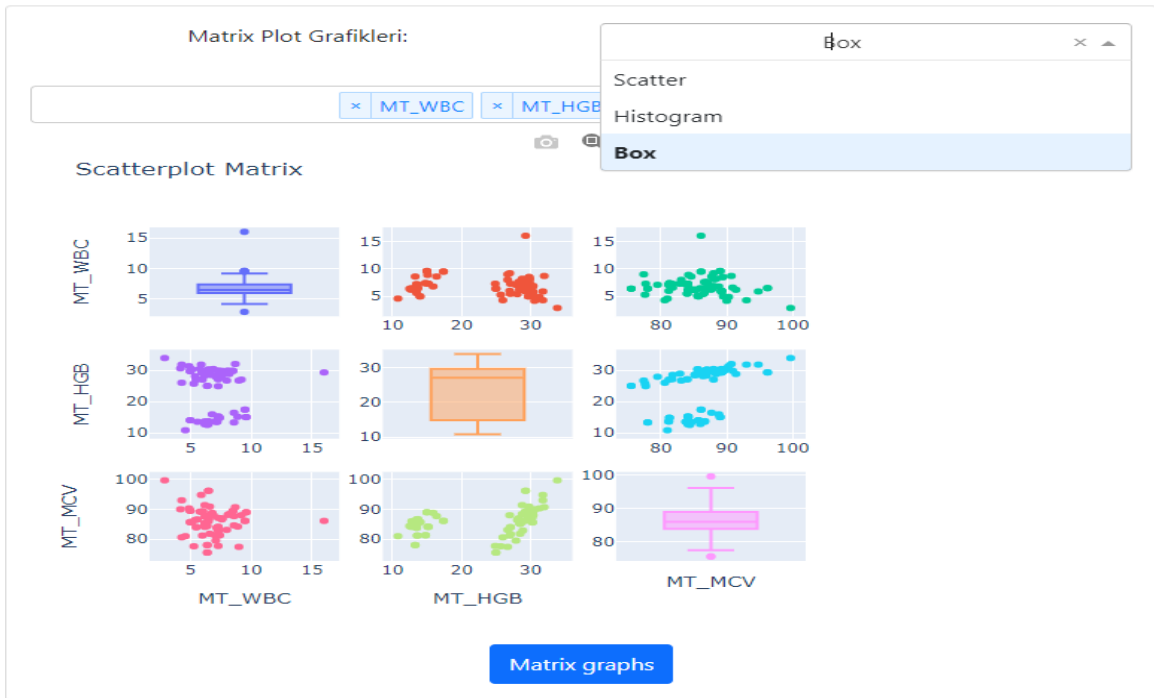
2.6.1. Veri seti analizi

Çalışma kapsamında geliştirilen MEDIC-AI uygulaması, kullanıcı veri analizi yapabilmektedir. Öyleki uygulama kullanıcıya Scatterplot Matrix grafikleri sunmaktadır. Kullanıcı bir input'tan tablodan gelen bütün özelliklerini liste halinde görebilmektedir. Böylece kullanıcı istediği özellikleri seçip bu grafikleri oluşturabilir. Özellikler listesi tabloya göre değişebilmektedir. Öyleki eğer kullanıcı ana tablo yerine işlem yaptığı tabloyu kullanmak istediğinde (Ör: hastaveri_1) liste o tablodaki özellikleriyle güncellenebilmektedir. Ayrıca grafikler de ona göre değişip güncellenebilir. Şekil 2.7'de veri analizi için sunulan grafik örnekleri görülmektedir.



Şekil 2.7. Veri analizi için grafikler

Kullanıcıya Şekil 2.8.'de gösterildiği gibi, 3 (üç) farklı grafik türleri sunulmuştur.



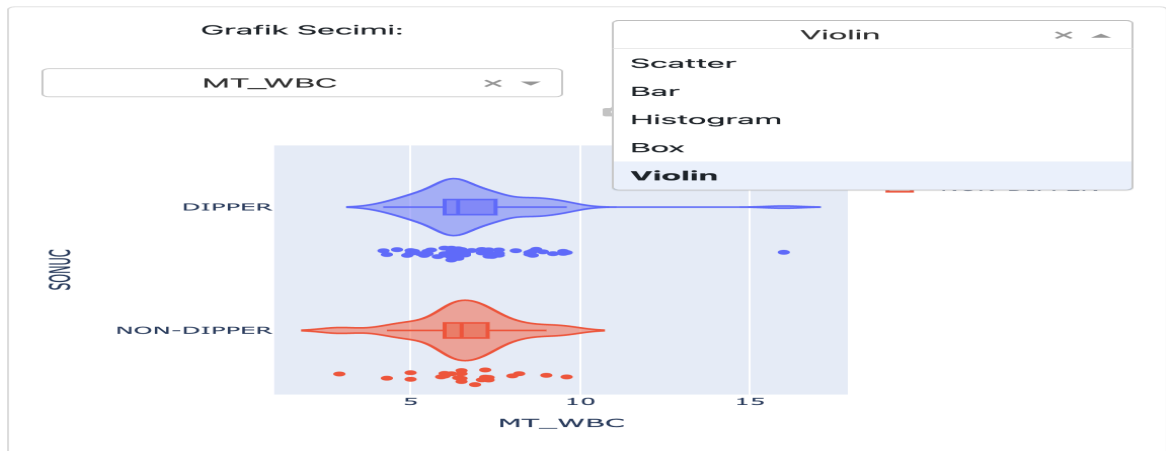
Şekil 2.8. Seçilen özelliklerinin kullanıcıya sunulan scatterplot Matrix grafiklerinin türleri.

Kullanıcı bu grafiği büyütmek istediğinde “Matrix graphs” butonunu tıklayarak, popup şeklinde bu grafiklerini daha iyi ve daha fazla özellik seçerek görebilmektedir.

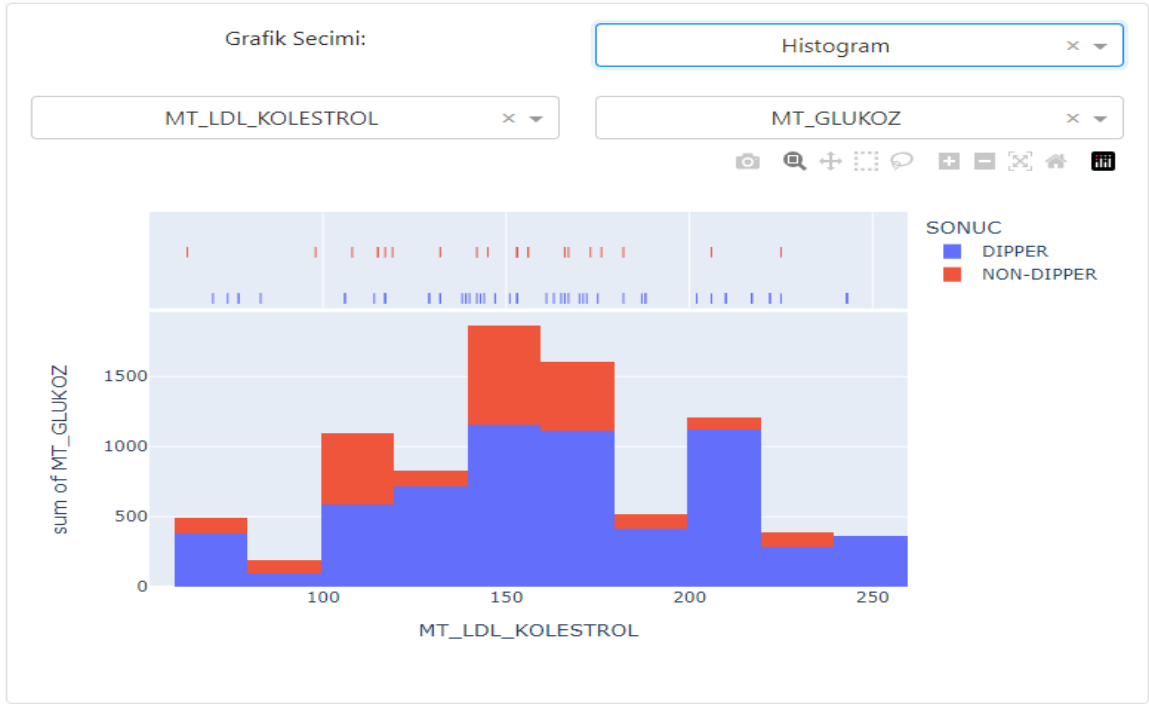


Şekil 2.9. Seçilen özelliklerinin scatterplot Matrix grafiklerinin popup ile gösterimi.

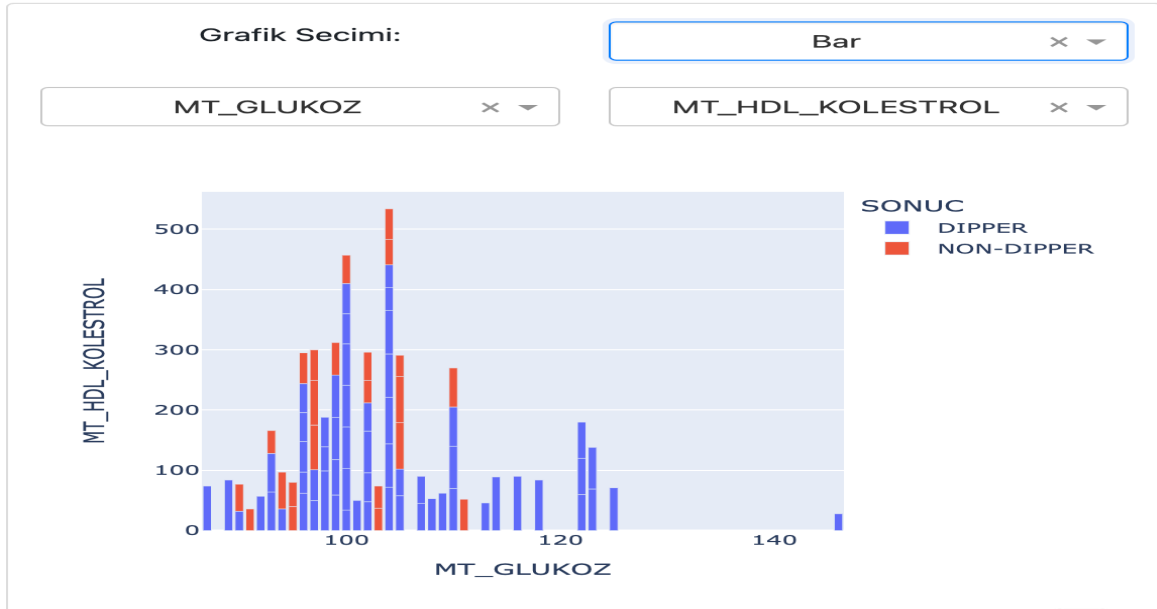
Web uygulaması üzerinde, veri analizi yapabilmek için ve 2 özelliklerini birbirine karşılaştırmak için kullanıcıya grafikler sunulmuştur. Şekil 2.10., 2.11 ve 2.12.’de gösterildiği gibi kullanıcıya “Scatter, bar, histogram” gibi grafik türleri sunulmuştur.



Şekil 2.10. İki özelliklerini karşılaştırmak için kullanıcıya sunulan grafik türleri.



Şekil 2.11. İki özelliğini histogram grafiği ile karşılaştırılması.



Şekil 2.12. İki özelliğini bar grafiği ile karşılaştırılması.

Plotly sayesinde yapılan web uygulamamızda grafiklerini rahatlıkla görülebilir ve analizlerini yapabilmekteyiz.

Yöntem kısmında kullanılan geliştirme ortamları, kütüphaneleri, makine öğrenme algoritmaları gibi detaylı olarak anlatılmıştır.

2.7. Yöntem

MEDIC-AI uygulamasını geliştirme sürecinde, geliştirme ortamı olarak Plotly Dash esas alındı. Birincil programlama dili olarak ise PYTHON programlama dili kullanıldı. Plotly Dash'ın sunduğu geliştirme kolaylıklarını Plotly Dash kısmında detaylı anlatılmıştır.

2.7.1. Python

Python her zaman popüler programlama dillerinden yer almaktadır [31].

Python programla dili, kullanışlı ve genel amaçlı bir dildir. Yapılandırılmış (özellikle prosedürel), nesne yönelimli, işlevsel programlama dahil olmak üzere çoklu programlama paradigmalarını destekler [32].

Python dili, Seaborn, TensorFlow, Keras, Pytorch, Scikit-learn, Pandas ve NumPy kütüphanelerin yardımıyla AI ve ML projelerinde yoğunlukla kullanılmaktadır [33].

Tablo 2.1. Python programlama dili makine öğrenimi için sunduğu kütüphaneler [34].

Kütüphane Adı	Kısaca açıklama
Pandas	Pandas, Python için yüksek performanslı, kolay kullanımlı, veri yapıları ve veri analiz araçları sağlayan bir kütüphanedir.
NumPy	NumPy, Python'a çok boyutlu dizi ve matris işleminin yanı sıra geniş bir üst düzey matematiksel işlemler koleksiyonu ekler. Genellikle bilimsel bilgi işlem için kullanılır ve bu nedenle makine öğrenimi için en çok kullanılan Python Paketlerinden biridir.
Scikit-learn	Bir Python kütüphanesi olan Scikit-Learn, matplotlib, NumPy ve SciPy kütüphanelerin üzerine inşa edilmiştir. Bu Python ML kütüphanesi, veri analizi ve veri madenciliği görevleri için çeşitli araçlara sahiptir.
Keras	Keras, çok popüler bir ML kütüphanesidir ve TensorFlow, veya Theano üzerinde çalışabilen üst düzey bir sinir ağı API'si sağlar.
TensorFlow	Aslen Google tarafından geliştirilen TensorFlow, veri akış grafiklerini kullanarak yüksek performanslı sayısal hesaplama için açık kaynaklı bir kütüphanedir.

MEDIC-AI uygulamasında kullanılan makine öğrenimi algoritmaları **Scikit-learn** Python kütüphanesinin yardımı ile yapılmıştır.

2.7.2. Scikit-learn

Scikit-learn kütüphanesi, Python ile entegre edilmiş, kolay kullanımlı bir arabirimi korumaktadır. Birçok ML uygulamalarını sağlamak için zengin bir ortam kullanır. Scikit-learn kütüphanesi, çeşitli nedenlerle Python'daki diğer ML kütüphanelerinden farklıdır [35-36]:

- BSD lisansı altında dağıtılmaktadır.
- MDP ve pybrain'den farklı olarak verimlilik için derlenmiş kod içerir.
- R ve SHOGUN gibi isteğe bağlı bağımlılıkları olan PYMVPA'nın aksine, kolay dağıtımı kolaylaştırmak için yalnızca numpy ve scipy'ye bağlıdır.
- Veri akışı çerçevesi kullanan PYBRAIN'den farklı olarak zorunlu programlamaya odaklanır.

Tez çalışması kapsamında yapılan MEDIC-AI ML destekli karar destek sisteminin grafikleri, Plotly Python kütüphanesini kullanarak yapılmıştır.

2.7.3. Plotly

Plotly bir Python kütüphanesidir. Birçok istatistiksel, finansal, bilimsel ve 3 boyutlu kullanımını kapsayan 40'tan fazla benzeri olmayan grafik türlerini desteklemektedir. Açık kaynaklı bir grafiksel kütüphanesidir. Dash'i kullanarak saf Python tarafından oluşturulmuş web uygulamalarının, güzel etkileşimli web tabanlı görselleştirmeler oluşturmasına yardımcı oluyor [37].

Plotly, bir veri analitiği ve görselleştirme geliştirme aracıdır. Plotly, çevrimiçi grafik, analiz ve istatistik araçlarının yanı sıra Python, R, MATLAB, Perl, Julia, Arduino ve REST gibi diller için bilimsel grafik kutuphaneleri sunar [38].

2.7.4. Dash

Dash, python, R, Julia ve F#(deneysel) dillerinde hızla veri uygulamaları oluşturmak için düşük kodlu kütüphanesidir. MIT lisansı altında yayınlanan açık kaynaklı bir python kütüphanesidir. Dash, analitik web uygulamaları oluşturmaya yönelik bir kullanıcı arayüzü kütüphanesidir. Python'u veri analizi, veri keşfi, görselleştirme, modelleme, araç kontrolü ve raporlama için kullananlar Dash'i kolaylıkla kullanacaklardır. Etkileşimli veri görselleştirme ile tam yığın bir web uygulaması oluşturmak için gereken tüm teknolojileri ve protokolleri soyutlar [39].

2.7.4.1. Dash-Plotly

Plotly Dash, Python'da bildirim dayalı olarak etkileşimli web uygulamaları oluşturmaları için veri bilimcilerini destekleyen bir kütüphanedir [40]. Plotly'nin dash kutuphanesi, Dash web uygulamaları geliştirmek için bildirim dayalı bir Python arayüzü sağlar. Dash, Ana dash kutuphanesine ek olarak, dash-html-components ve dash-core-components paketleri bir Dash uygulamasının yapı taşlarını oluşturur [41].

1. **dash-html-components**, Dash web uygulamasının düzenini oluşturmak için bir arayüz sağlar;
2. **dash-core-components**, bir Dash uygulamasıyla etkileşimler için kullanılan yaygın araçlardan oluşan bir pakettir ve plotly.py ile yapılan etkileşimli grafikler için bir dcc.Graph bileşeni içerir.

2.7.5. PostgreSQL

PostgreSQL, karmaşık veri yüklerini güvenli bir biçimde depolayan ve birçok özellikle birlikte SQL dilini kullanan, açık kaynaklı bir nesne ilişkisel veri tabanı sistemidir. PostgreSQL, mimarisi, güvenilirliği, veri bütünlüğü gibi birçok özellikler sunan bir veri taban sistemidir [42].

PostgreSQL veri tabanı, yazılımcıların uygulamalar oluşturmalarına, veri bütünlüğünü korumasına ve hataya dayanıklı ortamlar oluşturmaktadır. Veriler, büyük ya da küçük, nasıl olursa olsun verilerinin yönetimine yardımcı olmaktadır. PostgreSQL açık kaynaklı ve ücretsiz olarak kullanıcılara sunulmaktadır [43].

Tez çalışması kapsamında geliştirilen MEDIC-AI ML destekli karar destek sistemi web uygulamamızı, herkesin tarafından ulaşabilmek için, PostgreSQL veri tabanının kurulumunu ve uygulamamızı canlıya alabilmek için Heroku platformu kullanılmıştır.

2.7.6. Heroku

Heroku bulut platformu, farklı programlama dillerini destekleyen bir platformdur (PaaS). Bulut platformlarından ilklerden biri olan Heroku, sadece Ruby programlama dilini desteklediği Haziran 2007'den beri geliştirilmektedir. Bugünlere ise Java, Scsls, Node.js, Python, Clojure, PHP ve Go gibi dilleri desteklemektedir. Bu nedenle, bir geliştiricinin uygulamaları çoğu dilde benzer şekilde oluşturması, çalıştırması ve ölçeklendirmesi için özelliklere sahip olduğu için Heroku'nun çok dilli bir platform olduğu söylenir. Heroku, Salesforce tarafından 2010 yılında satın alındı [44].

2.7.7. Makine öğrenmesi algoritmaları

Bu tez çalışmasında kullanılan veri kümeleri çok sınıflı kategorik değişkenine sahiptir. Bu nedenle yapılan uygulamada tablo 2.3.'te türlere göre gruplandırılan makine öğrenimi algoritmalarından:

1. Sınıflandırma algoritmalarıyla tahminleme ve raporlama yapılmıştır.
2. Regresyon algoritmaları grafiksel olarak yer verilmiştir.
3. Kümeleme algoritmalarından 1 (bir) tanesi de grafiksel olarak gösterilmiştir.

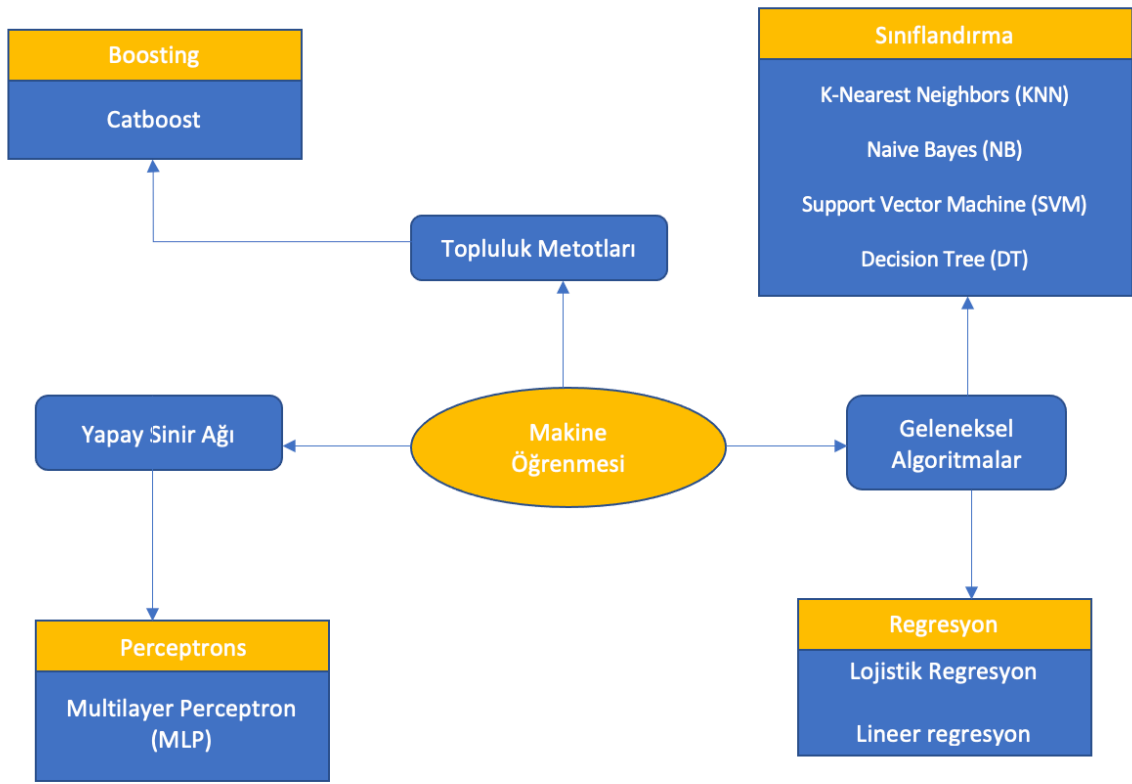
Tablo 2.2. Makine öğrenmesi algoritma türleri.

ML Algoritmaları	Denetimli	Denetimsiz
Kategorik	Sınıflandırma	Birliktelik Kuralı Madenciliği
Sayısal (Devamlı)	Regresyon	Kümeleme Boyutsal Küçülme

Denetimli makine öğrenmesi algoritmalarından, Şekil 2.13'te gösterilen algoritmalar kullanıldı. **Sınıflandırma**, eğitim için alınan verilere dayanarak yeni gözlemler kategorisini tanımlamak için kullanılan bir öğrenme yöntemidir. Sınıflandırmalarda, bir

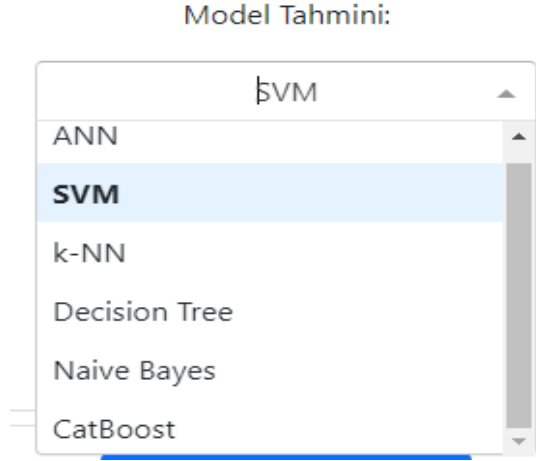
programı ya da sistemi, veriler kümesinden ya da gözlemlerden öğrenir, sonra ise yeni yapılan gözlemi birkaç sınıfa veya gruplara sınıflandırır. Sınıflar, hedef/etiket ya da kategori olarak adlandırılabilir [45].

