İZMİR KÂTİP ÇELEBİ ÜNİVERSİTESİ
2010
FEN BİLİMLERİ ENSTİTÜSÜ

Pima İndians Diabetes Veri Kümesi İle Makine Öğrenmesi Yöntemleri Kullanılarak Sınıflandırılması İşlemi

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Ömür AYCİBİN

ORCID 0009-0003-8007-9512

Proje Danışmanı: Doç. Dr. Aytuğ ONAN

Ocak 2024

Pima İndians Diabetes Veri Kümesi İle Makine Öğrenmesi Yöntemleri Kullanılarak Sınıflandırma İşlemi

Öz

Diyabet dünya genelinde görülme oranı giderek artan, yaygın sağlık sorunlarından biridir. Kronik bir hastalık olan diyabet kontrol altına alınmadığı takdirde göz, kalp, böbrek gibi birçok organda tahribata ve ölümlere neden olabilmektedir. Diyabetin erken teşhisi oluşabilecek komplikasyonları önleme ve yaşam kalitesini arttırma açısından önemlidir. Medikal alanda yaygın kullanılan makine öğrenmesi teknikleri farklı hastalıkların teşhisinde uzmanlar için zeki birer karar destek sistemi rolü üstlenmektedir. Bu çalışma, diyabetin erken teşhisine yönelik olarak 5 farklı makine öğrenmesi tekniği ile PIMA diyabet veri seti üzerinde gerçekleştirilen sınıflama çalışmalarını içermektedir. Sınıflama çalışmalarındaki temel amaç tahmin doğruluğunu arttırmaktır. Bu çalışmada sınıflandırıcıların başarıları arttırmak için veri seti üzerinde 14 farklı yeniden örnekleme yöntemi kullanılmıştır. Her bir makine öğrenmesi modeli için örnekleme olmaksızın ve yeniden örnekleme yapılarak, 90 sınıflama işlemi gerçekleştirilmiştir. Her bir sınıflandırma işleminin başarısı 5 farklı performans metriği ile raporlanmıştır. En başarılı sonuç %96,296 doğrulukla, InstanceHardnessThreshold az örnekleme tekniği ile birlikte Rastgele Orman modelinin kullanıldığı sınıflandırma işleminde elde edilmiştir. Yeniden örnekleme tekniklerinin genel olarak sınıflandırıcıların başarılarını arttırdığı ve kolektif öğrenme yöntemleri ile birlikte kullanıldığında daha başarılı sonuç verdiği görülmüştür.

Anahtar Sözcükler: Diyabet, Teşhis, Sınıflandırma, Makine öğrenmesi, Topluluk öğrenimi

Machine Learning Methods Using Classification Process with Pima İndian Diabetes Data Mileage

Abstract

Diabetes is one of the common health problems whose incidence is increasing worldwide. Diabetes, which is a chronic disease, can cause damage to many organs such as eyes, heart and kidneys and cause death if a control certificate is not obtained. Early diagnosis of diabetes, prevention of its complications and flexibility of life are important. Machine learning techniques, which are widely used in the medical field, serve as intelligent decision support systems for diagnosis in different diagnoses. This study includes existing classification analysis on the PIMA diabetes set with 6 different machine learning techniques for the early diagnosis of diabetes. The main purpose of classification studies is the estimation accuracy. To obtain the success of these possible classifiers, 14 different retrieval methods were used on the data set. For a machine learning model, 90 classification operations were performed without and using the hand again. Each daily throughput capacity is reported with 5 different performance metrics. The most successful result was obtained in the activity operation of the Random Forest model with InstanceHardnessThreshold few crafts, with 96.296% accuracy. It is seen that reprogramming techniques generally increase the success of classifiers and separation together with collective learning methods show more successful results

Keywords: Diabetes, Diagnosis, Classification, Machine learning, Ensemble learning

Teşekkür

Bu araştırmanın gerçekleştirilmesinde değerli bilgilerini paylaşarak bana yol gösteren, karşılaştığım güçlüklerde her zaman tüm desteğiyle yanımda olan, zamanını ve ilgisini hiçbir zaman esirgemeyerek çalışmamla yakından ilgilenen saygıdeğer danışman hocam Doç. Dr. Aytuğ Onan'a teşekkürü bir borç bilirim.