Regresyon, yaygın olarak kullanılan istatistiksel ve ML aracıdır. Amacı ise, verilerden alınan girdi için sürekli numerik değerler olan sonuç/çıktı yanıtları tahmin etmektir. Sonuç, modelin eğitim aşamasında öğrendiklerine dayalı olacaktır. Regresyon modelleri, girdiler ve çıktılar arasındaki özel ilişkiyi öğrenmek için girdi verisi özelliklerini ve bunlara karşılık gelen sürekli sayısal çıktı değerlerini kullanır. [46]

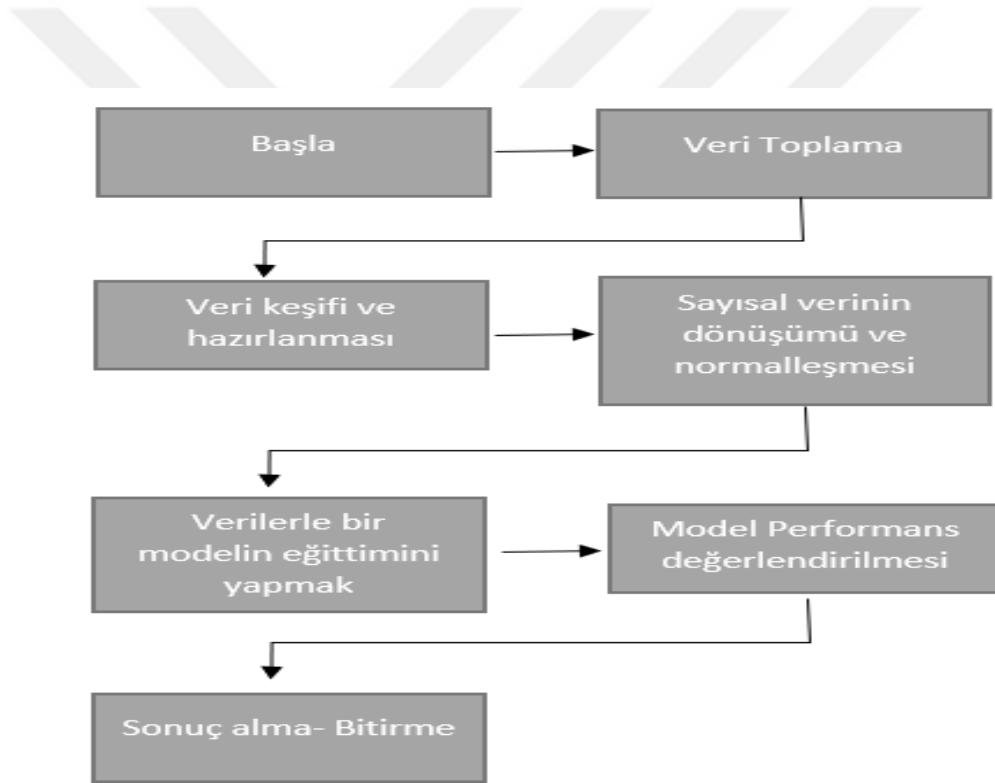


Şekil 2.13. Tez çalışmasında kullanılan makine öğrenmesi algoritmalar.

Tez çalışma kapsamında yapılan uygulamamızda, model tahmini için kullanılan denetimli sınıflandırma algoritmalarımız aşağı belirtilmiştir:



Şekil 2.14. Uygulamamızda kullanılan algoritmalar.



Şekil 2.15. Uygulamada kullanılan algoritmaların genel akış şeması.

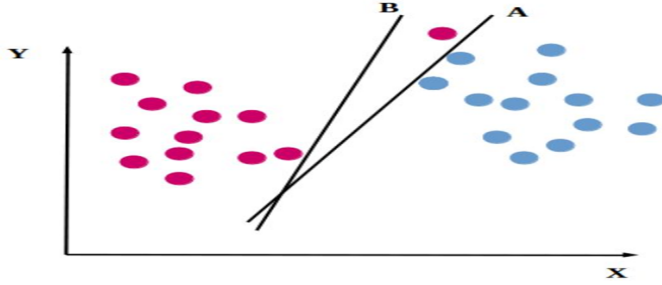
2.7.7.1. Support Vector Machine (SVM – Destek Vektör Makinesi)

Doğrusal olarak ayırt edilebilen ve edilmeyen veri kümesini sınıflandırmaktadır.

Doğrusal olmayan işlemleri, n boyutlu veri kümeleri için $m > n$ olacak biçimde m boyutlu veri kümesine çevrilmektedir.

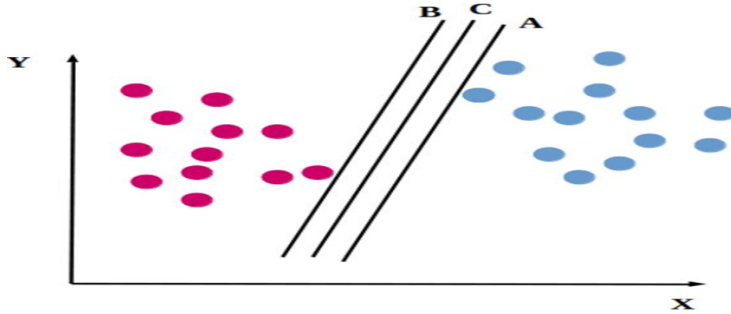
Uygun bir dönüşüm ile her zaman veri iki (2) sınıfa ayrılabilir.

Hiper düzleme en yakın öğrenme verileri destek vektörleri olarak adlandırılır.



Şekil 2.16. SVM' nin Sınır Çizgisinin Belirlenmesi [47].

İlk olarak iki (2) grup arasında sınır çizilir. Sınırların çizileceği yer, iki (2) küme üyelerine en uzak yer olması gereklidir. SVM bu sınırların neye göre ve nereye çizileceğini belirtmektedir.



Şekil 2.17. SVM' nin Sınır Çizgilerinin Paralel Çizgisinin Belirlenmesi [47].

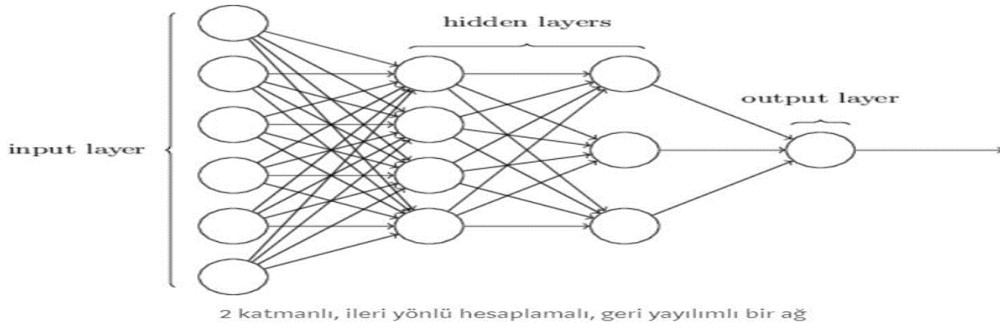
Sisteme giren her bir girdinin özellik çıkarımı yapılmıştır. İki (2) boyutlu düzlemde, bir girdiyi gösterilen farklı bir noktayı seçilmektedir. Bu noktaların sınıflandırılması, çıkarılmış olan özelliklere göre girdinin sınıflandırılması demektir [48].

2.7.7.2. Artificial Neural Network (ANN – Yapay Sinir Ağı)

Nöronlar içermektedir. Nöronlar farklı bir biçimde birbirleri ile bağlanarak ağı getirmektedirler. Bu ağlar, öğrenme, verilerle olan ilişkileri ortaya çıkarma, hafızaya alma kabiliyetine sahiptirler. İnsani düşünce ve gözlemlene becerilerini ortaya çıkartan sorunlara ait çözümler üretirler. Yapay sinir ağlarında da öğrenme nöronlar arasında olan sinaptik bağlantıların ayarlanmasıyla gerçekleşir. ANN birbirine bağlanmış birden fazla işlem elemanından oluşan matematiksel sistemlerdir.

İşlem elemanları gerçek nöronlara karşılık gelmektedir ve ağın içinde birbirleriyle bağlanırlar. Bu yapı, sinir ağlarını oluşturmaktadır. Yapay sinir ağları, girdi verinin, çıktı veriye ilişkilendirilmektedir. Nöronların transfer fonksiyonları bağlantıların ağırlıkları ile etkilenir. [49]

ANN mimari yapısı:

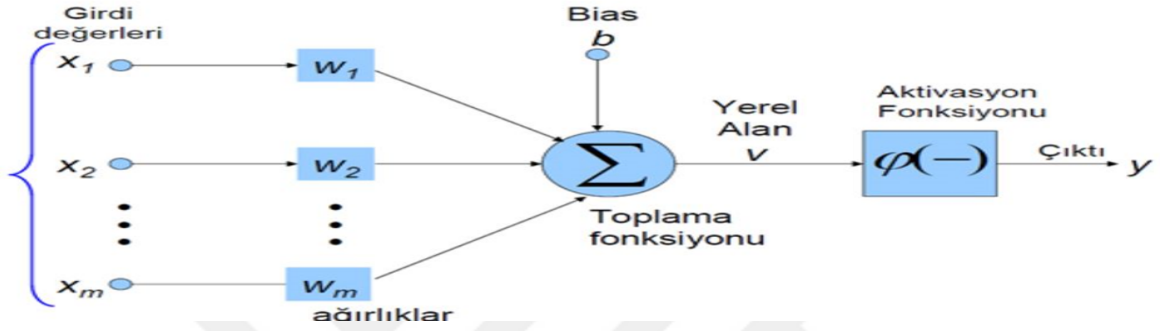


Şekil 2.18. ANN Mimari Yapısı.[50]

- Girdi Katmanı (Input Layer): Bir sinir ağının giriş katmanı, yapay girdi nöronlarından oluşur ve ilk verileri, sonraki yapay nöron katmanları tarafından daha fazla işlenmesi için sisteme getirir. Giriş katmanı, yapay sinir ağları için iş akışının en başlangıcıdır.

- Çıktı Katmanı (Output Layer): Çıktı katmanı nihai sonucu üretmekten sorumludur. Sinir ağlarında her zaman bir çıktı katmanı olmalıdır. Çıktı katmanı, kendisinden önceki katmanlardan gelen girdileri alır, nöronları aracılığıyla hesaplamaları yapar ve çıktıyı hesaplar.

- Gizli Katmanlar (Hidden Layers): Yapay sinir ağlarında, algoritmanın girdisi ve çıktısı arasında, işlevin girdilere ağırlıklar uyguladığı ve onları çıktı olarak bir aktivasyon işlevi aracılığıyla yönlendirdiği gizli bir katman bulunur. Kısacası, gizli katmanlar ağa girilen girdilerin doğrusal olmayan dönüşümlerini gerçekleştirir. Gizli katmanlar, sinir ağının işlevine bağlı olarak değiştiği gibi, katmanlar da ilişkili ağırlıklarına bağlı olarak farklılık gösterebilir. [50].



Şekil 2.19. ANN Çalışma Prensibi [51].

- Ağırlıklar: Bir ANN'deki ağırlıklar, bir girdinin çıktıyı etkilemek üzere dönüştürülmesinde en önemli faktördür. Bu, çıktıyı oluşturmak için bir ağırlığın girdi ile çarpıldığı lineer regresyondaki eğime benzer. Ağırlıklar, her bir nöronun değerini ne kadar güçlü şekilde etkilediğini belirleyen sayısal parametrelerdir. Ağırlıklar pozitif, negatif ya da sıfır de olabilirler.

- Toplama işlevi: Toplama fonksiyonları, ağırlıklı girdileri toplayarak yapay bir nöron hücresinin net girdisini hesaplayan işlevlerdir. En yaygın olanı, girdi ile çarpılan ağırlıkların toplamıdır.

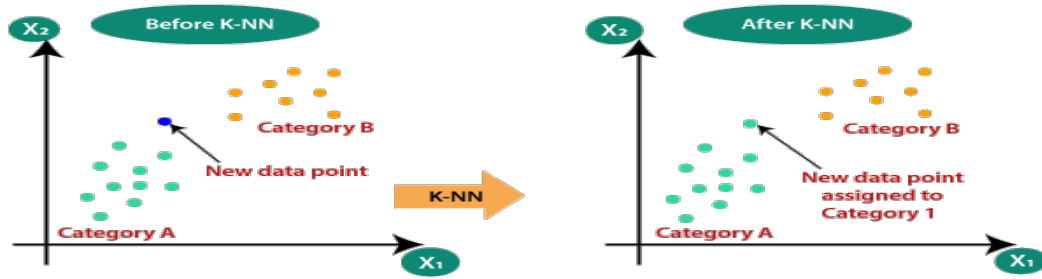
- Aktivasyon Fonksiyonu: Bir nöronun aktif hale getirilip getirilmeyeceğine karar verir. Bu, nöronun ağa girişinin önemli olup olmadığına, daha basit matematiksel işlemler kullanarak tahmin sürecinde karar vermektedir [52].

2.7.7.3. K-Nearest Neighbors (k-NN)

K-En Yakın Komşu (kNN) tekniği, veri madenciliği ve makine öğrenmede yoğun olarak kullanılmaktadır, çünkü basit ama üstün performansı ile çok kullanışlıdır. Sınıflandırma,

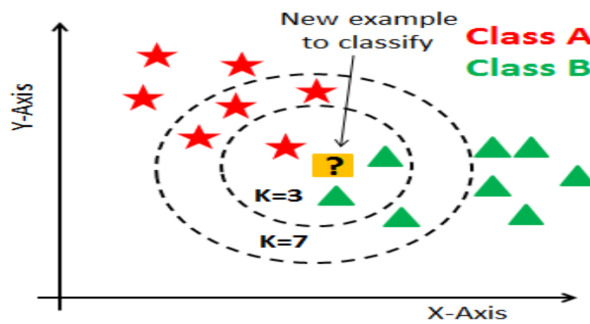
örnek verileri eğittikten sonra test veri noktalarının etiketlerini tahmin etmek için kullanılır [53].

kNN sınıflandırıcı, etiketlenmemiş gözlemleri en benzer etiketli örneklerin sınıfına atayarak sınıflandırmaktadır. Eğitim ve test veri seti için gözlemlerin özellikleri toplanır. kNN algoritması için kaç tane komşunun seçileceğine karar veren k parametresi. Uygun k seçiminin, kNN algoritmasının tanısıl performansı üzerinde önemli bir etkisi vardır. Büyük bir k, rastgele hatanın neden olduğu varyansın etkisini azaltır, ancak küçük ama önemli örüntüyü göz ardı etme riskini de beraberinde getirir. Uygun bir k değeri seçmenin anahtarı, fazla uydurma ve yetersiz uydurma arasında bir denge kurmaktır [54-55].



Şekil 2.20. K-NN Algoritmanın uygulanması [54].

Komşuluğun hacmini ve dolayısıyla yoğunluk tahminlerinin düzgünlüğünü kontrol eden komşuluk parametresi k, en yakın komşu sınıflandırıcısının performansında önemli bir rol oynar. Küçük veya orta derecede büyük örneklem büyüklükleri için, optimum k değerini seçmek için teorik bir kılavuz yoktur. Bu optimum değer, spesifik veri setine bağlıdır ve mevcut eğitim örneği gözlemleri kullanılarak tahmin edilecektir [56].

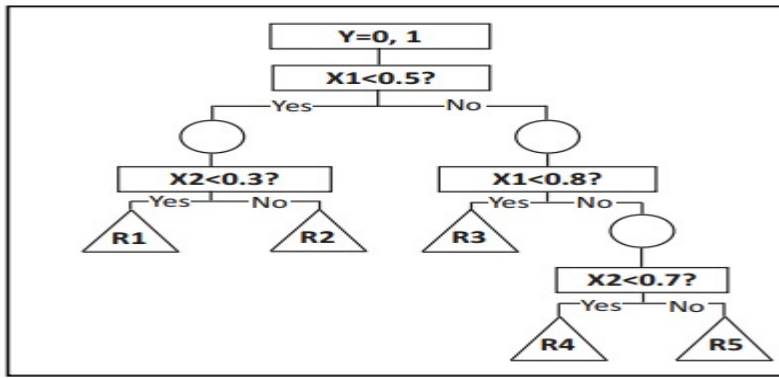


Şekil 2.21. KNN algoritmasının K parametre seçimi [56].

2.7.7.4. Decision Tree (DT)

Karar ağaçları, birden çok değişkene dayalı sınıflandırma sistemleri oluşturmaktadır. Bir hedef değişkeni için tahminleme algoritmaları geliştirmeye yönelik yaygın olarak kullanılan bir yöntemdir. Bu yöntem, bir popülasyonu kök düğüm, iç düğümler ve yaprak düğümlerle ters bir ağaç oluşturan dal benzeri bölümlere ayırır. Algoritma parametrik değildir ve karmaşık bir parametrik yapı dayatmadan büyük, karmaşık veri kümeleriyle etkin bir şekilde başa çıkabilir [57].

Şekil 2.23.'te basit bir karar ağacı algoritmasının nasıl çalıştığını göstermektedir.

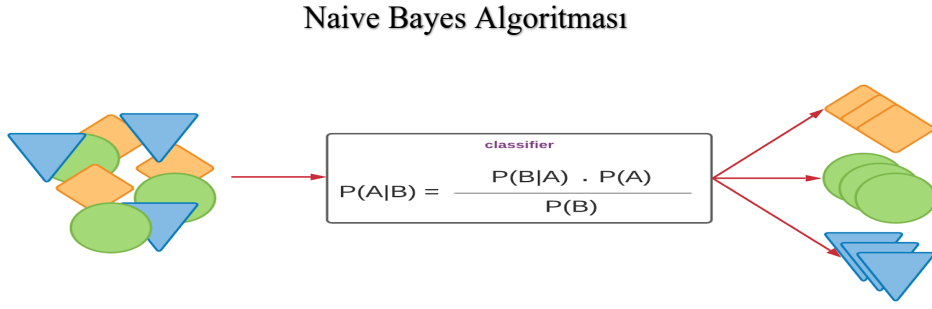


Şekil 2.22. İkili hedef değişken Y'ye dayalı örnek karar ağacı [57].

Karar ağacı, bir sınıfla etiketlenmiş yaprak düğüm veya iki veya daha fazla alt ağaca bağlı bir test düğümünden oluşan bir yapıdır. Bir test düğümü, her bir olası sonucun alt ağaçlardan biriyle ilişkilendirildiği bir örneğin öznelik değerlerine dayalı olarak bazı sonuçları hesaplar. Örnek, ağacın kök düğümünden başlanarak sınıflandırılır. Bu düğüm bir test ise, örneğin sonucu belirlenir ve uygun alt ağaç kullanılarak işlem devam eder. Sonunda bir yaprakla karşılaşıldığında, etiketi örneğin tahmin edilen sınıfını verir [58].

2.7.7.5. Naive Bayes (NB)

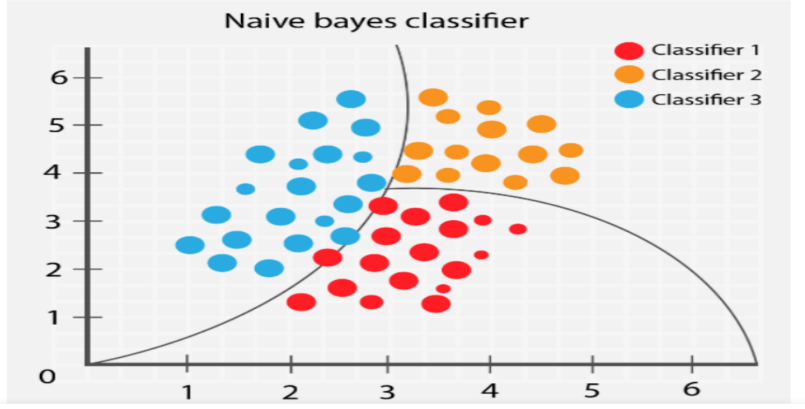
Naive Bayes ML algoritması, Bayes teoremine dayanmaktadır. Görüntülerin özellikleri arasında güçlü bir bağımsızlık sağlar. Bunlar, olasılıksal sınıflandırıcıların arka planından gelir. Bu sınıflandırıcıların uygulanması basittir ve bu nedenle makine öğrenimi sınıflandırmasında yaygın olarak kullanılırlar [59].



Şekil 2.23. Bayes teoremi örnek gösterimi [59].

Naive Bayes, en popüler veri madenciliği algoritmalarından biridir. Verimliliği, öznelik bağımsızlığı varsayımından gelir, ancak bu, gerçek dünyadaki birçok veri setinde ihlal edilebilir. Öznelik seçiminin önemli bir yaklaşım olduğu varsayımı azaltmak için birçok çaba gösterilmiştir. Bununla birlikte, naive Bayes'te öznelik seçimini gerçekleştirmek için yapılan geleneksel çabalar, ağır hesaplama yükünden muzdariptir [60].

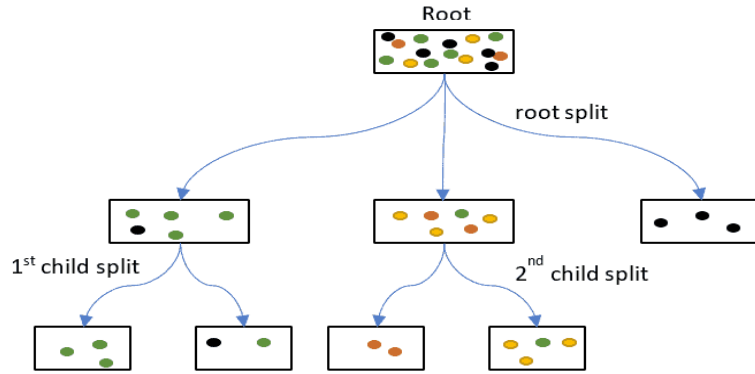
Bir naive Bayes modeli oluşturulması kolaydır, karmaşık tekrarlar parametre tahmini yoktur, bu da tıp bilimi alanında kalp hastalarının teşhisinde özellikle faydalıdır. Naive Bayes sınıflandırıcısı genellikle iyi sonuçlar verir ve çoğunlukla daha karmaşık sınıflandırma yöntemlerinden daha güzel performans sergilediği için yaygın olarak kullanılmaktadır [61].



Şekil 2.24. Naive Bayes sınıflandırması [61].

2.7.7.6. CatBoost

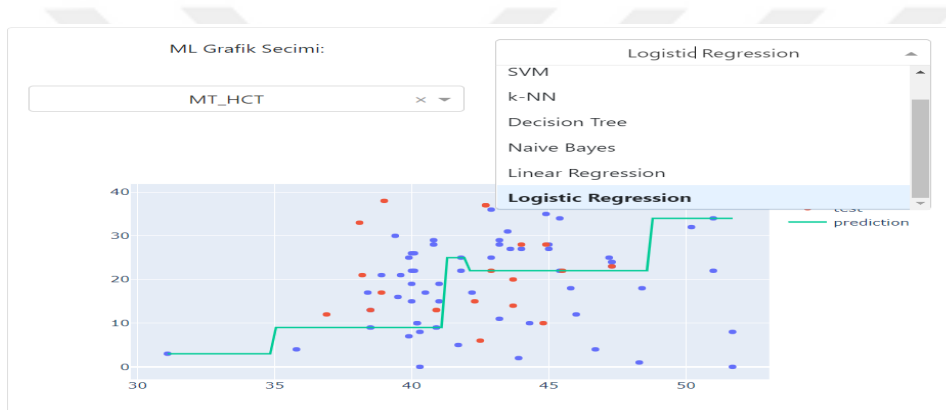
Degrade geliştirme, güçlü bir makine öğrenimi teknolojisidir, CatBoost ise yeni bir tür gradyan geliştirme teknolojisidir. İyi performansın bir sonucu olarak birçok alanda uygulanmıştır; kısa vadeli hava durumu tahmini, Kickstarter kampanyaları için tahmin, sürüş stili tanıma ve diyabet tahmini. Ek olarak, CatBoost yavaş yavaş ekin buharlaşma-terleme tahminine referans olarak kullanılmaktadır [62]. CatBoost, temel belirleyiciler olarak ikili karar ağaçlarını kullanan bir gradyan artırma uygulamasıdır [63].



Şekil 2.25. CatBoost algoritmanın çalışma prensibi [63].

Category Boosting (CatBoost), yakın zamanda açık kaynak haline gelen bir makine öğrenimi algoritmasıdır. CatBoost, kategori özelliklerini ve dizi türü özelliklerini destekleyen, gradyan artırma ağacına dayalı bir makine öğrenimi çerçevesidir. Ancak, bildiğimiz kadarıyla, CatBoost algoritması kullanılarak büyük veri setlerinde orta yaşlı ve yaşlı kişilerin depresyonuyla ilgili faktörlerin sistematik bir tahmini yapılmamıştır [64].

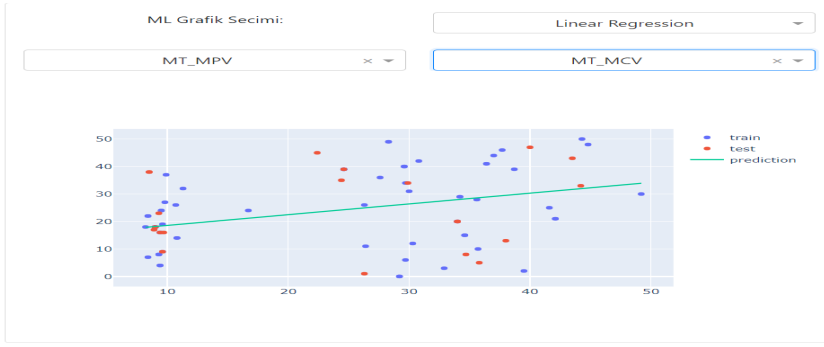
Tez çalışma kapsamında yapılan MEDIC-AI uygulamamızda, özellikler arası tahminleme grafikleri yapabilmek için kullanılan denetimli regresyon algoritmalarımız, yukarıda belirlenen algoritmalarından haricinde kullanılan diğer algoritmalar aşağıdaki belirtilen algoritmalarlardır.



Şekil 2.26. Uygulamamızda kullanılan regresyon algoritmalar.

2.7.7.7. Lineer Regresyon

Lineer regresyon, bir değişkenin değerini, diğer bir değişkenin değerine bağlı olarak tahmininde kullanılır. Tahmin etmek istenilen değişken, bağımlı değişken olarak adlandırılır. Diğer değişkenin değerini tahminini yapmak için kullanılan değişken, bağımsız değişken olarak adlandırılır [65].



Şekil 2.27. Uygulamada linear regresyon gösterimi.

İstatistikte doğrusal regresyon, skaler bir yanıt ile birden fazla açıklayıcı değişkenler aralarındaki ilişkiyi örneklemek için kullanılan doğrusal bir yaklaşımdır. Açıklayıcı değişkenler basit doğrusal regresyon olarak adlandırılır. Bu terim, tek bir skaler değişkenden ziyade birbiriyle ilişkili birçok bağımlı değişkenlerin tahmin edildiği çok değişkenli doğrusal regresyondan farklıdır [66].

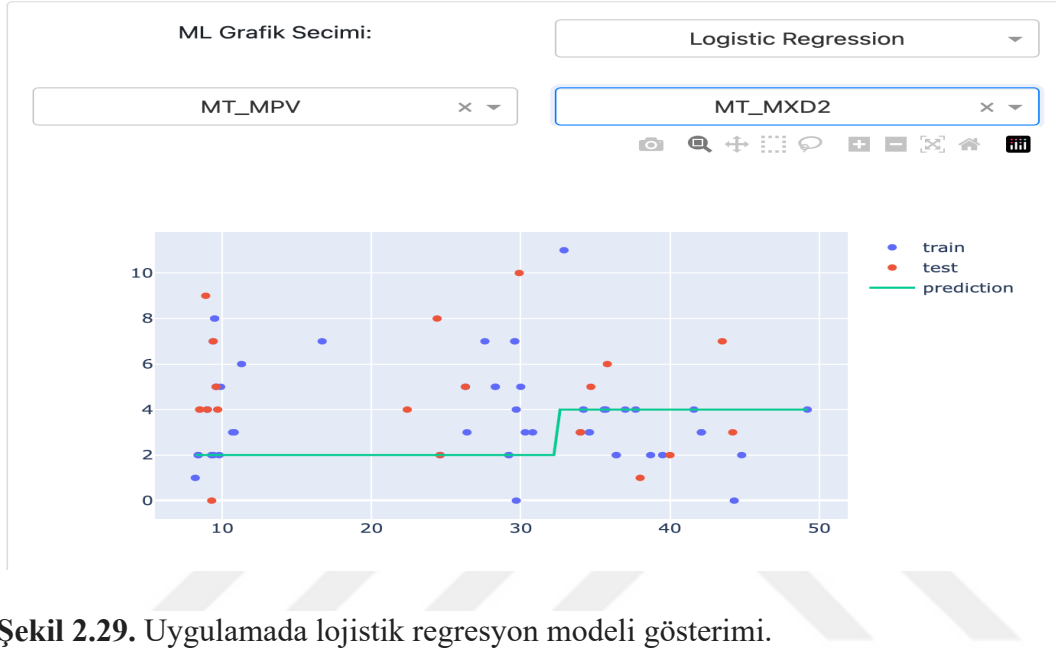
2.7.7.8. Lojistik Regresyon

İstatistikte, lojistik model (veya logit modeli), birden fazla bağımsız değişkenin lineer bir kombinasyonu olan olay için log oranlarını alarak modelleyen istatistiksel bir modeldir. Regresyon analizinde, lojistik regresyon (veya logit regresyon), lojistik modelin parametrelerini (lineer kombinasyondaki katsayılar) tahmin ediyor [67].



Şekil 2.28. Lojistik regresyon modeli genel gösterimi [67].

Lojistik regresyon, belirli bir bağımsız değişken veri kümesine dayalı olarak, oy verilmiş veya verilmemiş gibi bir olayın olma olasılığını tahmin eder. Sonuç bir olasılık olduğu için, bağımlı değişken 0 ile 1 arasında sınırlanmıştır. Lojistik regresyonda, olasılıklara bir logit dönüşümü uygulanır; yani başarı olasılığının başarısızlık olasılığına bölümü. Bu aynı zamanda genellikle log oranları veya oranların doğal logaritması olarak da bilinir [68].



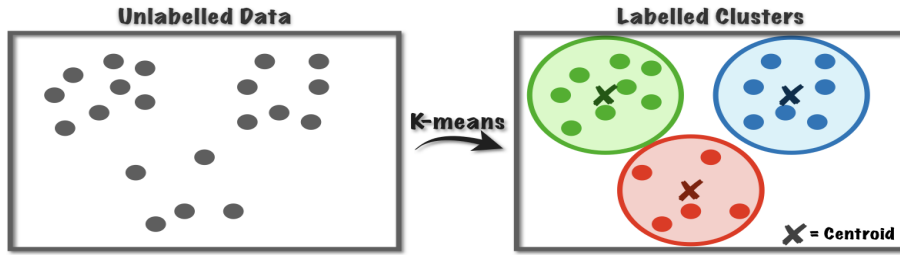
Şekil 2.29. Uygulamada lojistik regresyon modeli gösterimi.

Tez çalışmamızda ve uygulamamızda denetimsiz algoritmalarından, kümelemeye de yer verilmiştir. Uygulamamızda veri analizi için kümeleme algoritmalarından, K-Means kümeleme algoritması kullanılmıştır.

2.7.7.9. K-Means Kümeleme

K-means kümeleme, orijinal olarak sinyal işlemeden gelen, n gözlemi, her gözlemin en yakın ortalamaya sahip kümeye (küme merkezleri veya küme merkezi) ait olduğu ve kümenin bir prototipi olarak hizmet ettiği k kümeye bölmeyi amaçlayan bir vektör niceleme yöntemidir. Bu, veri uzayının Voronoi hücrelerine bölünmesiyle sonuçlanır. k-means kümelemesi küme içi varyansları (karesel Öklid mesafeleri) en aza indirir, ancak daha zor olan Weber problemi olan normal Öklid mesafelerini en aza indirmez: ortalama karesel hataları optimize ederken, sadece geometrik medyan Öklid mesafelerini en aza

indirir. Örneğin, k-ortalamlar ve k-medoidler kullanılarak daha iyi Öklid çözümleri bulunabilir [69].

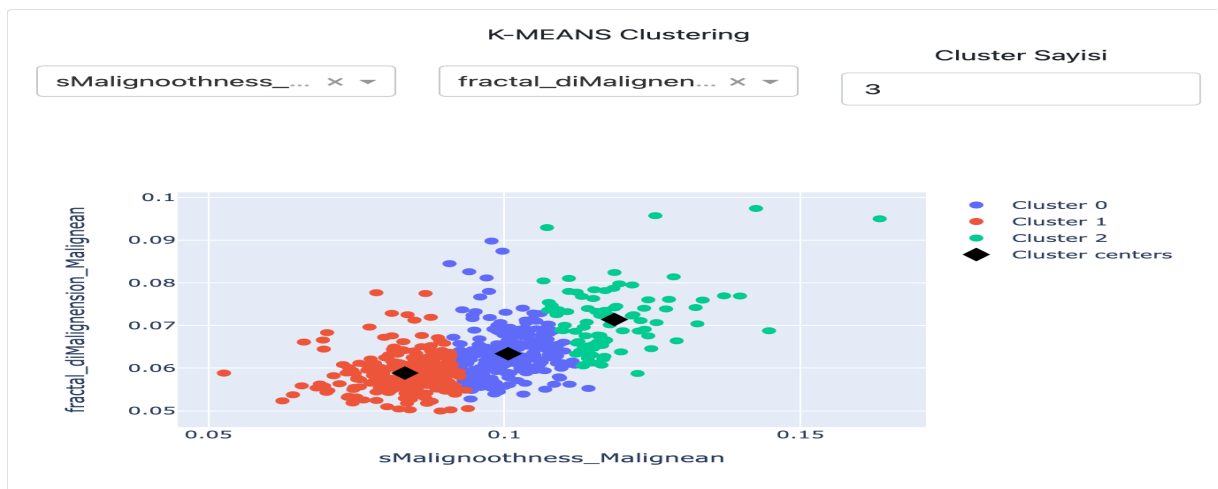


Sekil 2.30. K-means algoritmanın çalışma prensibi [69].

Veri madenciliğindeki K-ortalamlar algoritması, öğrenme verilerini işlemek için her kümenin başlangıç noktası olarak kullanılan rastgele seçilmiş bir ilk grup merkezle başlar ve ardından merkezlerin konumlarını optimize etmek için yinelemeli (tekrarlayan) hesaplamalar yapar [70].

Aşağıdaki durumlarda kümeleri oluşturmayı ve optimize etmeyi durdurur:

- Merkezler sabitlendiğinde- kümeleme başarılı olduğu için değerlerinde bir değişiklik olmadığında.
- Tanımlanan iterasyon sayısına ulaşılmıştır.



Şekil 2.31. Uygulamada K-Means kümeleme algoritma gösterimi.

3. BULGULAR

3.1. Performans Ölçütleri

Makine öğrenimi alanında ve özellikle istatistiksel sınıflandırma probleminde, hata matrisi olarak bilinen karışıklık matrisi, tipik olarak, denetimli bir öğrenme algoritmasının performansının görselleştirilmesine olanak tanıyan tablo düzenidir (denetimsiz öğrenmede genellikle eşleştirme matrisi). Her bir satırı gerçek bir sınıfın örneklerini temsil ederken, her sütun tahmin edilen bir sınıftaki örnekleri temsil eder veya bunun tersi de geçerlidir. Bu isim, sistemin iki sınıfı karıştırıp karıştırmadığını görmeyi kolaylaştırmasından kaynaklanmaktadır [71].

Karışıklık matrisi, sınıflandırma problemlerini çözerken kullanılan meşhur bir ölçüdür. İkili sınıflandırmalarda olduğu kadar çok sınıflı sınıflandırma problemleri için de uygulanabilir. [75]

İkili sınıflandırma için bir karışıklık matrisi örneği aşağıdaki tabloda gösterilmektedir.

Tablo 3.1. İkili sınıflandırma için bir karışıklık matrisi.

Gerçek Değer	TAHMİN EDİLEN DEĞER	
	Negatif	Pozitif
Negatif	TN	FP
Pozitif	FN	TP

True Positive (TP):

Pozitif değer. Pozitif olarak tahmin edilmiş. Doğru tahminleme.

False Positive (FP):

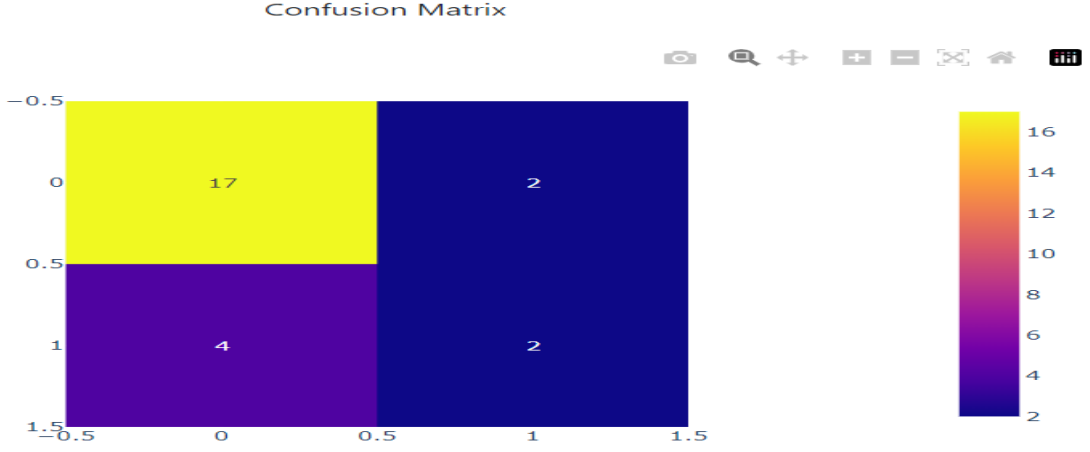
Negatif değer. Tahminleme pozitif olarak yapılmıştır. Yanlış tahminleme.

False Negative (FN):

Pozitif bir değerdir ama tahminleme sürecinde negatif olarak tahminlenmiştir. Yanlış tahminleme.

True Negative (TN):

Negatif bir değer ve negatif olarak tahmin edilmiş. Doğru tahminleme.



Şekil 3.1. Uygulamada karmaşıklık matris gösterimi.

Karışıklık matrisi, doğru sınıflandırılmış TP değerlerini, başka bir sınıfta olması gerekirken ilgili sınıfta yer alan FP değerlerini ve ilgili sınıfta olması gerekirken başka bir sınıfta yer alan FN değerlerini ve diğer sınıfta doğru sınıflandırılmış TN değerlerini temsil eder. Bu değerlere göre sınıflandırma için en sık kullanılan performans metrikleri, doğruluk (ACC), kesinlik (P), duyarlılık (Sn), özgüllük (Sp) ve F-skor değerleridir. Bu performans metriklerinin karışıklık matrisindeki değerlere göre hesaplanması aşağıdaki bölümlere gösterilmiştir [76].

3.1.1. Accuracy / Doğruluk / Kesinlik

Bir ML algoritmanın keskinliği, doğru sınıflandırılmış hasta olanların toplam sayısına oranı olarak temsil edilir [77].

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

Şekil 3.2. Doğruluk formula gösterimi [70-74].

3.1.2. Precision / Hassasiyet

ML algoritmanın hassasiyeti, hastalığı olan ve doğru bir şekilde sınıflandırılmış hastaların, hastalıklı olduğu tahminlenen toplam oranını olarak temsil edilir [77].

$$PPV = \frac{TP}{TP + FP} = 1 - FDR$$

Şekil 3.3. Hassasiyet veya pozitif tahmin değeri (PPV) [70-74].

3.1.3. Recall / Duyarlılık / Geri Çağırma / Sensitivity

Geri çağırma ölçütü, doğru bir şekilde sınıflandırılmış hastaların, gerçekte hastalığı olan toplam sayısına bölünmesiyle elde edilen oranını tanımlanmaktadır [77].

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN} = 1 - FNR$$

Şekil 3.4. Duyarlılık, çağırma, isabet oranı veya gerçek pozitif oran (TPR) [70-74].

Geri çağırmanın arkasındaki algı, kaç hastanın hastalığa sahip olarak sınıflandırıldığıdır. Geri çağırma aynı zamanda duyarlılık olarak da adlandırılır [77].

3.1.4. F1-Score

F1-score F Ölçütü olarak adlandırılmaktadır. F1 puanı, geri çağırma veya kesinlik aralarındaki dengeyi belirler [77].

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$$

Şekil 3.5. F1-score Hassasiyet ve duyarlılığın harmonik ortalamasıdır [70-74].

3.1.5. Özgüllük / Specificity

Özgüllük, toplam gerçek negatif örneklerin içindeki negatif değerlerin oranını verir. Başka bir deyişle, doğru tespit edilen gerçek negatif vakaların oranıdır [78].

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

Şekil 3.6. Özgüllük, seçicilik veya gerçek negatif oranı (TNR) [70-74].

3.1.6. Root Mean Square Error (RMSE) / Ortalama karesel hata karekökü

RMSE, bir model tarafından tahmin edilen değerler ile gerçekte gözlemlenen değerlerle farkların ölçüsü olarak tanımlanır. Denklemden gösterilen N sayısı, gözlem sayısıdır.[78].

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Predicted_i - Actual_i)^2}{N}}$$

Şekil 3.7. RMSE denklem gösterimi [70-74].

3.1.7. Balanced Accuracy / Dengeli Doğruluk

Dengeli doğruluk, ikili bir sınıflandırıcının, iyi olduğunu değerlendirirken kullanılacak metriktir. Özellikle, sınıflar dengesiz olduğunda, iki sınıftan biri diğerinden çok daha sık görüldüğünde kullanışlıdır. Bu durum, anomali tespiti ve bir hastalığın varlığı gibi birçok ortamda sıklıkla görülür. Dengeli doğruluk, daha yaygın olarak kullanılan iki ölçüte dayanır: **duyarlılık** (gerçek pozitif oranı veya geri çağırma olarak da bilinir) **TPR** ve **özgüllük** (gerçek negatif oranı veya yanlış pozitif oranı olarak da bilinir) **TNR** [79].

$$BA = \frac{TPR + TNR}{2}$$

Şekil 3.8. Dengeli doğruluk formula gösterimi [70-74].

Dengeli Doğruluk, ikili ve çok sınıflı sınıflandırmada kullanılmaktadır. Duyarlılık ve özgüllüğün aritmetik ortalamasıdır. Durumu dengesiz verilerle uğraşırken, hedef sınıflardan biri diğerinden çok fazla görüldüğünde kullanılır [80].

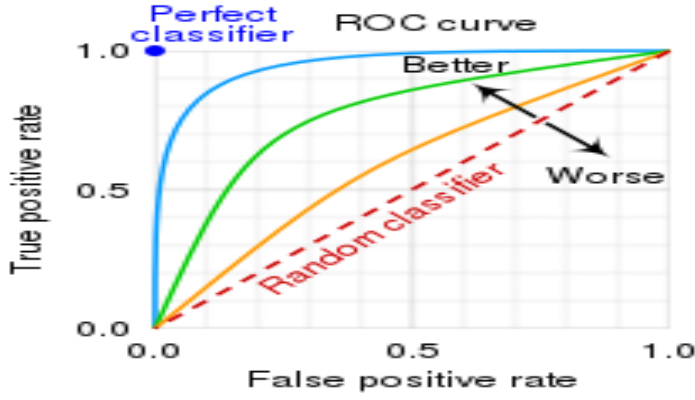
3.1.8. ROC ve PR Eğrileri

İkili (iki sınıflı) sınıflandırma tahmini modelleme problemleri için olasılıksal tahminin yorumlanmasına yardımcı olan iki tanılama aracı ROC Eğrileri ve Kesinlik-Tekrar Çağırma (Precision-Recall) PR eğrileridir [81]. Bununla bağlantılı olarak, ROC eğrisi altındaki alan (AUC, diğer adıyla AUROC) ve Kesinlik-Tekrar Çağırma eğrisi altındaki alan (AUPRC, diğer adıyla ortalama hassasiyet), ROC ve PR eğrilerini tek sayılarla özetleyen ölçütlerdir [82].

3.1.8.1 ROC Eğrisi

ROC eğrisi PR eğrisine benzer. Kesinlik ve geri çağırma yerine, Doğru Pozitif Oranını, Yanlış Pozitif Oranına karşı çizeriz. TPR, geri çağırmanın başka bir adıdır (duyarlılık olarak da adlandırılır). FPR (serpinti olarak da bilinir ve TNR, diğer bir deyişle özgüllük olarak da bilinir), toplam negatif sayısı (yanlış pozitifler artı gerçek negatifler) üzerinden yanlış pozitiflerin sayısı olarak tanımlanır [83].

- TPR geri çağırmadır: içeri giren tüm doğru değerlerden kaç tanesini başarılı bir şekilde geri çıkardık.
- FPR, örneğimizi yanlış bir pozitif ile kirletme şansımızdır.

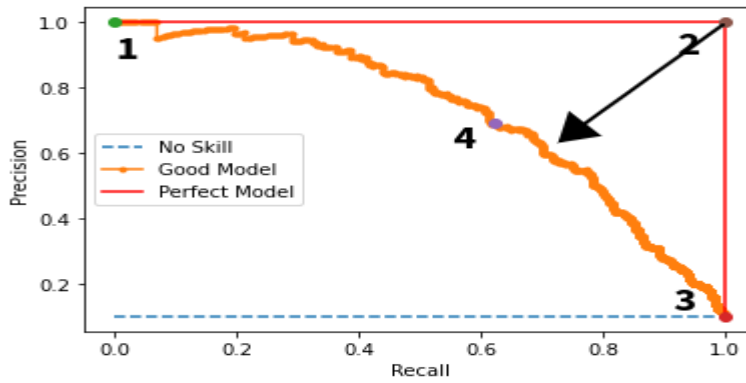


Şekil 3.9. ROC eğrisi grafiksel gösterimi [83].

3.1.8.2. Kesinlik-Tekrar Çağırma (Precision-Recall) PR Eğrisi

Hassasiyet-Geri Çağırma eğrisi tam olarak görüldüğü gibidir. Kesinlik (toplam pozitif sayısını üzerinden doğru pozitiflerin sayısı - doğru pozitifler ve yanlış pozitifler bir arada) ve geri çağırma (toplam gerçek doğru değer sayısını üzerinden doğru pozitiflerin sayısı - doğru pozitifler ve yanlış negatifler bir arada) arasındaki ilişkiyi görmemizi sağlar [83].

- Kesinlik, bildirilen tüm başarılarımız içinde kaç tane gerçek başarı olduğunu sorar
- Geri çağırma, içeri giren tüm gerçek değerlerden kaç tanesini başarılı bir şekilde geri çıkardığımızdır.



Şekil 3.10. PR eğrisi grafiksel gösterimi [83].

- ROC Eğrileri, farklı olasılık eşikleri kullanan bir tahmin modeli için gerçek pozitif oran ile yanlış pozitif oran arasındaki dengeyi özetlemektedir [81].
- Kesinlik-Tekrar Çağırma eğrileri, farklı olasılık eşikleri kullanan bir tahmin modeli için gerçek pozitif oran ile pozitif tahmin değeri arasındaki değiş tokuşu özetler [81].
- ROC eğrileri, gözlemler her bir sınıf arasında dengeli olduğunda uygun olurken, hassasiyet-tekrar çağırma eğrileri dengesiz veri kümeleri için uygundur [81].

3.2. Analiz Sonuçları

Bu çalışmada farklı anonim veri setleri kullanılmıştır. MEDIC-AI uygulamasından veri içeriği ve analizi, gereken silmeler ve eklemeler yapıldıktan sonra, tahminleme kısmında, uygulamada kullanılan makine öğrenme algoritmalarını kullanarak tahminleme sonuçları elde edilmeye başlanmıştır. Elde edilen tahminleme sonuçları, aşağıda kullanılan algoritmayla beraber, karmaşıklık matrisi, sınıflandırma raporları, ROC ve PR eğrileriyle gösterilmiştir. Uygulamada eğitim ve test veri dağıtımı %70 eğitim, %30 test varsayılmıştır. Sistemde yüklenmiş farklı tablo verilerden üzerinde veriler herhangi bir veri seçerek tahminleme sonuçları elde edilmiş ve raporlanmıştır.

3.2.1. Yapay Sinir Ağları (Artificial Neural Network ANN)

ANN algoritması için %70 eğitim ve %30 test verileri olarak veri tablosu dağıtımı yapılmıştır. Sklearn kütüphanesinden **MLPClassifier** kütüphanesi kullanılmıştır.

```
ANN_numeric_data_train, ANN_numeric_data_test, ANN_target_data_train, ANN_target_data_test = train_test_split(
    numeric_data, targetData, test_size=0.30)

scaler = StandardScaler()
scaler.fit(ANN_numeric_data_train)

ANN_numeric_data_train = scaler.transform(ANN_numeric_data_train)
ANN_numeric_data_test = scaler.transform(ANN_numeric_data_test)

mlp = MLPClassifier(hidden_layer_sizes=(50, 50, 50))
model_ANN = mlp.fit(ANN_numeric_data_train, ANN_target_data_train)
```

Şekil 3.11. Uygulamada MLPClassifier kütüphane kullanımı.

Rastgele olarak **cancer_** tablosu üzerine bir veri seçildi ve “Tahmin Yap” butonunu basarak sonuçları elde edilmiş. Seçilen verinin sonuçlar aşağıdaki gibidir.

Model Tahmini:

ANN ▼

SONUC:

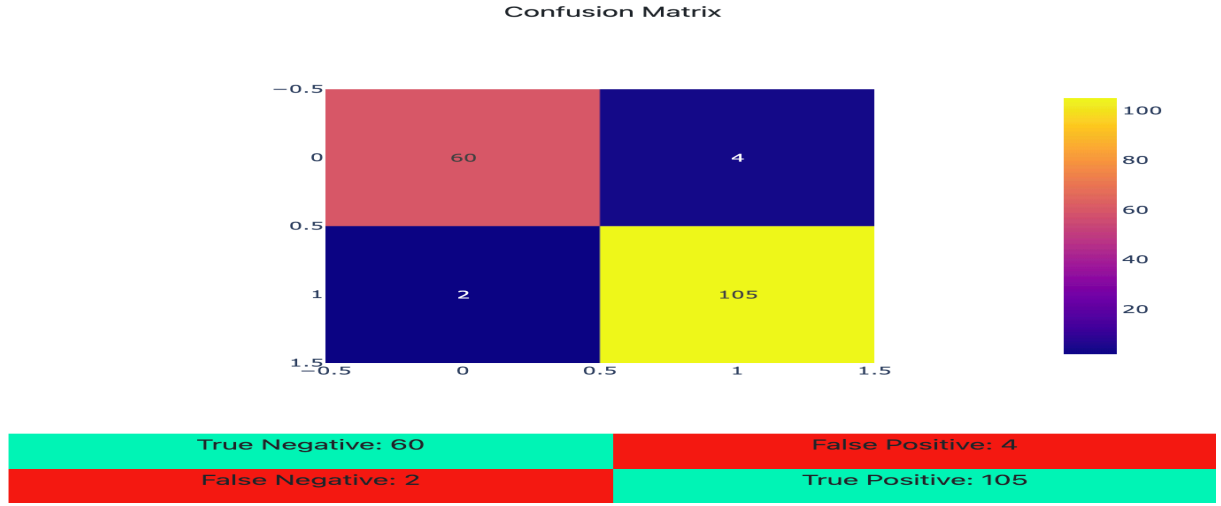
Malign

Tahmin Yap

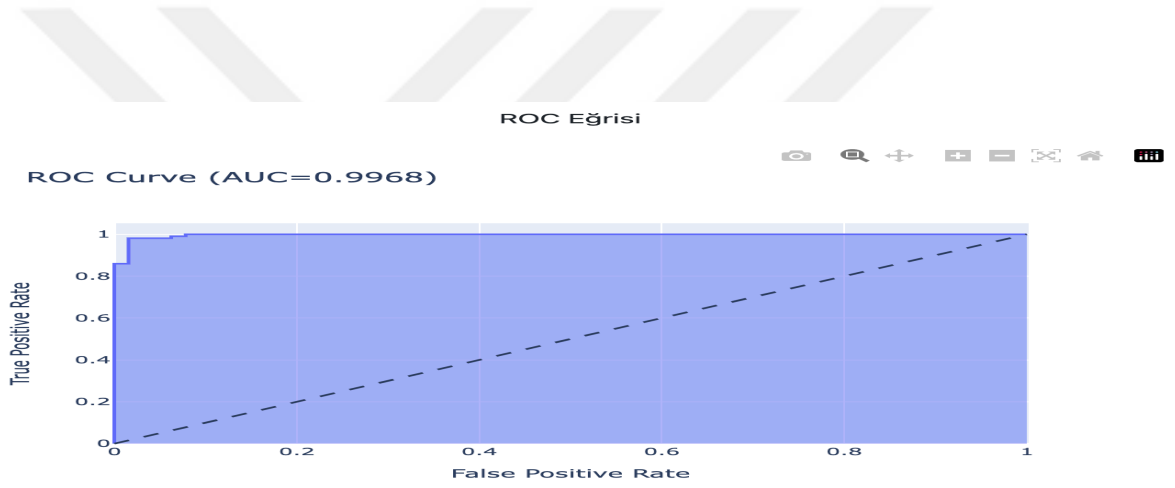
Şekil 3.12. Seçilen verinin ANN tahmini sonuç.

Tablo 3.2. Seçilen verinin ANN algoritmanın sınıflandırma Raporu.

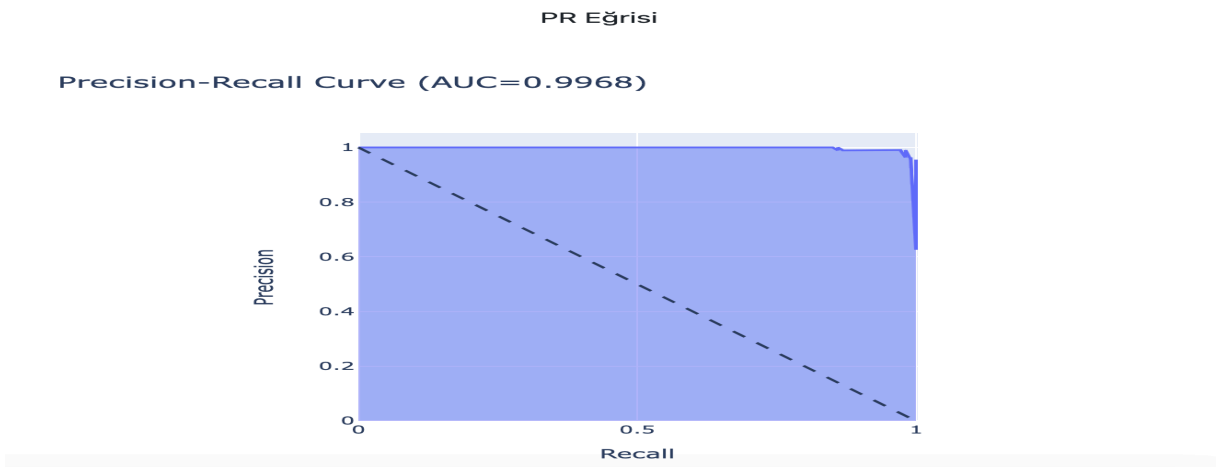
İndex	Precision	Recall	F1-score	Support
1	0,9677	0,9375	0,9523	64
2	0,9633	0,9813	0,9722	107
accuracy	0,9650	0,9650	0,9649	0,9650
macro avg	0,9655	0,9594	0,9623	171
weighted avg	0,9650	0,9650	0,9647	171



Şekil 3.13. Seçilen verinin ANN algoritmanın karmaşıklık matrisi.

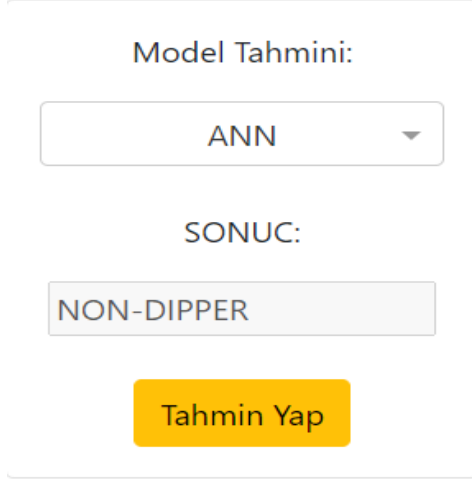


Şekil 3.14. Seçilen verinin ANN algoritmanın ROC eğimi.



Şekil 3.15. Seçilen verinin ANN algoritmanın PR eğimi.

Rastgele olarak **hastaveri_** tablo üzerine bir veri seçildi ve “Tahmin Yap” butonunu basarak sonuçları elde edilmiş. Seçilen verinin sonuçlar aşağıdaki gibidir.



Model Tahmini:

ANN

SONUC:

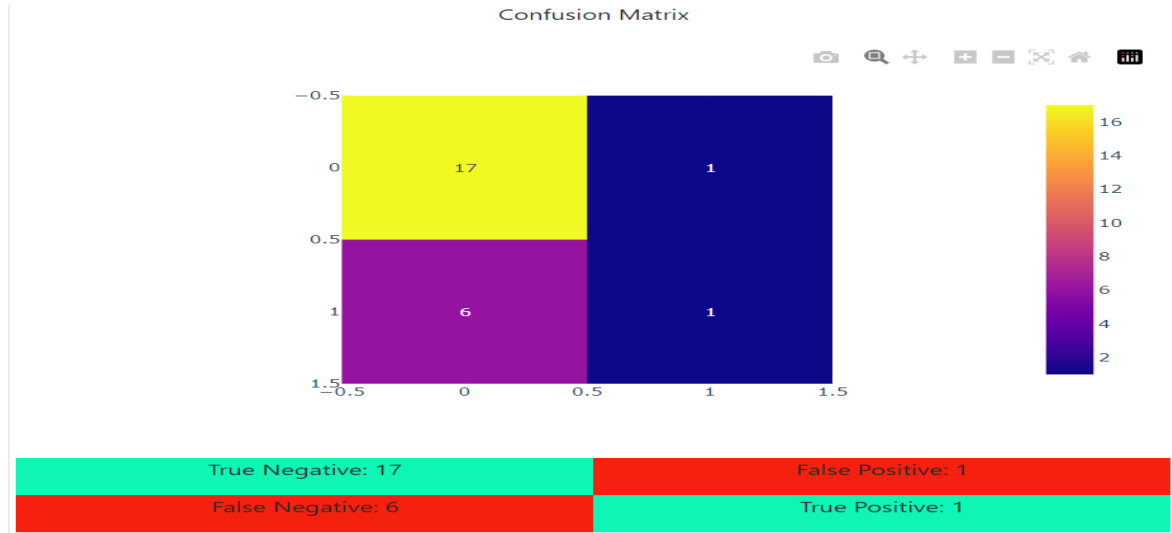
NON-DIPPER

Tahmin Yap

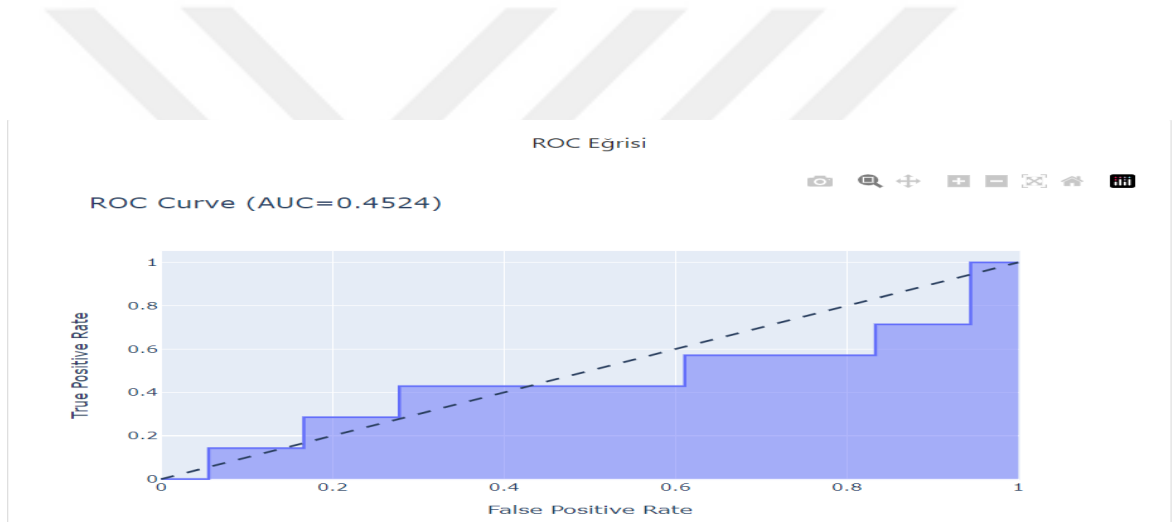
Şekil 3.16. Seçilen verinin ANN tahmini sonuç

Tablo 3.3. Seçilen verinin ANN algoritmanın sınıflandırma raporu.

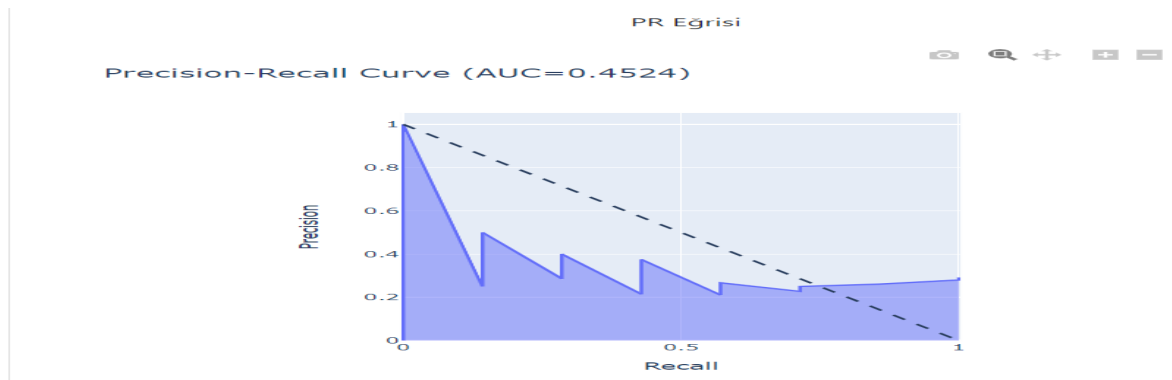
İndex	Precision	Recall	F1-score	Support
1	0.739	0.9444	0.8292	18
2	0.5	0.1428	0.2222	7
accuracy	0.72	0.72	0.72	0.72
macro avg	0.6195	0.5436	0.5257	25
weighted avg	0.6721	0.72	0.6592	25



Şekil 3.17. Seçilen verinin ANN karmaşıklık matrisi.



Şekil 3.18. Seçilen verinin ANN ROC Eğimi.



Şekil 3.19. Seçilen verinin ANN PR eğimi.

3.2.2 Destek Vektör Makinesi (Support Vector Machine SVM)

SVM algoritması için %70 eğitim ve %30 test verileri olarak veri tablosu dağıtımı yapılmıştır. Sklearn kütüphanesinden SVC kütüphanesi kullanılmıştır.

```
SVMnumeric_data_train, SVMnumeric_data_test, SVMtarget_data_train, SVMtarget_data_test = train_test_split(
    numeric_data, targetData, test_size=0.30)
scaler = StandardScaler()
scaler.fit(SVMnumeric_data_train)

SVMnumeric_data_train = scaler.transform(SVMnumeric_data_train)
SVMnumeric_data_test = scaler.transform(SVMnumeric_data_test)
svcclassifier = SVC(kernel='poly', degree=3)
svcclassifier.fit(SVMnumeric_data_train, SVMtarget_data_train)
```

Şekil 3.20. SVM algoritmanın SVC kütüphane kullanımı.

Tekrardan hastaveri_ tablo üzerinden rastgele bir veri seçildi ve “Tahmin Yap” butonunu basarak sonuçları elde edilmiş. Seçilen verinin sonuçlar aşağıdaki gibidir:

Model Tahmini:

SVM

SONUC:

DIPPER

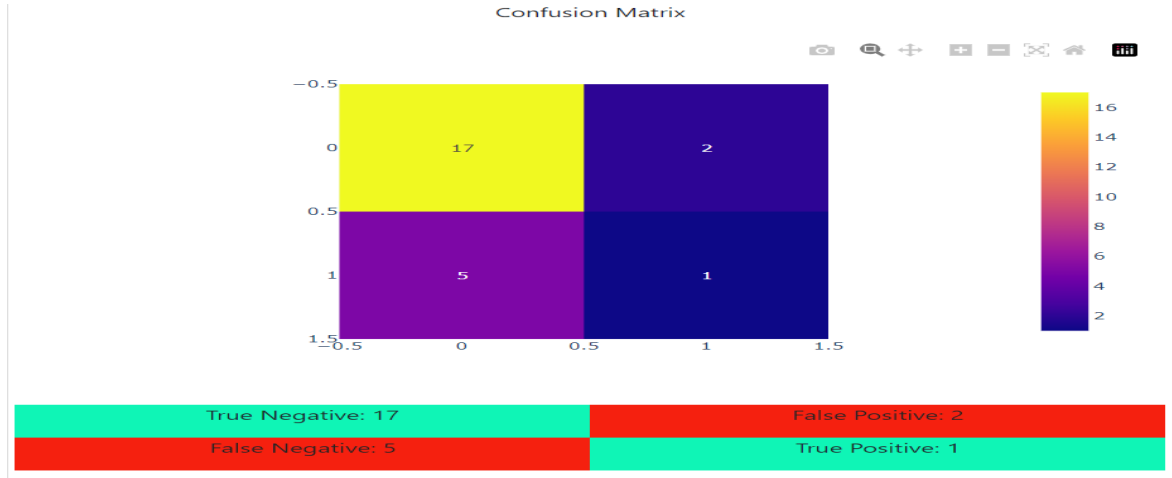
Tahmin Yap

Model Tahmin Sonuclari

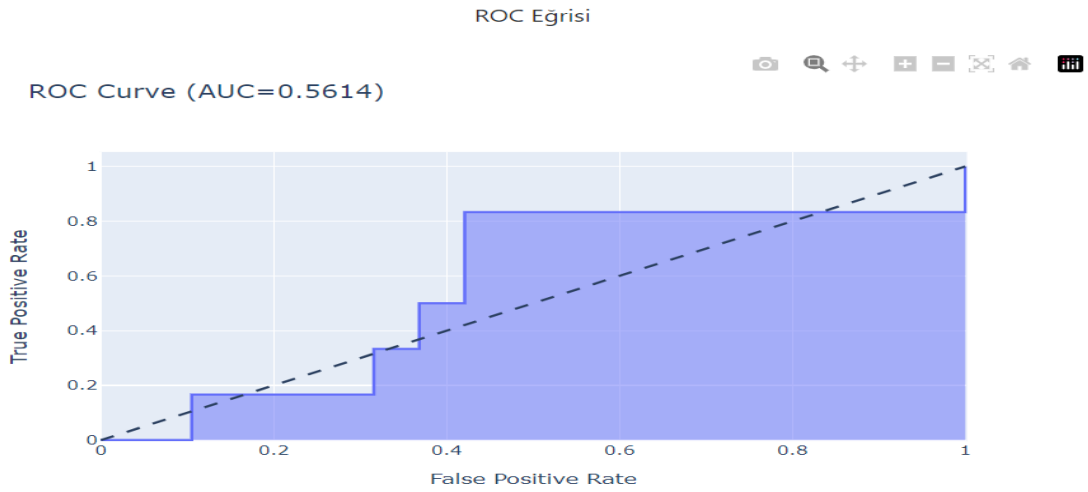
Şekil 3.21. Seçilen verinin SVM tahmini sonuç.

Tablo 3.4. Seçilen verinin SVM algoritmanın sınıflandırma raporu

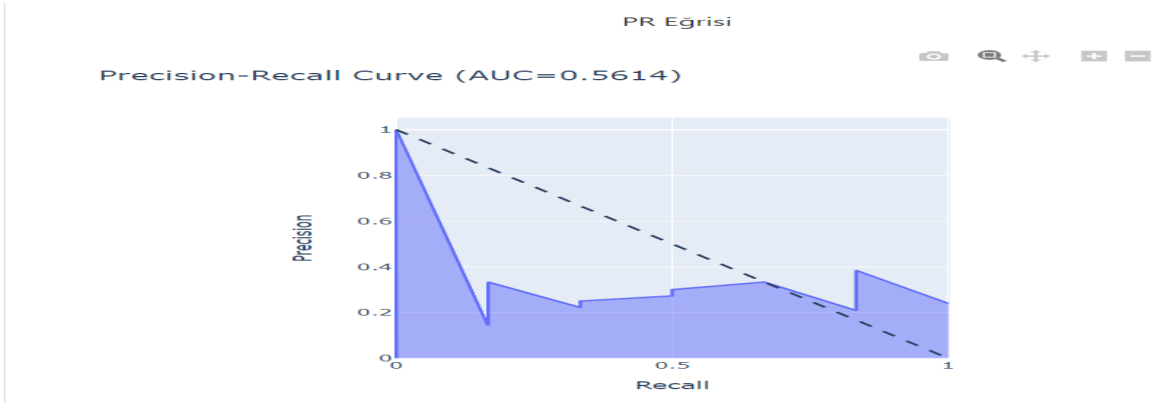
İndex	Precision	Recall	F1-score	Support
1	0.7727	0.8947	0.8292	19
2	0.3333	0.1666	0.2222	6
accuracy	0.72	0.72	0.72	0.72
macro avg	0.5530	0.5307	0.5257	25
weighted avg	0.6672	0.72	0.6835	25



Şekil 3.22. Seçilen verinin SVM algoritmanın karmaşıklık matrisi.



Şekil 3.23. Seçilen verinin SVM algoritmanın ROC eğimi.



Şekil 3.24. Seçilen verinin SVM algoritmanın PR eğimi.

Tabloyu değiştirerek **cancer_** adındaki tablo üzerinde tekrardan rastgele bir veri seçildi ve “Tahmin Yap” butonunu basarak sonuçları elde edilmiş. Seçilen verinin sonuçlar aşağıdaki gibidir.

Model Tahmini:

SVM

SONUC:

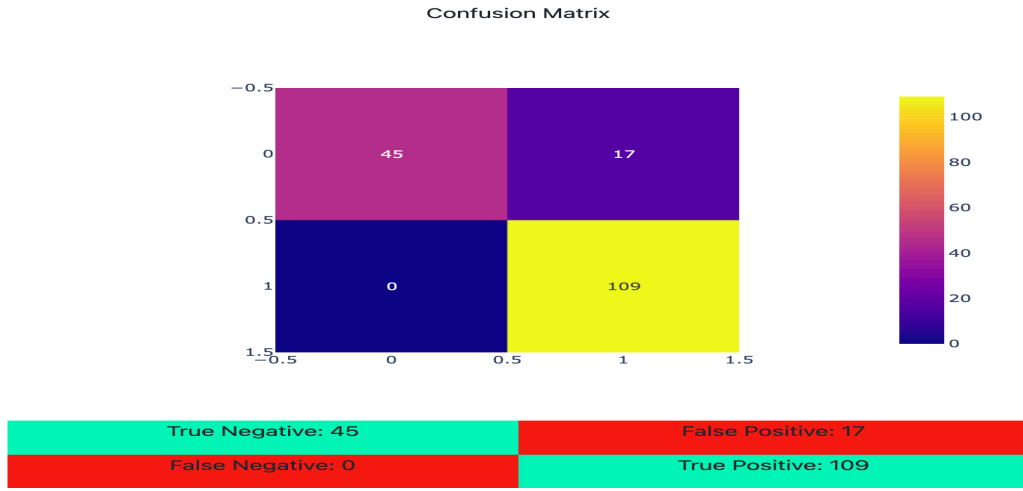
Bening

Tahmin Yap

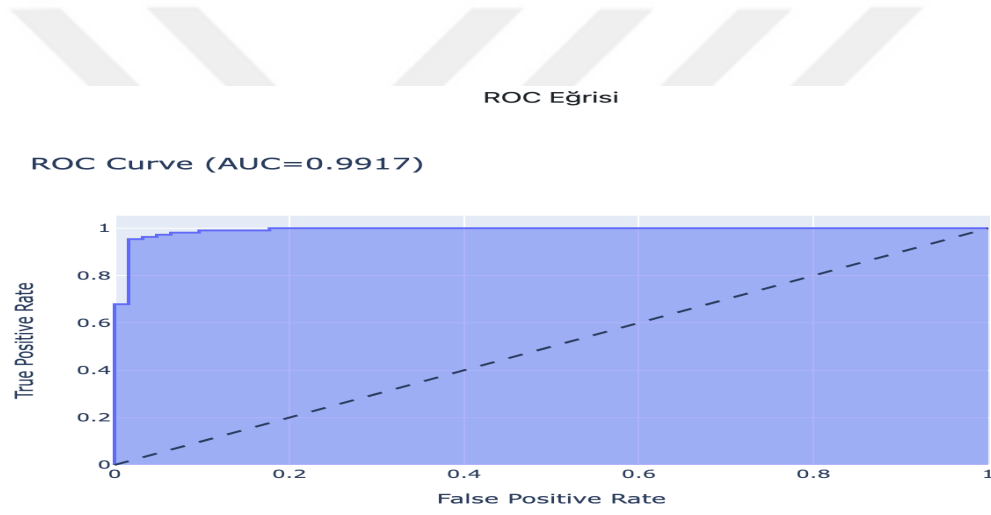
Şekil 3.25. Seçilen verinin SVM algoritmanın tahmini sonuç.

Tablo 3.5. Seçilen verinin SVM algoritmanın sınıflandırma raporu.

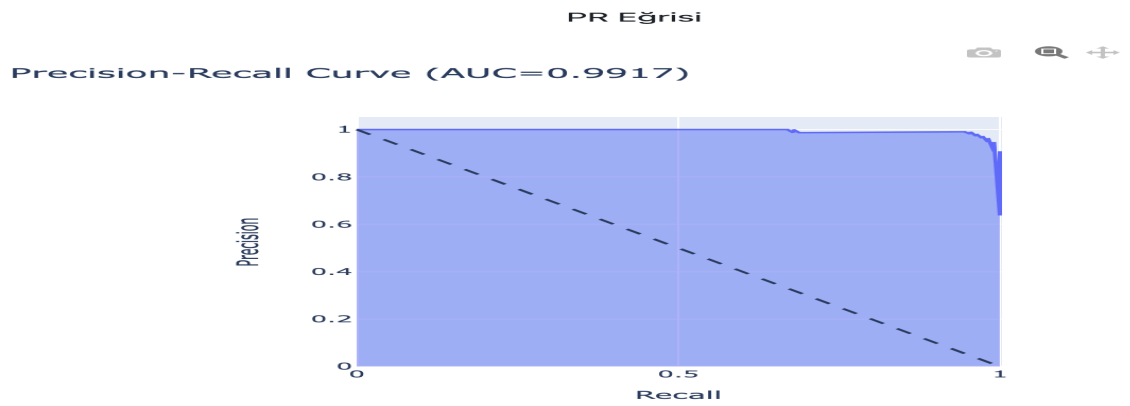
İndex	Precision	Recall	F1-score	Support
1	1	0,7258	0,8411	62
2	0,8650	1	0,9276	109
accuracy	0,9005	0,9005	0,9005	0,900
macro avg	0,9325	0,8629	0,8843	171
weighted avg	0,9139	0,9005	0,8962	171



Şekil 3.26. Seçilen verinin SVM algoritmanın karmaşıklık matrisi.



Şekil 3.27. Seçilen verinin SVM algoritmanın ROC eğimi.



Şekil 3.28. Seçilen verinin SVM algoritmanın PR eğimi.

3.2.3. K-en Yakın Komşu (K-Nearest Neighbor K-NN)

KNN algoritması için %70 eğitim ve %30 test verileri olarak veri tablosu dağıtımı yapılmıştır. Sklearn kütüphanesinden **KNeighborsClassifier** kütüphanesi kullanılmıştır.

```
knn_numeric_data_train, knn_numeric_data_test, knn_target_data_train, knn_target_data_test = train_test_split(
    numeric_data, targetData, test_size=0.30)
scaler = StandardScaler()
scaler.fit(knn_numeric_data_train)

knn_numeric_data_train = scaler.transform(knn_numeric_data_train)
knn_numeric_data_test = scaler.transform(knn_numeric_data_test)

knn_model = KNeighborsClassifier(n_neighbors=3)

knn_model.fit(knn_numeric_data_train, knn_target_data_train)
```

Şekil 3.29. K-NN algoritmanın KNeighborsClassifier kütüphane kullanımı.

Uygulamada veriler kısmında heardesease_ tablo üzerinde rastgele bir veri seçildi ve “Tahmin Yap” butonunu basarak sonuçları elde edilmiş. Seçilen verinin sonuçlar aşağıdaki gibidir:

Model Tahmini:

k-NN

SONUC:

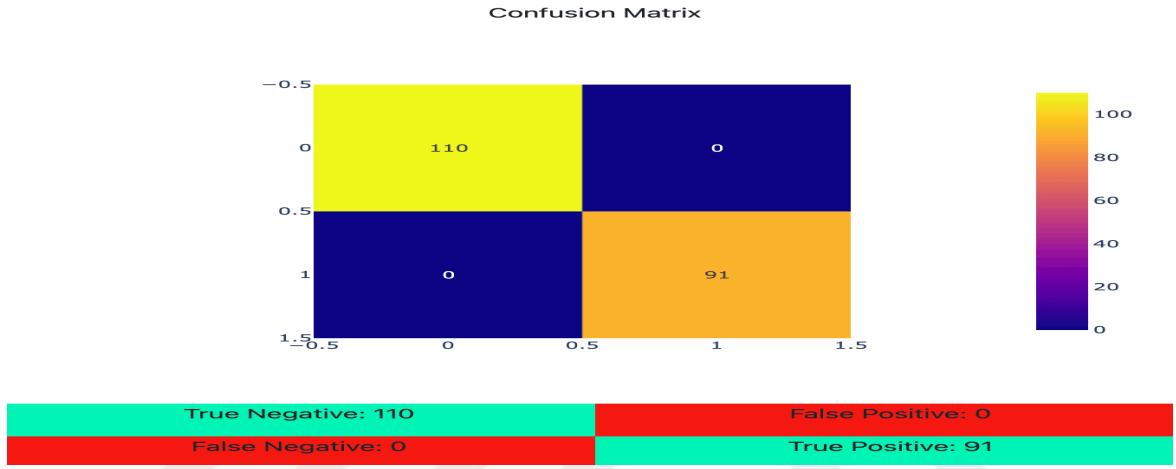
Low

Tahmin Yap

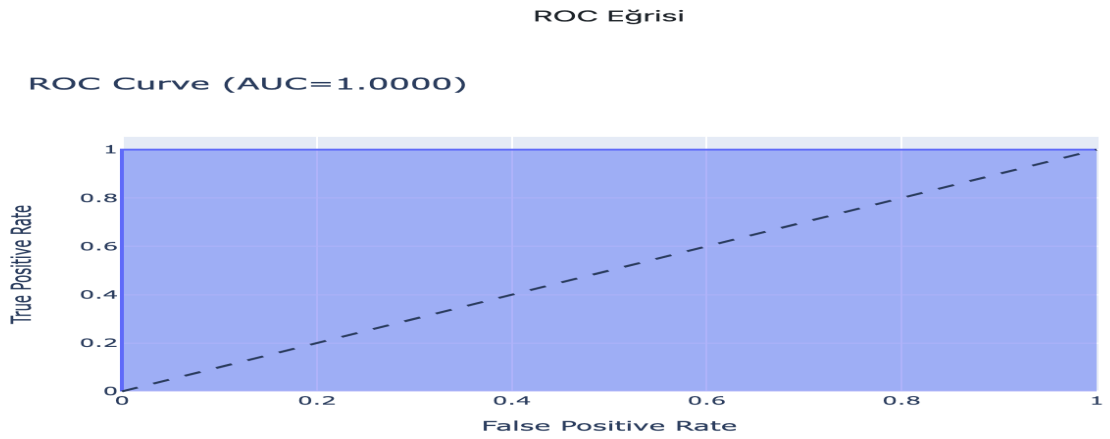
Şekil 3.30. Seçilen verinin K-NN algoritma tahmini sonuç.

Tablo 3.6. Seçilen verinin K-NN algoritma sınıflandırma raporu.

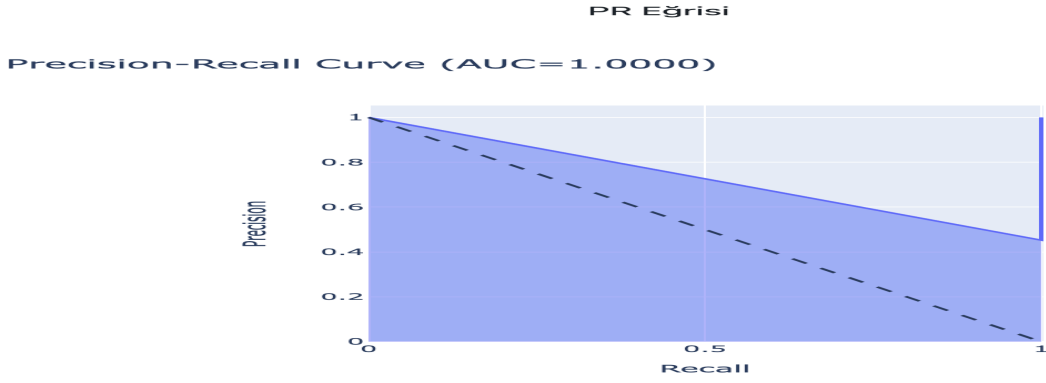
İndex	Precision	Recall	F1-score	Support
1	1	1	1	110
2	1	1	1	91
accuracy	1	1	1	1
macro avg	1	1	1	201
weighted avg	1	1	1	201



Şekil 3.31. Seçilen verinin K-NN algoritma karmaşıklık matrisi.



Şekil 3.32. Seçilen verinin K-NN algoritma ROC eğimi.



Şekil 3.33. Seçilen verinin K-NN algoritma PR eğrisi.

Cancer_ adındaki Tablo üzerine bir veri seçildi ve “Tahmin Yap” butonunu basarak sonuçları elde edilmiş. Seçilen verinin sonuçlar aşağıdaki gibidir:

Model Tahmini:

k-NN

SONUC:

Malign

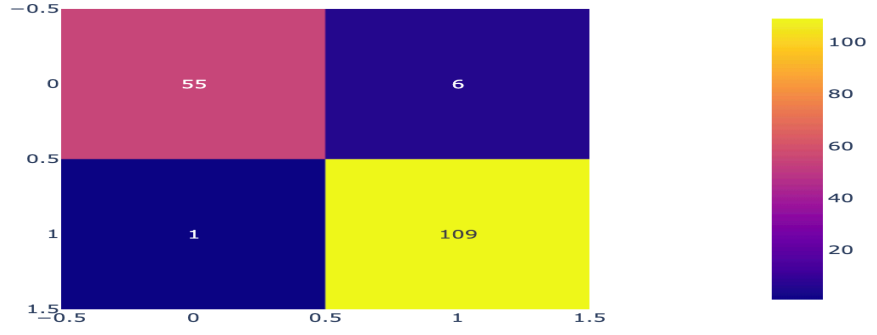
Tahmin Yap

Şekil 3.34. Seçilen verinin K-NN algoritma tahmini sonuç.

Tablo 3.7. Seçilen verinin K-NN algoritma sınıflandırma raporu.

İndex	Precision	Recall	F1-score	Support
1	0,9821	0,9016	0,9401	61
2	0,9478	0,9909	0,9688	110
accuracy	0,9590	0,9590	0,9590	0,9590
macro avg	0,9649	0,9462	0,9545	171
weighted avg	0,9600	0,9590	0,9586	171

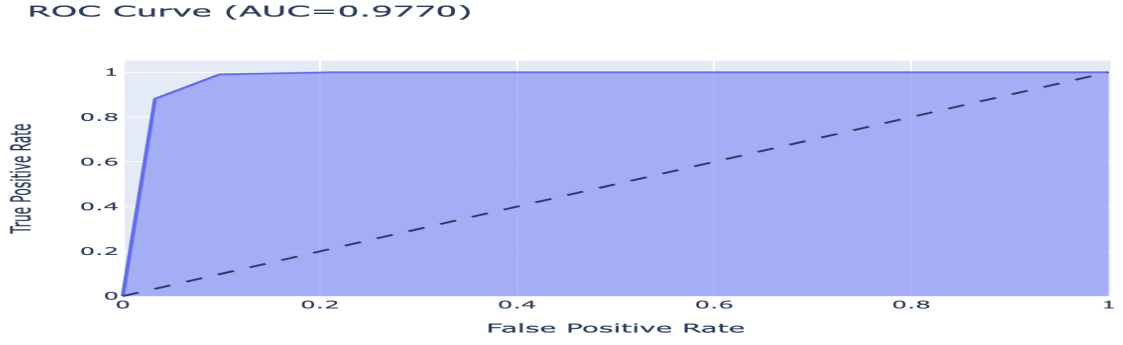
Confusion Matrix



True Negative: 55	False Positive: 6
False Negative: 1	True Positive: 109

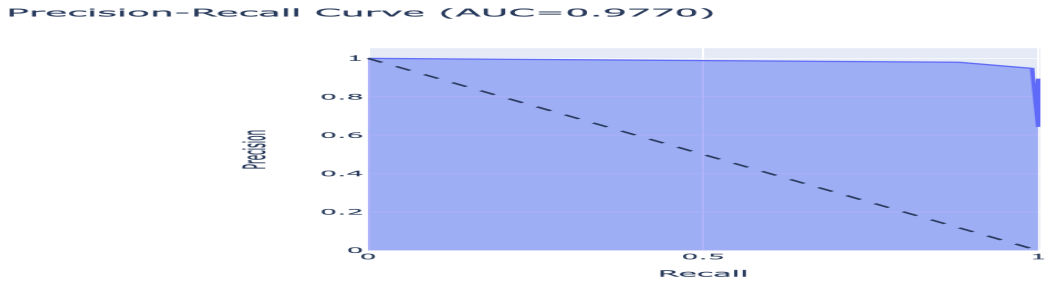
Şekil 3.35. Seçilen verinin K-NN algoritma karmaşıklık matrisi.

ROC Eğrisi



Şekil 3.36. Seçilen verinin K-NN algoritma ROC eğimi.

PR Eğrisi



Şekil 3.37. Seçilen verinin K-NN algoritma PR eğimi.

3.2.4 Karar ağacı (Decision Tree DT)

DT algoritması için %70 eğitim ve %30 test verileri olarak veri tablosu dağıtımı yapılmıştır. Sklearn kütüphanesinden **DecisionTreeClassifier** kütüphanesi kullanılmıştır.

```
d_tree_numeric_data_train, d_tree_numeric_data_test, d_tree_target_data_train, d_tree_target_data_test = train_test_split(
    numeric_data, targetData, test_size=0.30)
scaler = StandardScaler()
scaler.fit(d_tree_numeric_data_train)

knn_numeric_data_train = scaler.transform(d_tree_numeric_data_train)
knn_numeric_data_test = scaler.transform(d_tree_numeric_data_test)

d_tree_classifier = DecisionTreeClassifier(criterion='entropy', random_state=0)

d_tree_classifier.fit(d_tree_numeric_data_train, d_tree_target_data_train)
```

Şekil 3.38. DT algoritma DecisionTreeClassifier kütüphane kullanımı.

Uygulama üzerine heartdesease_ tablodan rastgele herhangi bir veri seçildi ve “Tahmin Yap” butonunu basarak sonuçları elde edilmiş. Seçilen verinin sonuçlar aşağıdaki gibidir:

Model Tahmini:

Decision Tree ▼

SONUC:

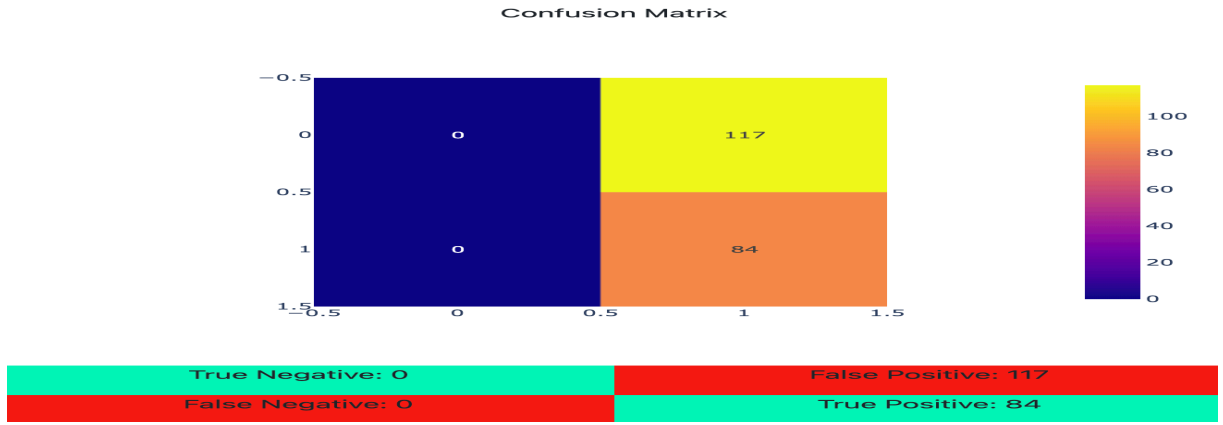
Low

Tahmin Yap

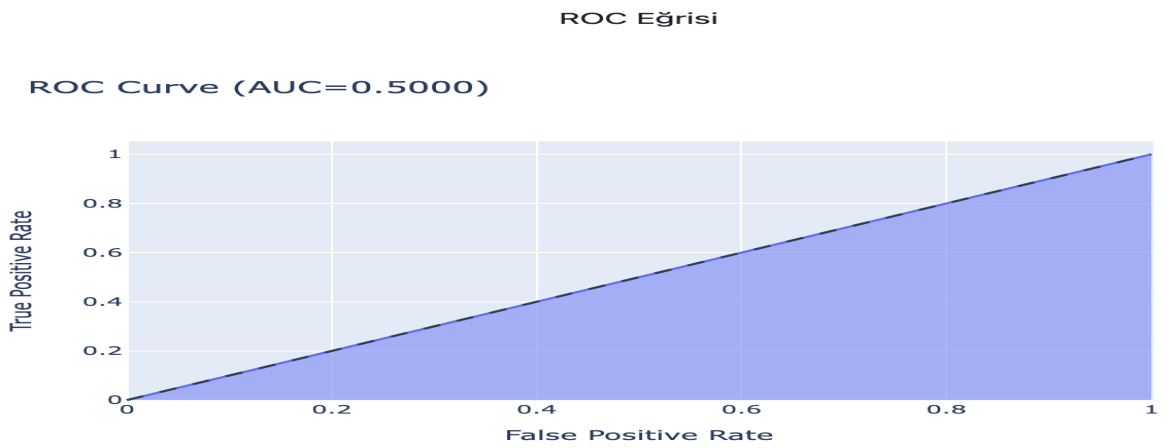
Şekil 3.39. Seçilen verinin DT algoritma tahmini sonuç.

Tablo 3.8. Seçilen verinin DT algoritma sınıflandırma raporu.

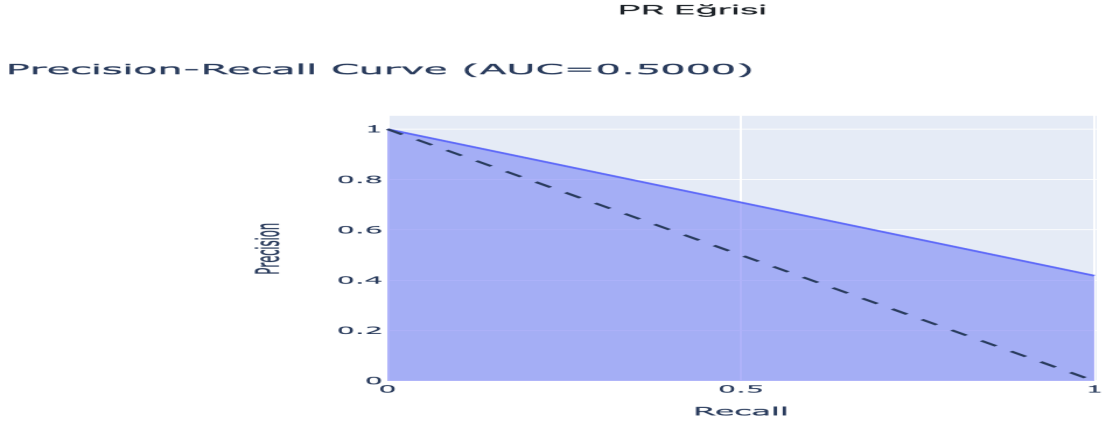
İndex	Precision	Recall	F1-score	Support
1	0	0	0	117
2	0,4179	1	0,5894	84
accuracy	0,4179	0,4179	0,4179	0,4179
macro avg	0,2089	0,5	0,2947	201
weighted avg	0,1746	0,4179	0,2463	201



Şekil 3.40. Seçilen verinin DT algoritma karmaşıklık matrisi.



Şekil 3.41. Seçilen verinin DT algoritma ROC eğrisi.



Şekil 3.42. Seçilen verinin DT algoritma PR eğrisi.

3.2.5. Naive Bayes NB

NB algoritması için %70 eğitim ve %30 test verileri olarak veri tablosu dağıtımı yapılmıştır. Sklearn kütüphanesinden **GaussianNB** kütüphanesi kullanılmıştır.

```
naiveB_numeric_data_train, naiveB_numeric_data_test, naiveB_target_data_train, naiveB_target_data_test = train_test_split(
    numeric_data, targetData, test_size=0.30)
scaler = StandardScaler()
scaler.fit(naiveB_numeric_data_train)

naiveB_numeric_data_train = scaler.transform(naiveB_numeric_data_train)
naiveB_numeric_data_test = scaler.transform(naiveB_numeric_data_test)

naiveB_classifier = GaussianNB()

naiveB_classifier.fit(naiveB_numeric_data_train, naiveB_target_data_train)
```

Şekil 3.43. Naive Bayes algoritma GaussianNB kütüphane kullanımı.

Cancer_ adındaki tabloda bir veri seçildi ve “Tahmin Yap” butonunu basarak sonuçları elde edilmiş. Seçilen verinin sonuçlar aşağıdaki gibidir:

Model Tahmini:

Naive Bayes ▾

SONUC:

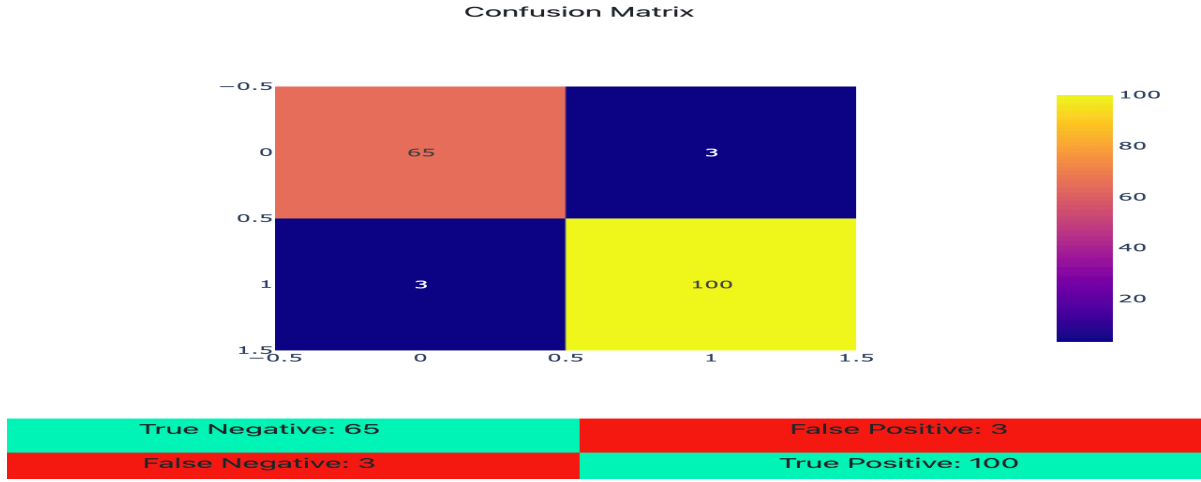
Bening

Tahmin Yap

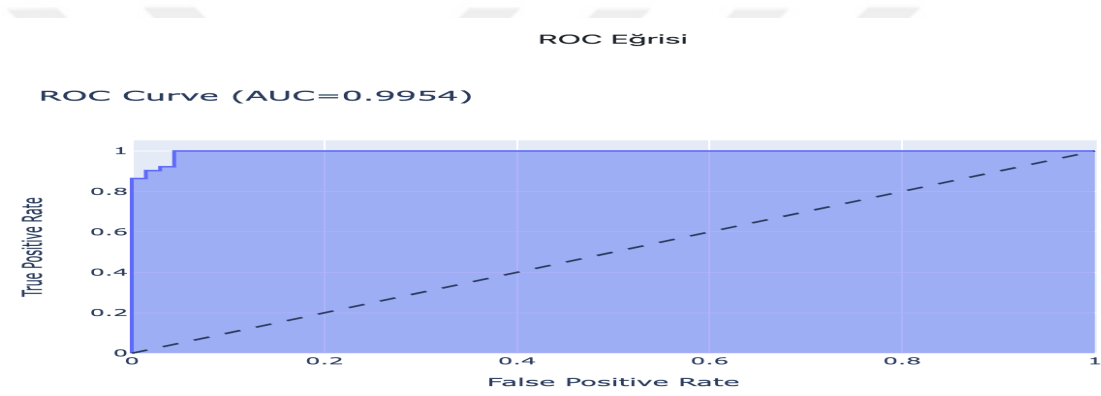
Şekil 3.44. Seçilen verinin Naive Bayes algoritma tahmini sonuç.

Tablo 3.9. Seçilen verinin Naive Bayes algoritma sınıflandırma raporu.

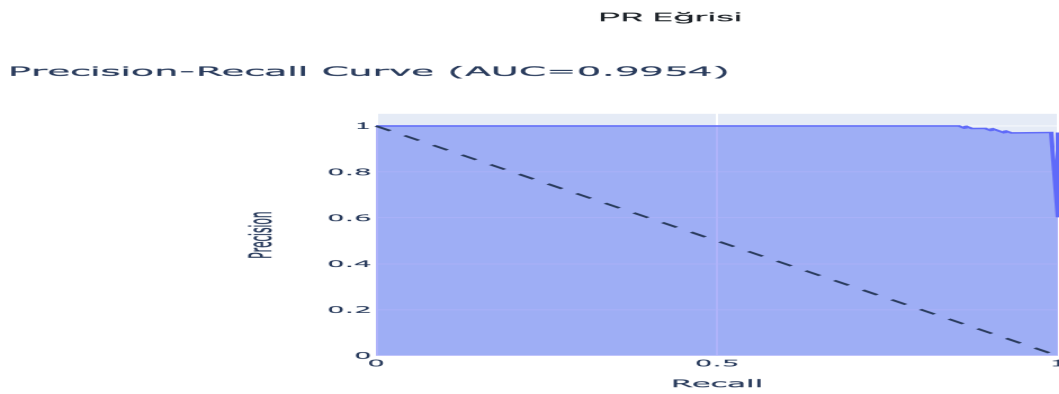
İndex	Precision	Recall	F1-score	Support
1	0,9558	0,9558	0,9558	68
2	0,9708	0,9708	0,9708	103
accuracy	0,9649	0,9649	0,9649	0,9649
macro avg	0,9633	0,9633	0,9633	171
weighted avg	0,9649	0,9649	0,9649	171



Şekil 3.45. Seçilen verinin Naive Bayes algoritma karmaşıklık matrisi.



Şekil 3.46. Seçilen verinin Naive Bayes algoritma ROC eğrisi.



Şekil 3.47. Seçilen verinin Naive Bayes algoritma PR eğrisi.

3.2.6 CatBoost

CatBoost algoritması için %70 eğitim ve %30 test verileri olarak veri tablosu dağıtımı yapılmıştır. Sklearn kütüphanesinden **CatBoostClassifier** kütüphanesi kullanılmıştır.

```
CatBoost_numeric_data_train, CatBoost_numeric_data_test, CatBoost_target_data_train, CatBoost_target_data_test = train_test_split(
    numeric_data, targetData, test_size=0.30)

CatBoost_classifier = CatBoostClassifier(
    iterations=5,
    learning_rate=0.1,
    # loss_function='CrossEntropy'
)

CatBoost_classifier.fit(CatBoost_numeric_data_train, CatBoost_target_data_train)
```

Şekil 3.48. Catboost algoritma CatBoostClassifier kütüphane kullanımı.

hastaveri_ adındaki tabloda bir veri seçildi ve “Tahmin Yap” butonunu basarak sonuçları elde edilmiş. Seçilen verinin sonuçlar aşağıdaki gibidir:

Model Tahmini:

CatBoost

SONUC:

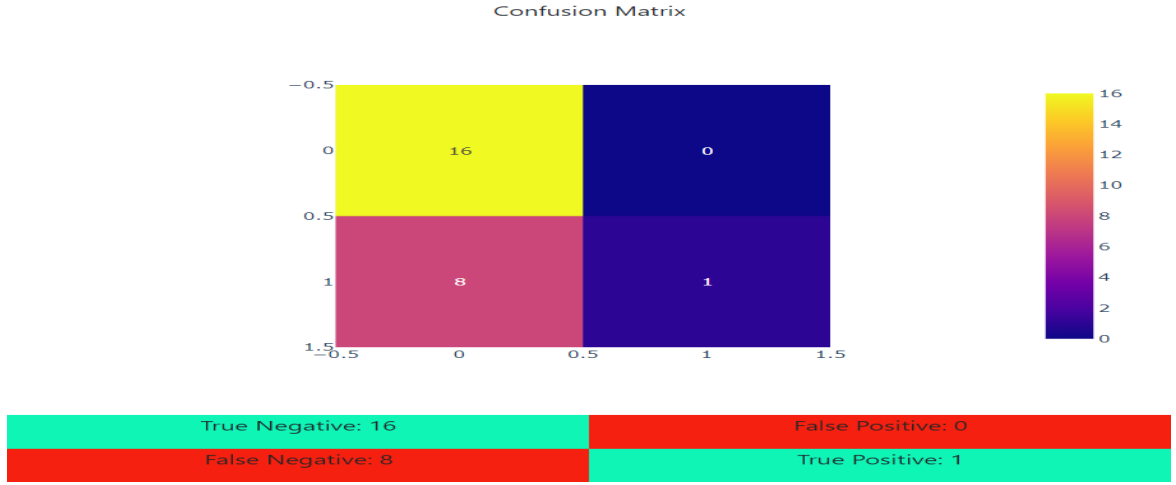
DIPPER

Tahmin Yap

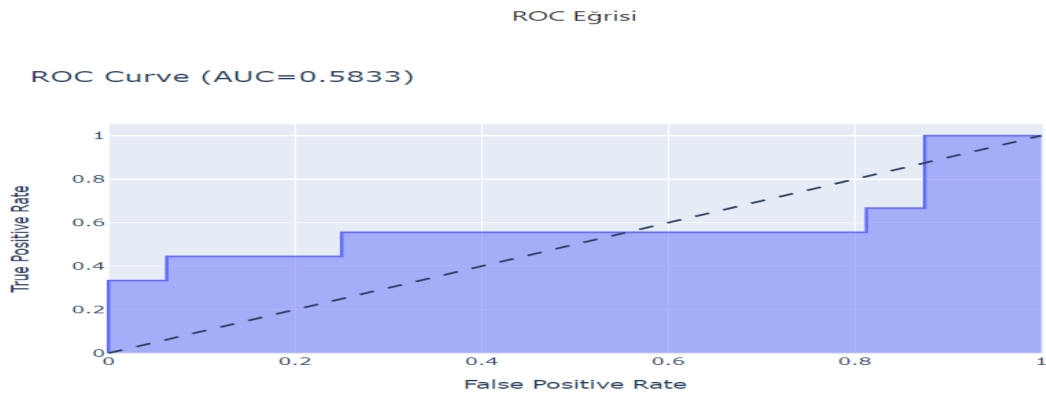
Şekil 3.49. Seçilen verinin CatBoost algoritma tahmini sonuç.

Tablo 3.10. Seçilen verinin CatBoost algoritma sınıflandırma raporu.

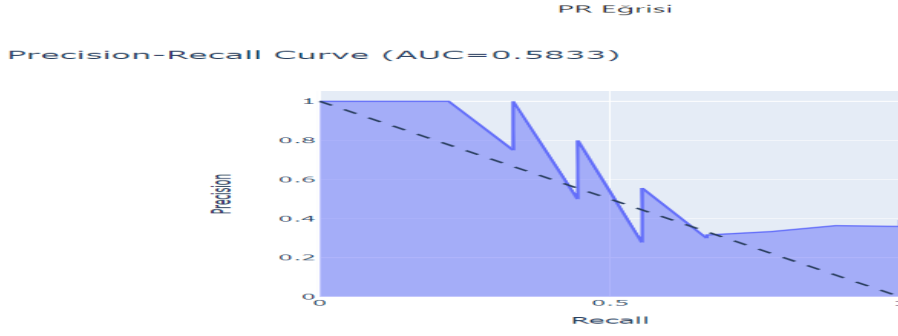
İndex	Precision	Recall	F1-score	Support
1	0.6666	1	0.8	16
2	1	0.1111	0.2	9
accuracy	0.68	0.68	0.68	0.68
macro avg	0.8333	0.5555	0.5	25
weighted avg	0.7866	0.68	0.584	25



Şekil 3.50. Seçilen verinin CatBoost algoritma karmaşıklık matrisi.



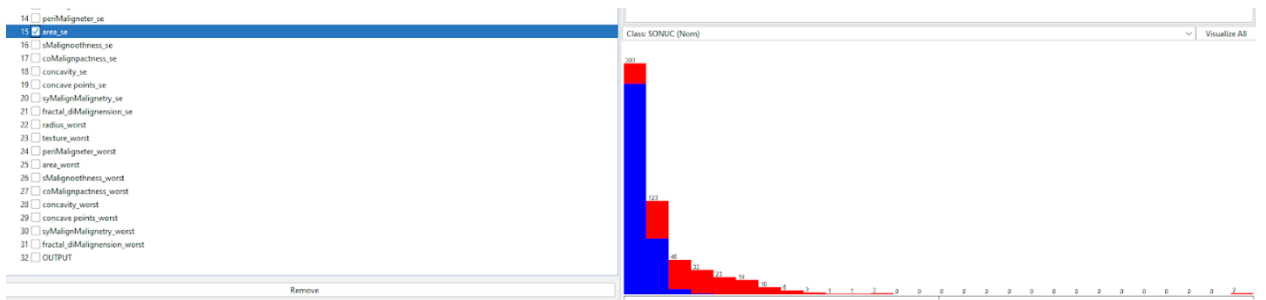
Şekil 3.51. Seçilen verinin CatBoost algoritma ROC eğrisi.



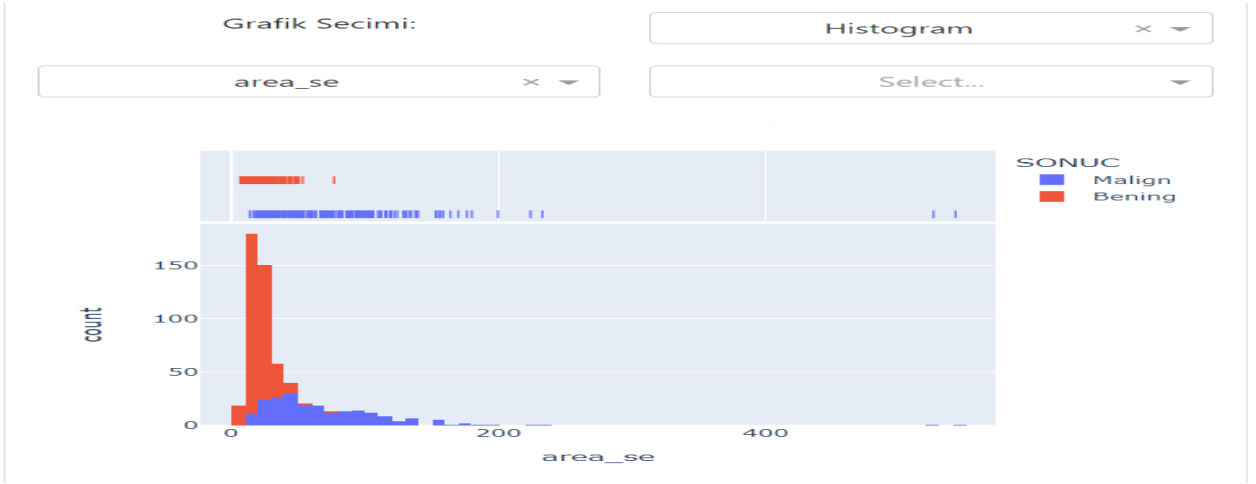
Şekil 3.52. Seçilen verinin CatBoost algoritma PR eğrisi.

3.3. Karşılaştırma ve Tartışma

MEDIC-AI web uygulaması beta versiyonunda medikal alanında yapılan araştırmalar ve bu alanda kullanılan anonim veri setleri kullanarak geliştirilmiştir. Geliştirilen web uygulamasının sonuçlarını teyit edebilmek için ve kullanılan makine öğrenme algoritmalarının tahminleme sonuçlarını doğruluklarını kontrol etmek için, WEKA uygulaması ile, anonime veri setleri üzerine karşılaştırma yapılmıştır. Karşılaştırmada veri seti olarak web uygulamasında cancer_ adında adlandırılan anonim veri seti kullanılmıştır. İki (2) uygulama arasında Naive Bayes ve ANN ve algoritmalarının tahmini sonuçlarının, sınıflandırma raporu, K-Means kümeleme algoritmanın ve histogram gibi grafikleri karşılaştırılmıştır. Sonuçlar aşağıdaki gösterilen görsellerdeki gibi çıkmıştır.

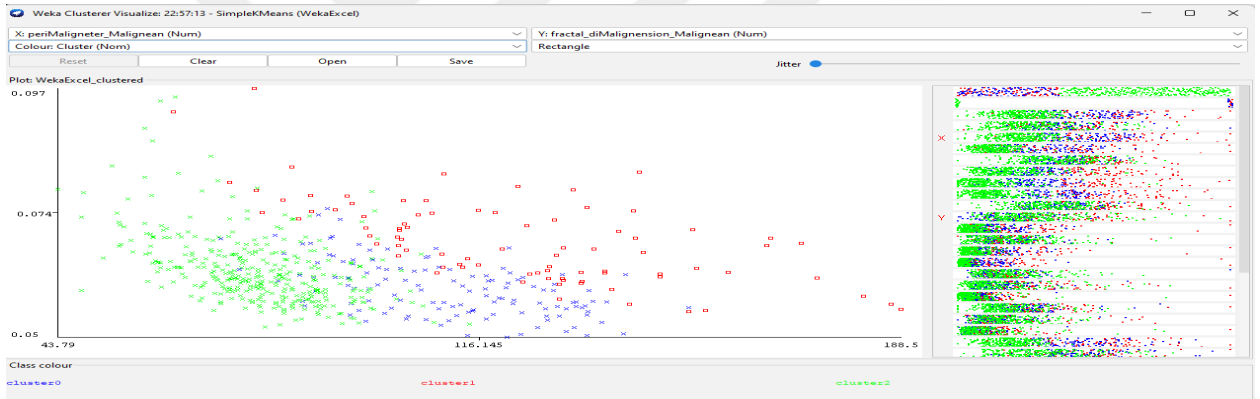


3.53. Weka uygulamasında histogram gösterimi.

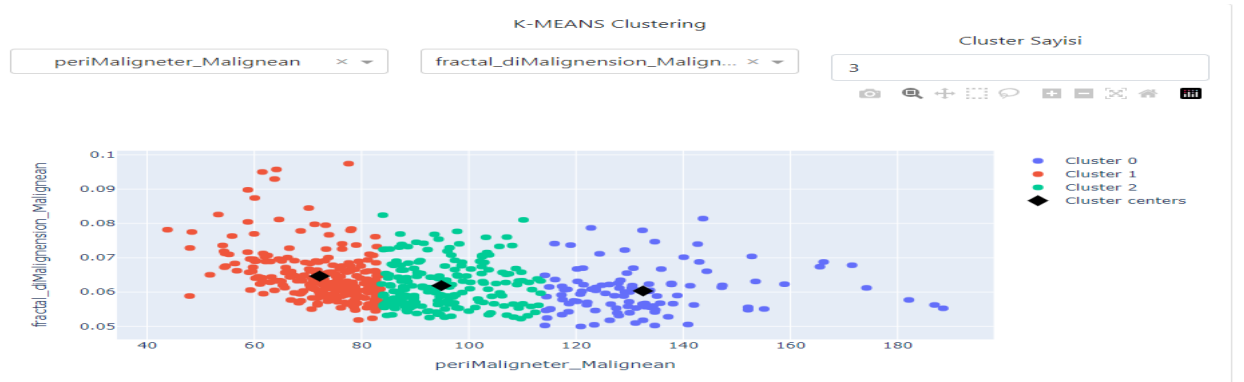


3.54. MEDIC-AI uygulamasında histogram gösterimi.

İki uygulamayla K-Means kümeleme karşılaştırması yapılmıştır. Sonuçlar aşağıdaki gibi görsellerde gösterilmiştir.



3.55. WEKA uygulamasında k-means kümeleme gösterimi.



3.56. MEDIC-AI uygulamasında k-means kümeleme gösterimi.

MEDIC-AI ve WEKA uygulamalarıyla bazı makine öğrenme algoritmalarıyla tahminleme ve analiz sonuçlarının karşılaştırılması yapılmıştır. MEDIC-AI uygulamasında sınıflandırma raporundaki ve karmaşıklık matrisindeki indeksler aşağıdaki gibi ayrılmıştır: 1- Malign, 2 – Bening

```

=== Detailed Accuracy By Class ===
                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0.990    0.070    0.952      0.990    0.971      0.928    0.980     0.981     Bening
                0.930    0.010    0.985      0.930    0.957      0.928    0.978     0.971     Malign
Weighted Avg.   0.965    0.045    0.966      0.965    0.965      0.928    0.979     0.977

=== Confusion Matrix ===
 a  b  <-- classified as
99  1  |  a = Bening
 5  66 |  b = Malign

```

3.57. WEKA uygulamasında Naive Bayes algoritmanın sonuçları.

Classification Report

Export				
index	precision	recall	f1-score	support
1	0.9402985074626866	0.875	0.9064748201438849	72
2	0.9134615384615384	0.9595959595959596	0.9359605911330049	99
accuracy	0.9239766081871345	0.9239766081871345	0.9239766081871345	0.9239766081871345
macro avg	0.9268800229621126	0.9172979797979798	0.921217705638445	171
weighted avg	0.9247613148830744	0.9239766081871345	0.9235455296639018	171

3.58. MEDIC-AI uygulamasında Naive Bayes algoritmanın sınıflandırma raporu.

```

=== Detailed Accuracy By Class ===
                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Bening
                1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Malign
Weighted Avg.   1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===
 a  b  <-- classified as
100  0  |  a = Bening
 0  71 |  b = Malign

```

3.59. WEKA uygulamasında ANN algoritmanın sonuçları.

Classification Report

index	precision	recall	f1-score	support
1	0.9857142857142858	0.971830985915493	0.9787234042553192	71
2	0.9801980198019802	0.99	0.9850746268656716	100
accuracy	0.9824561403508771	0.9824561403508771	0.9824561403508771	0.9824561403508771
macro avg	0.982956152758133	0.9809154929577465	0.9818990155604954	171
weighted avg	0.9824883992158615	0.9824561403508771	0.9824375695245313	171

3.60. MEDIC-AI uygulamasında ANN algoritmanın sınıflandırma raporu.

Yapılan karşılaştırma sonucunda iki (2) uygulama arasında %80 ile %90 benzerlik çıkmaktadır. Bu demek ki şu ana kadar geliştirdiğimiz MEDIC-AI web uygulamasında kullanılan makine öğrenme algoritmaları, k-means kümeleme grafiksel gösterimi ve diğer grafikler büyük ölçüde doğru bir şekilde kullanılmıştır. WEKA ile karşılaştırılan makine öğrenme doğruluk oranlarının %100 oranlarında çıkmak için MEDIC-AI uygulamasında sunulan makine öğrenme algoritmalarının daha fazla geliştirmeye gerektiğini kanaat getirdik.

Ayrıca tez kapsamında geliştirilen bu web uygulaması, algoritmalarda test ve eğitim dağıtımını, algoritma özelliklerini (K-NN algoritmasında k değerini belirlemesi gibi) kullanıcı tarafından girilmesini bir sonraki sürüm için geliştirilmektedir. Veri analizi için yeni grafikler (korelasyon grafikleri gibi vs.), verinin özelliklerini ve içindekilerini anlamak için bir ‘Veri Tanıma’ sayfasını da eklenecektir. Böylece veri yüklendiğinde, verinin içindekilerini ve özelliklerin anlamlarını öğrenebilmek için, veri analizinde ilk bakışta kullanıcıya çok fazla yardımda bulunacaktır. MEDIC-AI uygulaması son sürümünde geldiğinde, kullanıcıya veri anlama, veri analizi ve grafikler üzerine incelemeler vs. gibi işlemlerini rahatlıkla yapabilecek, bunun yanında seçebileceği algoritmalarından tahmini sonuçlar yapabilecek ve istediği grafikleri seçip inceleyebilecektir. Birden fazla algoritmaları sunarak kullanıcıya tahmini sonuçlarını inceleyip, hangi algoritma hangi veri setine daha iyi sonuçlar verdiğini rahatlıkla anlayabilecektir. Literatürde baktığımızda yaptığımız çalışmanın benzer çalışmalarını ve uygulamalarını değerlendirdik ve MEDIC-AI uygulamasıyla karşılaştırdık. Karşılaştırma yaptığımız uygulamalar ise CELLAR ve WEKA uygulamalarıdır. CELLAR uygulamasını <https://cellar.cmu.hubmapconsortium.org/app/cellar> link üzerine ulaşılabilir. Çıkan bazı sonuçlar aşağıdaki tabloda yer verilmiştir.

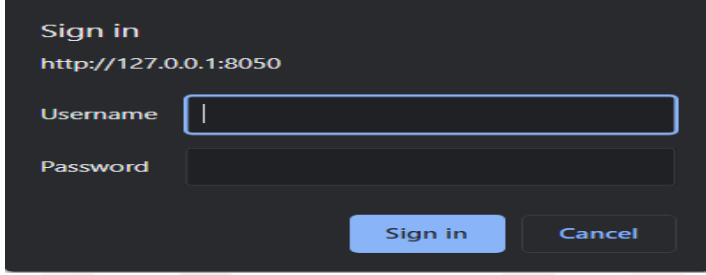
Tablo 3.11. Uygulamalar arasında karşılaştırma.

Karşılaştırma	Uygulamalar (MEDIC-AI, WEKA, CELLAR)
Benzerlikler	Veri Yükleme, .csv veya excel formatlarını tanıma, ön veri işleme, MEDIC-AI ve CELLA: veri tabanında kaydetme ve silme, birden fazla ML algoritma seçme ve tahminleme, tahminleme sonucu, grafiksel gösterimi üzerine veri analizi gibi. 2 uygulamalar da erişim kolaylıkları sunmak için web tabanlı olarak geliştirilmiştir.
Farklılıklar	<p>MEDIC-AI beta versiyonu: ML algoritmalarında varsayılan değerler tanımlı, tahminleme sonuçlarını tablo gösterimi, ROC ve PR eğimlerini grafiksel gösterimi, tablo üzerine boş satır gösterimi, yeni örneklem ve özellik ekleme gibi özellikler.</p> <p>CELLAR ve WEKA: Seçilen ML algoritmalarında istenilen değerler atılmaktadır, varsayılan değerlerle çalışmamaktadır. CELLAR biyomedikal veriler üzerine geliştirilmiş ve sadece biyomedikal verilerle çalışmaktadır. WEKA ise farklı veri setlerle çalışmaktadır ve kendine tanımlı dosya formatı (.arff) bulunmaktadır.</p> <p>Weka desktop tabanlı olup, MEDIC-AI ve CELLAR web tabanlı uygulamalardır.</p>

3.4. MEDIC-AI ML destekli KDS web uygulaması

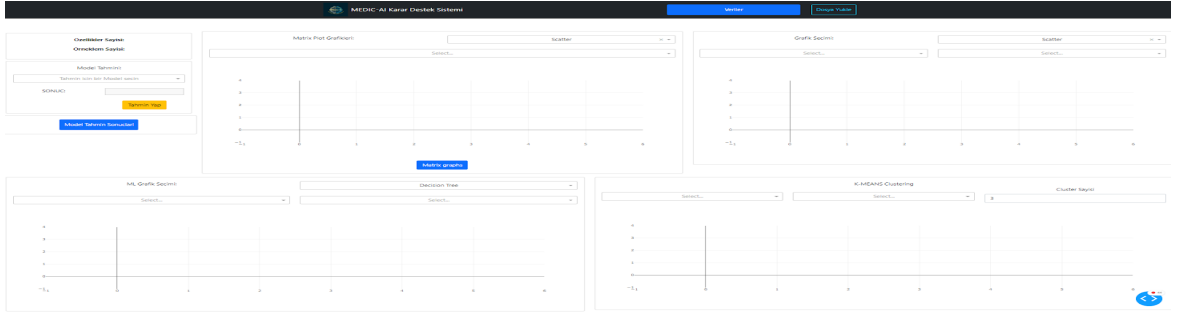
Yapılan MEDIC-AI karar destek web uygulamasının temel özellikleri, kullanıcı tarafından .csv ve .xlsx formatından istediği gibi veri yükleyebilmesidir. Yine de uygulama üzerinde tablo halinde veri görüntüleyebilir, veri içeriğini inceleyebilir ve grafikler oluşturabilir. Ayrıca sadece bir ML algoritma ile değil birden fazla ML algoritmaları kullanarak tahminleme yapabilir ve bu tahminlemelerin sonuçlarını, yani tahminlemenin sonucunu, ROC ve PR eğrilerini, sınıflandırma raporunu, karışıklık matrisini gösterilmektedir.

MEDIC-AI web uygulaması <http://medicai-beta.herokuapp.com/> link üzerine erişebilmektedir. Uygulama beta versiyonunda olduğu için bulut platformunda medicaidbeta olarak isimlendirilmiştir. İlk olarak uygulamaya girmeye çalıştığımızda bir giriş ekranı karşınıza çıkıyor.



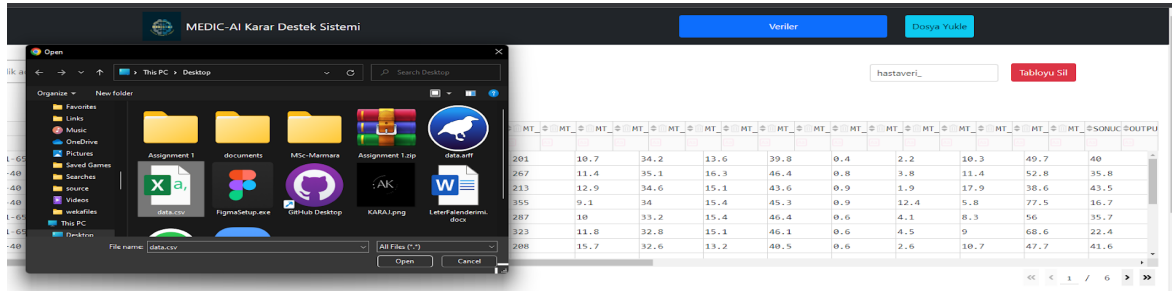
Şekil 3.61. MEDIC-AI web uygulamasının giriş ekranı.

Kullanıcı Adı ve şifre girdikten sonra şekil 3.55’te gösterilen ekran önünüzde çıkmaktadır. Sayfa boş olarak, hiçbir grafik olmaksızın ilk kez açılmaktadır.



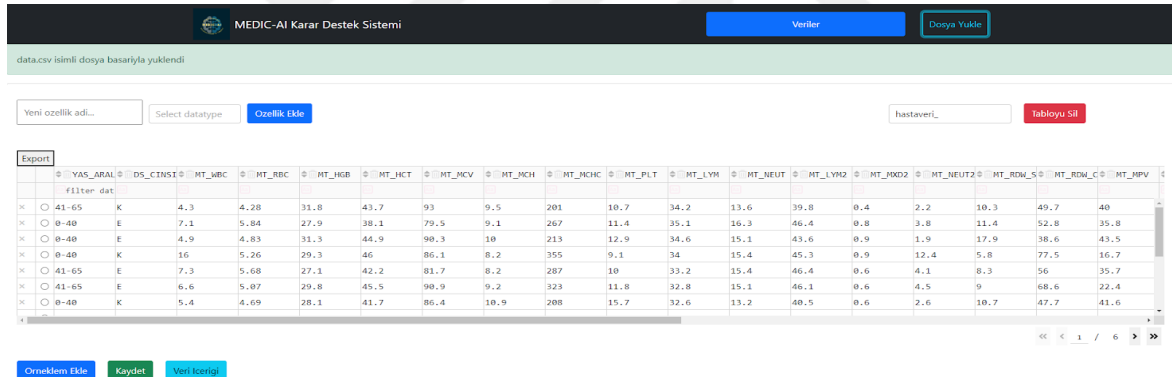
Şekil 3.62. MEDIC-AI uygulamasından giriş yaptıktan sonra ilk ekran.

Bu demek ki veri tabanımızda herhangi bir tablo verimiz bulunmamaktadır. Veri yüklemek için “Dosya Yükle” butonunu basarak .csv veya .xlsx formatından veri yükleyebiliriz.



Şekil 3.63. Uygulamadan .csv formatında veri yüklemek.

Dosya yüklerken dikkat etmemiz gereken bir kural vardır. Daha sonra makine öğrenimi algoritmalarını ve grafiklerini sorunsuz bir şekilde çalışmasını istiyorsak veri dosyasında mecburi olarak “SONUC” kolonu olması gerekmektedir. Bu kolonun olması mecburiyeti makine öğrenme algoritmalarından kullandığımız “target label” dediğimiz “OUTPUT” kolonunu oluşturmak içindir. Dosya yükleme ve kaydetme esnasında verilen uyarı mesajını aşağıdaki görsellerdeki gibidir.



Şekil 3.64. Dosyanın başarıyla bir şekilde yüklediğini gösteren uyarı mesajı.

Şekil 3.58.’de dosya yükleme ve kırmızı alanın içinde “OUTPUT target label” oluşturulan kodları gösterilmektedir.

```

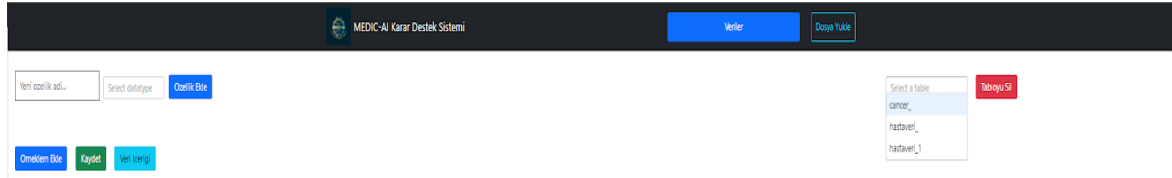
# callbacks for data uploading
def parse_contents(contents, filename):
    content_type, content_string = contents.split(',')
    fileNameCSV = filename.split('.csv')[0]
    fileNameXLS = filename.split('.xlsx')[0]

    decoded = base64.b64decode(content_string)
    try:
        if 'csv' in filename:
            # Assume that the user uploaded a CSV file
            df = pd.read_csv(
                io.StringIO(decoded.decode('utf-8')))
            # label_dict = dict(zip(range(len(df['SONUC'].unique())), df['SONUC'].unique()))
            label_dict = dict(zip(np.arange(1, len(df['SONUC'].unique()) + 1), df['SONUC'].unique()))
            df_label = pd.DataFrame.from_dict(label_dict, orient='index').reset_index().rename(
                columns={0: 'SONUC', 'index': 'OUTPUT'})
            df_new = df.merge(df_label, on='SONUC')
            df_new.to_sql(name=fileNameCSV.lower() + '_', con=db.engine, if_exists='replace', index=False)
        elif 'xlsx' in filename:
            # Assume that the user uploaded an excel file
            df = pd.read_excel(io.BytesIO(decoded))
            # label_dict = dict(zip(range(len(df['SONUC'].unique())), df['SONUC'].unique()))
            label_dict = dict(zip(np.arange(1, len(df['SONUC'].unique()) + 1), df['SONUC'].unique()))
            df_label = pd.DataFrame.from_dict(label_dict, orient='index').reset_index().rename(
                columns={0: 'SONUC', 'index': 'OUTPUT'})
            df_new = df.merge(df_label, on='SONUC')
            df_new.to_sql(name=fileNameXLS.lower() + '_', con=db.engine, if_exists='replace', index=False)
    except Exception as e:
        print(e)

```

Şekil 3.65. Dosya yükleme ve OUTPUT target label oluşturma.

Veri yüklediğimizde “Veriler” butonunu basarak tablo seçimi yapabiliriz. İntput kısmında veri tabanında bulunan tablo isimlerini görebiliriz. Böylece bir tablo ismini seçtiğimizde veriler tablo halinde uygulamamızda görebiliriz. Giriş bölümünde şekil 1.5.’te gösterilmiştir. Aynı şekil üzerinde “Veri İçeriği” butonu altında tablomuzda bulunan bütün sütun isimlerini göstermektedir.



Şekil 3.66. Veri tabanından tablo çekimi.

Veri tabanında kaydedilen yeni yüklediğimiz verilerden herhangi birini seçip, veri tabanından çekip tablo şeklinde kullanıcıya rahatlıkla gösterebiliyoruz. Yanı sıra veri tabanından çekilen tablonun özelliklerin, “Veri İçeriği” adındaki butona basılarak, isimlerini de kullanıcıya sunmaktadır.

MEDIC-AI Karar Destek Sistemi

Veriler Dosya Yükle

Yeni özellik adı... Select datatype Özellik Ekle cancer_ Tabloyu Sil

Export

	SONUC	radius	texture	periMal	area_Ma	sMalign	coMalign	concavi	concave	syMalign	fractal	radius	texture	periMal	area_se	sMalign	coMalign	concavi
x	Benign	6.981	13.43	43.79	143.3	0.117	0.07308	0	0.193	0.07818	0.2241	1.508	1.553	9.833	0.00508	0.006098	0.01069	
x	Benign	12.18	20.52	77.22	458.7	0.08013	0.04038	0.02383	0.0177	0.1739	0.05677	0.1924	1.571	1.183	14.68	0.00508	0.006098	0.01069
x	Benign	9.876	19.4	63.95	298.3	0.1005	0.09697	0.06154	0.03029	0.1945	0.06322	0.1803	1.222	1.528	11.77	0.009058	0.02196	0.03029
x	Benign	10.49	19.29	67.41	336.1	0.09989	0.08578	0.02995	0.01201	0.2217	0.06481	0.355	1.534	2.302	23.13	0.007595	0.02219	0.0288
x	Benign	11.64	18.33	75.17	412.5	0.1142	0.1017	0.0707	0.03485	0.1801	0.0652	0.306	1.657	2.155	20.62	0.00854	0.0231	0.02945
x	Benign	12.36	18.54	79.01	466.7	0.08477	0.06815	0.02643	0.01921	0.1602	0.06066	0.1199	0.8944	0.8484	9.227	0.003457	0.01047	0.01167
x	Benign	11.34	21.26	72.48	396.5	0.08759	0.06575	0.05133	0.01899	0.1487	0.06529	0.2344	0.9861	1.597	16.41	0.009113	0.01557	0.02443
x	Benign	9.777	16.99	62.5	290.2	0.1037	0.08404	0.04334	0.01778	0.1584	0.07065	0.403	1.424	2.747	22.87	0.01385	0.02932	0.02722

Orneklem Ekle Kaydet Veri İçeriği

```

SONUC radius_Malignean texture_Malignean periMalignean area_Malignean sMalignnoothness_Malignean coMalignpactness_Malignean concavity_Malignean concave points_Malignean
syMalignMalignetry_Malignean fractal_dMalignension_Malignean radius_se texture_se periMaligneter_se area_se sMalignnoothness_se coMalignpactness_se concavity_se concave points_se
syMalignMalignetry_se fractal_dMalignension_se radius_worst texture_worst periMaligneter_worst area_worst sMalignnoothness_worst coMalignpactness_worst concavity_worst concave points_worst
syMalignMalignetry_worst fractal_dMalignension_worst OUTPUT

```

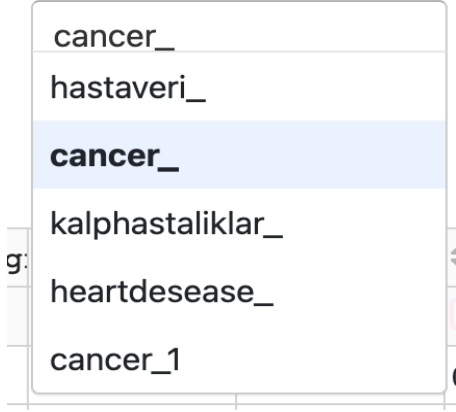
Şekil 3.67. Veri tabanından seçilen tablonun gösterimi ve veri içeriği

Ayrıca uygulama üzerine yüklenen verinin kaç tane özellik (sütun) ve örneklem (sıra) sayısı olduğunu da gösterilmektedir. Böylece ilk bakışta veri yüklendiğinde veri hakkında birkaç bilgi öğrenebilmekteyiz.

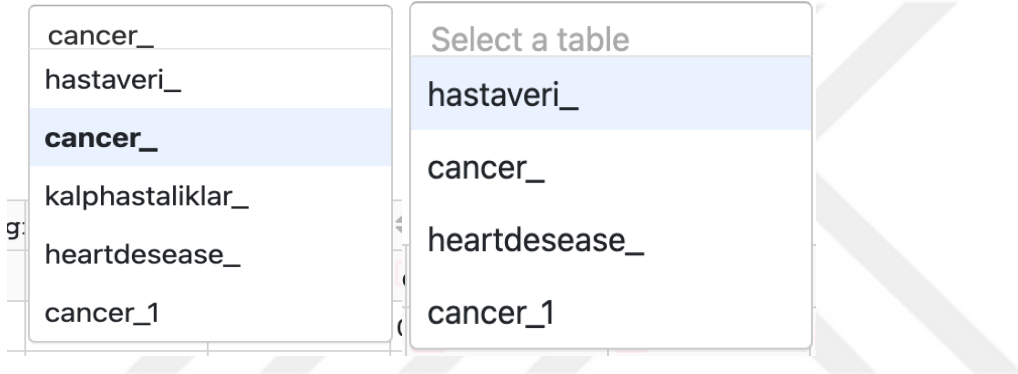
Ozellikler Sayısı: 32	Ozellikler Sayısı: 26	Ozellikler Sayısı: 31
Orneklem Sayısı: 569	Orneklem Sayısı: 1000	Orneklem Sayısı: 83

Şekil 3.68. Farklı tabloların Özellikler ve Örneklem sayısı gösterimi.

MEDIC-AI uygulamasında Veriler kısmında kullanıcıya 'Kaydet' butonuyla tablo üzerine yaptığı herhangi bir değişiklik farklı adında ana tabloyu bozmadan kaydetmektedir. Yine de uygulama üzerine 'Tablo Sil' adındaki özellik ile kullanıcı istediği tabloyu silebilir.



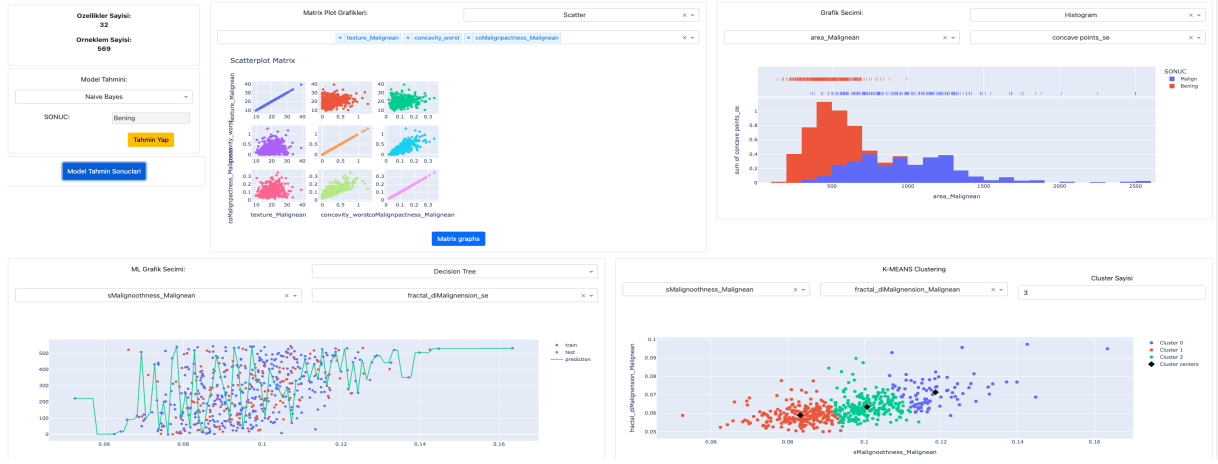
Şekil 3.69. kaydetme esnasında oluşturulan yeni cancer_1 adındaki tablo



Şekil 3.70. Silme esnasında silinen kalphastaliklar_ tablonun silinmesi

MEDIC-AI uygulaması üzerine özellik veya örneklem silme ve ekleme, kullanıcı tarafından kolaylıkla fark edilsin diye boş satır renklendirme gösterimi de yapılmıştır.

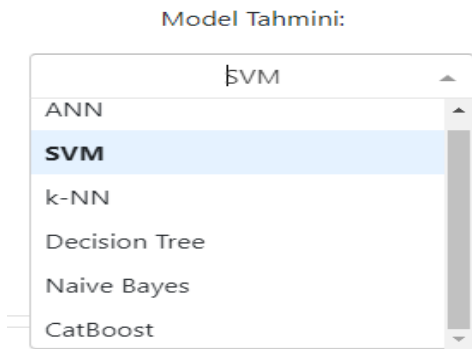
Uygulama üzerine kolaylıkla veri analizi yapılmaktadır.



Şekil 3.71 Uygulama genel bakış veri analizi.

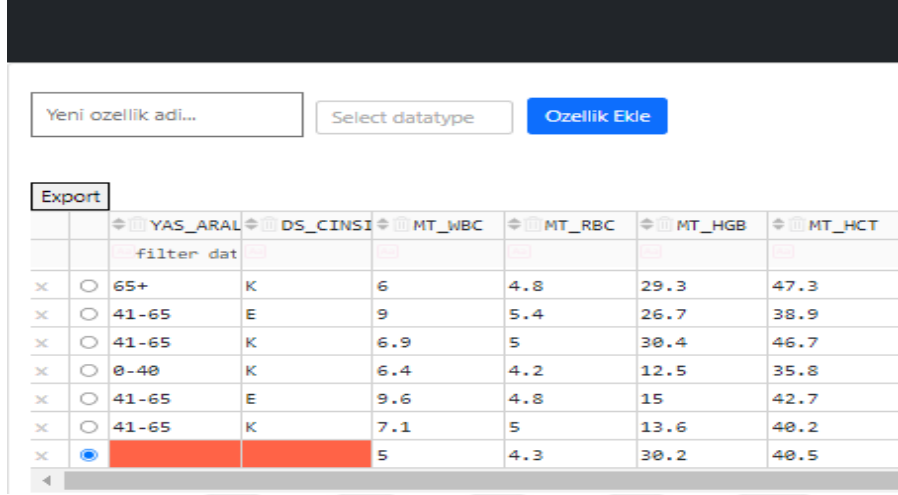
Şekil 3.72. Boş veri renklendirme gösterimi.

MEDIC-AI uygulamasında kullanıcı yeni gelen bir verinin tahmini ve bu tahminden seçilen algoritmaya göre, yukarıda belirttiğimiz gibi, ROC ve PR eğrilerini, karmaşıklık matrisini ve sınıflandırma raporunu görebilir ve inceleyebilir.



Şekil 3.73. Uygulamada sunduğumuz makine öğrenme algoritmaları.

Yeni bir hastadan gelen verileri tabloda “Yeni örneklem ekle” butonunu basarak verilerimizi tablo üzerine girebiliriz. Veri girişini bitirdikten sonra tabloda bulunan “radio butonunu” seçerek o veriyi tahminini yapabiliriz.



Yeni özellik adı... Select datatype Özellik Ekle

Export

		YAS_ARAL	DS_CINSI	MT_WBC	MT_RBC	MT_HGB	MT_HCT
		filter dat					
x	<input type="radio"/>	65+	K	6	4.8	29.3	47.3
x	<input type="radio"/>	41-65	E	9	5.4	26.7	38.9
x	<input type="radio"/>	41-65	K	6.9	5	30.4	46.7
x	<input type="radio"/>	0-40	K	6.4	4.2	12.5	35.8
x	<input type="radio"/>	41-65	E	9.6	4.8	15	42.7
x	<input type="radio"/>	41-65	K	7.1	5	13.6	40.2
x	<input checked="" type="radio"/>			5	4.3	30.2	40.5

Şekil 3.74. Yeni gelen verini tahmini için seçimi.



Model Tahmini:

SVM

SONUC:

DIPPER

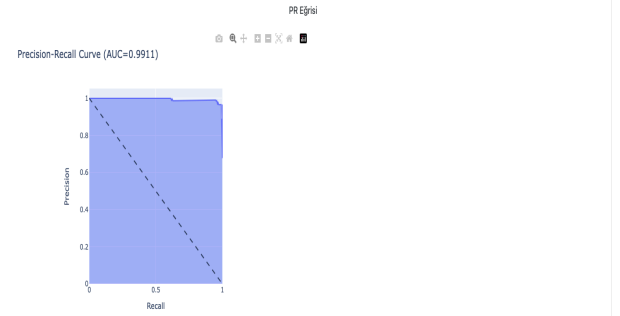
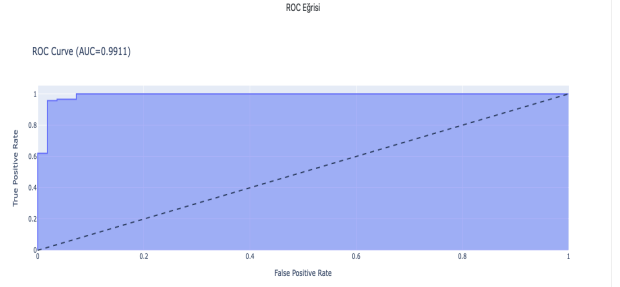
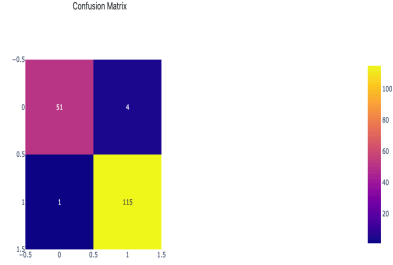
Tahmin Yap

Şekil 3.75. SVM algoritmayla tahmini sonuç gösterimi.

Model tahminini yaptıktan sonra “Model Tahmin Sonuçları” butonunu basarak uygulamanın sayfa sonunda bize tahminleme yapılan seçili makine öğrenme algoritmanın sonuçlarını göstermektedir.

Classification Report

Index	precision	recall	f1-score	support
1	0.8807692207692307	0.9272727272727272	0.9532710280373831	55
2	0.9663865546218487	0.9913793103448276	0.9787234042553192	116
accuracy	0.97076402339181286	0.97076402339181286	0.97076402339181286	0.97076402339181286
macro avg	0.9735778926953397	0.9593260188087774	0.9659972161463511	171
weighted avg	0.9710125615688371	0.97076402339181286	0.9705369674600767	171



Şekil 3.76. CatBoost algoritmanın tahmini sonuçları.

4. SONUÇLAR

Kardiyovasküler hastalıklar (KVH) insan sağlığı için çok riskli ve ölümcül bir hastalıktır. Dünya genelinde kardiyovasküler hastalığı (KVH) dünyada önde gelen ölüm nedenidir ve KVH'nin ölüm oranları yıldan yıla artmaktadır. Her geçen gün sağlık ve bilişim alanlarında Kardiyovasküler hastalıklarını önüne geçebilmesi ya da teşhis konabilmesi için çalışmalar yapılmaktadır. Dünya genelinde yapılan araştırmalar ve yazılan makalelere bakılarak, sağlık alanında çok sayıda makine öğrenmesi, büyük veri teknikleri, derin öğrenme gibi uygulamalar yapılmıştır ve halen devam etmektedir. Bunun için farklı makine öğrenmesi algoritmaları, veri madenciliği yöntemleri ya da derin öğrenme teknikleri kullanılarak kan basıncı, hipertansiyon gibi kardiyovasküler hastalıklarının teşhisini ve analizini büyük ölçüde destek olmaktadır. Yapılan çoğu karar destek sistemleri (AI veya ML destekli), gerek sağlık personellerine gerek ise hastalara yardımcı olmayı amaçlanmaktadır.

Bunu göz önünde bulundurarak bu tez çalışması ve bu teze bağlı geliştirilen MEDIC-AI ML destekli web uygulama, farklı hastalıklar üzerine araştırılmış ve yapılmıştır. Uygulamamızda farklı anonim veri setler kullanılmıştır. Geliştirilen MEDIC-AI ML destekli web uygulaması sadece kardiyovasküler hastalıklara bağlı olarak geliştirilmedi. MEDIC-AI ML destekli web uygulaması, dosya yükleme özelliği ile, yüklenen herhangi (.csv veya .xlsx formatındaki) bir dosya farklı hastalıklar teşhisi de yapılmaktadır. Bunun yanında eklenen birden fazla makine öğrenme teknikleri özelliği ile, farklı ML algoritmalarından tahmini sonuç alma ve sunulan sınıflandırma raporları, PR ve ROC eğrileri ve karmaşıklık matrisi ile kıyaslama yapılmaktadır. Böylece kullanıcı tarafından istenilen ML algoritmalarını seçerek tahmini sonuçlar elde edilmektedir. Yukarıda bölümlerde de anlatıldığı gibi yüklenen dosyalar uygulama üzerine tablo halinde gösterebilmektedir. Ayrıca uygulama üzerine sunulan grafikler çevrimiçi olarak veri analizi kolaylıkla yapılmaktadır. Plotly'nin sunduğu web üzerine ki grafiksel yapısı, yapılacak veri analizini kolay anlaşılabilir. Bu tez çalışma kapsamında geliştirilen MEDIC-AI ML destekli web uygulama, beta versiyonunda olmaktadır.

Beta versiyonunda kullanılan ML algoritmalar varsayılan değerlerle beraber kullanılmışlardır. Öyleki test ve eğitim veri seti dağıtımında varsayılan değerler olarak, %30 test, %70 eğitim olarak belirlenmiştir. Kullanılan ML algoritmalar ise varsayılan değerler olarak ise aşağıdaki gibi kullanılmıştır.

- SVM: `SVC(kernel='poly', degree=3)`
- ANN: `MLPClassifier(hidden_layer_sizes=(50, 50, 50))`
- K-NN: `KNeighborsClassifier(n_neighbors=3)`
- DT: `DecisionTreeClassifier(criterion='entropy', random_state=0)`
- Naive Bayes: `GaussianNB()`
- CatBoost: `CatBoostClassifier(iterations=5, learning_rate=0.1)`

Farklı veri setleri kullanarak çıkan sonuçlarının doğruluk ya da kesinlik oranları aşağıdaki tablolarda gösterilmektedir.

Tablo 4.1. hastaveri_ adındaki tablonun tahmini sonuçlarının ML algoritmalarının doğruluk oranları.

ML Adı	Precision	Recall	F1-score	Support	
ANN	0.68	0.68	0.68	0.68	Accuracy
SVM	0.72	0.72	0.72	0.72	Accuracy
K-NN	0.6	0.6	0.6	0.6	Accuracy
DT	0.68	0.68	0.68	0.68	Accuracy
NB	0.6	0.6	0.6	0.6	Accuracy
CatBoost	0.68	0.68	0.68	0.68	Accuracy

Tablo 4.2. cancer_ adındaki tablonun tahmini sonuçlarının ML algoritmaların doğruluk oranları.

ML Adı	Precision	Recall	F1-score	Support	
ANN	0.96	0.97	0.97	0.97	Accuracy
SVM	0.91	0.91	0.91	0.91	Accuracy
K-NN	0.97	0.97	0.97	0.97	Accuracy
DT	0.73	0.73	0.73	0.73	Accuracy
NB	0.92	0.92	0.92	0.92	Accuracy
CatBoost	0.95	0.95	0.95	0.95	Accuracy

Yapılan test çalışmalarında ML algoritmaların farklı anonim veri setlerinde farklı doğruluk oranları göstermektedir. Uygulamaya yüklenen veri seti ne kadar büyük ise (örneklem sayısı fazla ise) ML algoritmaları daha iyi çalışıp daha iyi sonuçlar vermektedir. İki (2) tabloyu karşılaştırsak **cancer_** adındaki tablo **hastaveri_** adındaki tablodan daha iyi sonuçlar vermektedir. Bunun nedeni ise bu tablonun örneklem sayısı 500'den fazla olmasıdır ve her **target label** için oldukça fazla örnekler barındırmaktadır.

Uygulama üzerinde yeni data girişi yaparak tahminleme ve sınıflandırma raporu çıkartılmıştır.

Tablo 4.3. hastaveri_ tablosunda yapılan yeni verinin doğruluk oranları.

ML Adı	Precision	Recall	F1-score	Support	
ANN	0.5	0.5	0.5	0.5	Accuracy
SVM	0.6538	0.6538	0.6538	0.6538	Accuracy
K-NN	0.5	0.5	0.5	0.5	Accuracy

Tablo 4.4. cancer_ tablosunda yapılan yeni verinin doğruluk oranları.

ML Adı	Precision	Recall	F1-score	Support	
ANN	0.95	0.95	0.95	0.95	Accuracy
SVM	0.91	0.91	0.96	0.91	Accuracy
K-NN	0.97	0.97	0.97	0.97	Accuracy

MEDIC-AI ML destekli karar destek Web uygulamamız deneme sürümünde olduğundan dolayı bazı algoritmalar ve değerler varsayılan olarak verilmiştir. Öyleki uygulamada daha fazla ML algoritmaları eklenecektir. ML algoritmalarına varsayılan değerler değil kullanıcı tarafından değer girişi yapılacaktır. Ör:

K-NN algoritmanın K komşuluk sayısını şu an ki versiyonunda uygulamamızda 3 olarak belirlenmiştir, ama bir sonraki versiyonunda K değeri kullanıcı belirtecektir. Ayrıca ML algoritmalarının eğitim ve test ayırımını yine de kullanıcı tarafından verilecektir. ML algoritmalar daha da geliştirilecek, uygulamaya farklı grafik türleri gibi farklı yeni özellikler eklenmeye planlanmaktadır.

KAYNAKLAR

- [1] Bohr A, Memarzadeh K. The rise of artificial intelligence in healthcare applications. *Artificial Intelligence in Healthcare*. 2020;25–60.
- [2] Basu K, Sinha R, Ong A, Basu T. Artificial Intelligence: How is It Changing Medical Sciences and Its Future? *Indian J Dermatol*. 2020 Sep-Oct;65(5):365-370.
- [3] Miller D.D., Brown E.W. Artificial intelligence in medical practice: the question to the answer? *Am J Med*. 2018;131(2):129–133.
- [4] Kirch D.G., Petelle K. Addressing the physician shortage: the peril of ignoring demography. *JAMA*. 2017;317(19):1947–1948.
- [5] Combi C., Pozzani G., Pozzi G. Telemedicine for developing countries. *Appl Clin Inform*. 2016;07(04):1025–1050.
- [6] Schaefer, J., Lehne, M., Schepers, J. et al. The use of machine learning in rare diseases: a scoping review. *Orphanet J Rare Dis* 15, 145 (2020).
- [7] Douibi, K., Settouti, N., Chikh, M.A. et al. An analysis of ambulatory blood pressure monitoring using multi-label classification. *Australas Phys Eng Sci Med* 42, 65–81 (2019).
- [8] Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*. 2016; 316:2402–10.
- [9] Greenfield, D., & Wilson, S. (2019). Artificial intelligence in medicine: Applications, implications and limitations. *Science in the News*.
- [10] Topol, E.J. High-performance medicine: the convergence of human and artificial intelligence. *Nat Med* 25, 44–56 (2019). <https://doi.org/10.1038/s41591-018-0300-7>
- [11] Chotruangnapa, C., Tansakun, T. & Roubsanthisuk, W. Clinical risk factors and predictive score for the non-dipper profile in hypertensive patients: a case-control study. *Clin Hypertens* 27, 22 (2021). <https://doi.org/10.1186/s40885-021-00180-4>

[12] Haq IU, Chhatwal K, Sanaka K, Xu B. Artificial Intelligence in Cardiovascular Medicine: Current Insights and Future Prospects. *Vasc Health Risk Manag.* 2022;18:517-528

<https://doi.org/10.2147/VHRM.S279337>

[13] Haq IU, Haq I, Xu B. Artificial intelligence in personalized cardiovascular medicine and cardiovascular imaging. *Cardiovasc Diagn Ther.* 2021;11:911–923. doi:10.21037/cdt.2020.03.09

[14] Güneş ED, Yaman H, Çekyay B, Verter V. Matching patient and physician preferences in designing a primary care facility network? *J Oper Res Soc.* 2017;65:483–96. doi: 10.1057/jors. 2012.71.

[15] I. Tobore, J. Li, L. Yuhang, et al. Deep learning intervention for health care challenges: some biomedical domain considerations *JMIR mHealth uHealth*, 7 (2019), p. e11966, [10.2196/11966](https://doi.org/10.2196/11966)

[16] L.G. Pee, S.L. Pan, L. Cui Artificial intelligence in healthcare robots: a social informatics study of knowledge embodiment *J Assoc Inf Sci Technol*, 70 (2019), pp. 351-369, [10.1002/asi.24145](https://doi.org/10.1002/asi.24145)

[17] M. Ardan, G.B.G. Ferry Fadzlul Rahman The influence of physical distance to student anxiety on Covid-19, Indonesia *J Crit Rev*, 7 (2020), pp. 1126-1132, [10.31838/jcr.07.17.141](https://doi.org/10.31838/jcr.07.17.141)

[18] Riaño, D., Wilk, S., & ten Teije, A. (2019). *Artificial Intelligence in Medicine*. Springer International Publishing.

[19] Matheny, M., Israni, S. T., Ahmed, M., & Whicher, D. (2019). *Artificial intelligence in healthcare: The hope, the hype, the promise, the peril*. Washington, DC: National Academy of Medicine

[20] Bajwa, J., Munir, U., Nori, A., & Williams, B. (2021). Artificial intelligence in healthcare: transforming the practice of medicine. *Future Healthcare Journal*, 8(2), e188.

[32] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260.

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

- [22] Kersting, K. (2018). Machine learning and artificial intelligence: two fellow travelers on the quest for intelligent behavior in machines. *Frontiers in big Data*, 1, 6.
- [23] Frankenfield, J. (2022, September 12). Artificial Intelligence: What it is and how it is used. Investopedia. Retrieved November 9, 2022, from <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp>
- [24] By: IBM Cloud Education. (n.d.). What is Artificial Intelligence (AI)? IBM. Retrieved November 7, 2022, from <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>
- [25] Burns, E., Laskowski, N., & Tucci, L. What is artificial intelligence (AI)? Definition, benefits and use cases. SearchEnterpriseAI. <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence> (accedido el 3 de agosto de 2022).
- [26] Gui, C., & Chan, V. (2017). Machine learning in medicine. *University of Western Ontario Medical Journal*, 86(2), 76-78.
- [27] Rajula HSR, Verlatto G, Manchia M, Antonucci N, Fanos V. Comparison of Conventional Statistical Methods with Machine Learning in Medicine: Diagnosis, Drug Development, and Treatment. *Medicina*. 2020; 56(9):455. <https://doi.org/10.3390/medicina56090455>
- [28] Char DS, Shah NH, Magnus D. Implementing Machine Learning in Health Care - Addressing Ethical Challenges. *N Engl J Med*. 2018 Mar 15;378(11):981-983. doi: 10.1056/NEJMp1714229. PMID: 29539284; PMCID: PMC5962261.
- [29] Nevin, L., & PLOS Medicine Editors. (2018). Advancing the beneficial use of machine learning in health care and medicine: toward a community understanding. *PLoS Medicine*, 15(11), e1002708.
- [30] Ngiam, K. Y., & Khor, W. (2019). Big data and machine learning algorithms for health-care delivery. *The Lancet Oncology*, 20(5), e262-e273.

[31] Van, R. G. (2009). the history of python: A brief timeline of python. Blogger. Accessed March, 5, 2021.

[32] Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.

[33] Raschka S, Patterson J, Nolet C. Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. Information. 2020; 11(4):193. <https://doi.org/10.3390/info11040193>

[34] Cordny NederkoornCordny Nederkoorn is a software testing and content marketing consultant with over 10 years of experience in finance. (2021, April 15). Top 10 python packages for Machine Learning. ActiveState. Retrieved December 5, 2022, from <https://www.activestate.com/blog/top-10-python-machine-learning-packages/>

[35] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

[36] Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd.

[37] Getting. Getting started with plotly in Python. (n.d.). Retrieved November 26, 2022, from <https://plotly.com/python/getting-started/>

[38] Machlis, S. (2013, November 6). Beauty and brains: Plotly combines dataviz and serious statistical analysis. Computerworld. Retrieved December 21, 2022, from <https://web.archive.org/web/20131202232439/http://blogs.computerworld.com/business-intelligenceanalytics/23078/beauty-and-brains-plotly-combines-dataviz-and-serious-statistical-analysis-cloud>

[39] Kudale, H. S., Phadnis, M. V., Chittar, P. J., Zarkar, K. P., & Bodhke, B. K. DATA ANALYSIS AND VISUALIZATION OF OLYMPICS USING PYSPARK AND DASH-PLOTLY.

- [40] Balzer, C., Oktavian, R., Zandi, M., Fairen-Jimenez, D., & Moghadam, P. Z. (2020). Wiz: A web-based tool for interactive visualization of big data. *Patterns*, 1(8), 100107.
- [41] Hossain, S., Calloway, C., Lippa, D., Niederhut, D., & Shupe, D. (2019). Visualization of bioinformatics data with dash bio. In *Proceedings of the 18th Python in Science Conference* (pp. 126-133).
- [42] what is postgresql? PostgreSQL Documentation. (2022, November 10). Retrieved December 21, 2022, from <https://www.postgresql.org/docs/current/intro-what-is.html>
- [43] PostgreSQL. (n.d.). Retrieved November 26, 2022, from <https://www.postgresql.org/about/>
- [44] What is Heroku. Heroku. (n.d.). Retrieved December 21, 2022, from <https://www.heroku.com/what>
- [45] Classification algorithm in Machine Learning - Javatpoint. www.javatpoint.com. (n.d.). Retrieved December 5, 2022, from <https://www.javatpoint.com/classification-algorithm-in-machine-learning>
- [46] Regression algorithms - overview. Tutorials Point. (n.d.). Retrieved December 5, 2022, from https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_regression_algorithms_overview.htm
- [47] "Destek Vektör Makineleri (Support Vector Machine)." *Veri Bilimcisi*. September 22, 2017. Accessed May 08, 2019. <https://veribilimcisi.com/2017/07/19/destek-vektor-makineleri-support-vector-machine/>.
- [48] Patel, Savan, and Savan Patel. "Chapter 2 : SVM (Support Vector Machine) - Theory." *Medium*. May 03, 2017. Accessed May 07, 2019. <https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72>.
- [49] Brooks, Gabriel A., Nancy Lynn Keating, Savannah L. Bergquist, Mary Beth Landrum, and Sherri Rose. "Classifying lung cancer stage from health care claims with a clinical algorithm or a machine-learning approach." (2018): 6589-6589.
- [50] GÜZEL, Kadir, and Kadir GÜZEL. "Geri Yayılımlı Çok Katmanlı Yapay Sinir Ağları-1." *Medium*. March 01, 2018. Accessed May 07, 2019.

<https://medium.com/@billmuhh/geri-yayılımlı-çok-katmanlı-yapay-sinir-ağları-1-47daa3856247>.

[51] Psikolojik.gen.tr. "Yapay Sinir A.", Yapay Sinir Ağlar Nasıl Oluşur. Accessed May 07, 2019. <https://www.psikolojik.gen.tr/yapay-sinir-aglari.html>.

[52] "Yapay Sinir Ağları (YSA) Nedir ?" Kod5.org. February 11, 2016. Accessed May 07, 2019. <http://kod5.org/yapay-sinir-aglari-ysa-nedir/>.

[53] Pandey, A., & Jain, A. (2017). Comparative analysis of KNN algorithm using various normalization techniques. International Journal of Computer Network and Information Security, 9(11), 36.

n_algorithms_overview.htm

[54] Zhang Z. Introduction to machine learning: k-nearest neighbors. Ann Transl Med. 2016 Jun;4(11):218. doi: 10.21037/atm.2016.03.37. PMID: 27386492; PMCID: PMC4916348.

[55] Zhang Z. Too much covariates in a multivariable model may cause the problem of overfitting. J Thorac Dis 2014;6:E196-7.

[56] Ghosh, A. K. (2006). On optimum choice of k in nearest neighbor classification. Computational Statistics & Data Analysis, 50(11), 3113-3123.

[57] Song YY, Lu Y. Decision tree methods: applications for classification and prediction. Shanghai Arch Psychiatry. 2015 Apr 25;27(2):130-5. doi: 10.11919/j.issn.1002-0829.215044. PMID: 26120265; PMCID: PMC4466856.

[58] Quinlan, J. R. (1996). Learning decision tree classifiers. ACM Computing Surveys (CSUR), 28(1), 71-72.

[59] Kamble, V. H., & Dale, M. P. (2022). Machine learning approach for longitudinal face recognition of children. In Machine Learning for Biometrics (pp. 1-27). Academic Press.

[60] Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. Knowledge-Based Systems, 192, 105361.

- [61] Vembandasamy, K., Sasipriya, R., & Deepa, E. (2015). Heart diseases detection using Naive Bayes algorithm. *International Journal of Innovative Science, Engineering & Technology*, 2(9), 441-444.
- [62] Zhang, Y., Zhao, Z., & Zheng, J. (2020). CatBoost: A new approach for estimating daily reference crop evapotranspiration in arid and semi-arid regions of Northern China. *Journal of Hydrology*, 588, 125087.
- [63] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.
- [64] Zhang, C., Chen, X., Wang, S., Hu, J., Wang, C., & Liu, X. (2021). Using CatBoost algorithm to identify middle-aged and elderly depression, national health and nutrition examination survey 2011–2018. *Psychiatry Research*, 306, 114261.
- [65] About linear regression. IBM. (n.d.). Retrieved December 6, 2022, from <https://www.ibm.com/topics/linear-regression#:~:text=Resources-,What%20is%20linear%20regression%3F,is%20called%20the%20independent%20variable.>
- [66] Yan, X., & Su, X. (2009). *Linear regression analysis: theory and computing*. world scientific.
- [67] Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression*. John Wiley & Sons. New York.
- [68] What is logistic regression? IBM. (n.d.). Retrieved December 7, 2022, from <https://www.ibm.com/topics/logistic-regression>
- [69] Pelleg, D., & Moore, A. (1999, August). Accelerating exact k-means algorithms with geometric reasoning. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 277-281).
- [70] (LEDU), E. E. (2018, September 12). *Understanding K-means clustering in machine learning*. Medium. Retrieved December 7, 2022, from <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1>

- [71] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8), 861-874.
- [72] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.
- [73] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1), 1-13.
- [74] Chicco, D., Tötsch, N., & Jurman, G. (2021). The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData mining*, 14(1), 1-22.
- [75] Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. In *data democracy* (pp. 83-106). Academic Press.
- [76] Demir, F. (2022). Deep autoencoder-based automated brain tumor detection from MRI data. In *Artificial Intelligence-Based Brain-Computer Interface* (pp. 317-351). Academic Press.
- [77] Singh, P., Singh, N., Singh, K. K., & Singh, A. (2021). Diagnosing of disease using machine learning. In *Machine learning and the internet of medical things in healthcare* (pp. 89-111). Academic Press.
- [78] Sharma, D. K., Chatterjee, M., Kaur, G., & Vavilala, S. (2022). Deep learning applications for disease diagnosis. In *Deep Learning for Medical Applications with Unique Data* (pp. 31-51). Academic Press.
- [79] Kjytay, Kjytay, 7, V. on M., 7, kjytay on M., 8, T. T. on O., 8, kjytay on O., 16, N. S. H. on D., & 16, N. on D. (2020, October 8). *What is balanced accuracy?* Statistical Odds & Ends. Retrieved December 8, 2022, from <https://statisticaloddsandends.wordpress.com/2020/01/23/what-is-balanced-accuracy/>
- [80] Olugbenga, M. (2022, November 14). Balanced accuracy: When should you use it? *neptune.ai*. Retrieved December 8, 2022, from <https://neptune.ai/blog/balanced-accuracy#:~:text=Balanced%20Accuracy%20is%20used%20in,lot%20more%20than%20the%20other.>

[81] Brownlee, J. (2021, January 12). How to use ROC curves and precision-recall curves for classification in Python. MachineLearningMastery.com. Retrieved December 8, 2022, from <https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/>

[82] Sigrist, F. (2022, March 4). Demystifying roc and precision-recall curves. Medium. Retrieved December 8, 2022, from <https://towardsdatascience.com/demystifying-roc-and-precision-recall-curves-d30f3fad2cbf>

[83] Hinton, S. (2020, July 15). PR vs ROC curves - which to use? SRH. Retrieved December 8, 2022, from https://cosmiccoding.com.au/tutorials/pr_vs_roc_curves



ÖZGEÇİMİ

Adı Soyadı: Amir Karaj

Eğitim Bilgileri:

Marmara Üniversitesi- Yüksek Lisans: Bilgisayar Mühendisliği ·

İstanbul Sabahattin Zaim Üniversitesi- Lisans: Bilgisayar Mühendisliği.

Uluslararası Fatih Sultan Mehmet Anadolu İmam Hatip Lisesi

Deneyimler:

Yazılım Mühendisi | Ritech International Ag | 24.08.2020~(Devam) · C#, Asp.Net, .Net Core, Nodejs, Python, Cakephp

Danışmanlık(Part-Time) | İstanbul Sabahattin Zaim Üniversitesi Kütüphane | 04.09.2018~31.05.2019

Stajyer | Türk Hava Yolları | 24.06.2018~03.08.2018 · Müşteri Çözümleri Bşk. (Ücret ve Ödeme Çözümleri Md.)

Gönüllü Stajyer | Gamelab İstanbul | 12.02.2018~02.04.2018

Beceriler Ve Yetenekler:

C#, Asp.Net, .Net Core, Entity Framework, Python, Linux