Her konuda bana destek olan, beni yüreklendiren ve bu süreçte de en büyük yardımcım olan sevgili eşim Göksu AYCİBİN'e, yaşından beklenmeyecek bir sabır ve anlayış gösteren moral kaynağım sevgili kızım Öykü AYCİBİN'e, sonsuz teşekkürlerimi sunarım.

İçindekiler

Şekiller Listesi

Tablolar Listesi

Kısaltmalar Listesi

ORCID	Open Researcher and Contributor ID
MÖ	Makine Öğrenmesi
ML	Machine Learning
IOT	Nesnelerin İnterneti (Internet of things)
IDF	Uluslararası Diyabet Federasyonu
OR	Olasılık Oranı
LR	Lojistik Regresyon
SMOTE	Sentetik Azınlık Aşırı-Örnekleme
K-NN	K-Nearest Neighbors
DSÖ	Dünya Sağlık Örgütü
SVM	Support Vector Machine
DP	Doğru Pozitif
DN	Doğru Negatif
YN	Yanlış Negatif
YP	Yanlış Pozitif
TP	True Positive
TN	True Negative
FN	False Negative
FP	False Positive

Semboller Listesi

Eyeni	Öznitelik vektörü
Ei	İncelenen öznitelik vektörü
Ej	İncelenen öznitelik vektörüne en yakın komşu vektör
σ	Standart sapma dağılım varyansı
μ	Örnek ortalama merkezi
z	Standart ölçülendirme yöntemi
x	Standart ölçeklendirme yöntemi
δ	0 ile 1 arasında değişim değeri

Bölüm 1

Giriş

Şeker hastalığı yani tıp dilinde Diabetes Mellitus olarak adlandırılan diyabet hastalığı, kan şekerinin yükselmesine neden olan en ölümcül ve kronik hastalıklardan biri olarak kabul edilmektedir. Kan şekerini düzenleyen insülin hormonunun eksikliği, yeterince kullanılamaması veya üretilememesi durumlarında ortaya çıkmaktadır. Dünya Sağlık Örgütü'nün (DSÖ-World Health Organization) son yıllarda yayınlamış olduğu göstergelere bakıldığında dünya genelinde yaklaşık olarak 422 milyon insanın diyabet hastası olduğu ve her yıl meydana gelen ölümlerin yaklaşık olarak 1.6 milyonun doğrudan diyabete bağlı olduğu açıklanmıştır. Aynı zamanda 2015 ve sonrası Uluslararası Diyabet Federasyonu (IDF) verilerine bakıldığında dünyada 415 milyon birey diyabet hastası iken bu sayının 2040 yılında %55 artarak 642 milyona ulaşacağı tahmin edilmektedir. Bu durum diyabet hastalığının tüm dünya popülasyonuna ve tüm yaş gruplarına yayılmış, yaygın ve hızlı artan bir çeşit hastalık olduğunu göstermektedir. Diyabet hormonal duruma bağlı olmasından dolayı ömür boyu süren hastalıklardan biridir. Şeker hastalığı, başta böbrek fonksiyonları ve tansiyon olmak üzere vücut genelinde ciddi tahribata yol açmaktadır. Hastalığın erken teşhis edilmesi ve tedaviye başlanması yani zamanında tedbir alınması beraberinde ve sonrasında oluşacak diğer hastalıkların önüne geçmek ve engellemek için çok büyük önem arz etmektedir. Son yıllarda teknolojinin gelişmesiyle birlikte özellikle tıbbi teşhis alanında makine öğrenmesi yöntemleri kullanılmaktadır. Makine öğrenimi, herhangi bir insan müdahalesi olmadan verilerden ve analizlerinden daha iyi öğrenmeye yardımcı olan, yaygın olarak büyüyen bir alandır. Ciddi ve karmaşık durumları analiz etmek ve tespit etmek için özellikle sağlık hizmetleri alanında popüler bir şekilde kullanılmaktadır. Makine öğrenmesinde kullanılan sınıflandırma algoritmaları yüksek oranda doğruluk sonuçları vermektedir. Bu durum daha hızlı karar verme ve hekimlere yardımcı olma açısından çok önemlidir. Diğer bütün hastalıklarda olduğu gibi şeker hastalığının erken teşhis ve tedavi süreci hayat kurtarmakla birlikte kişilerin yaşam kalitesini daha iyi hale getirmektedir. Literatüre baktığımızda Diyabet hastalığın

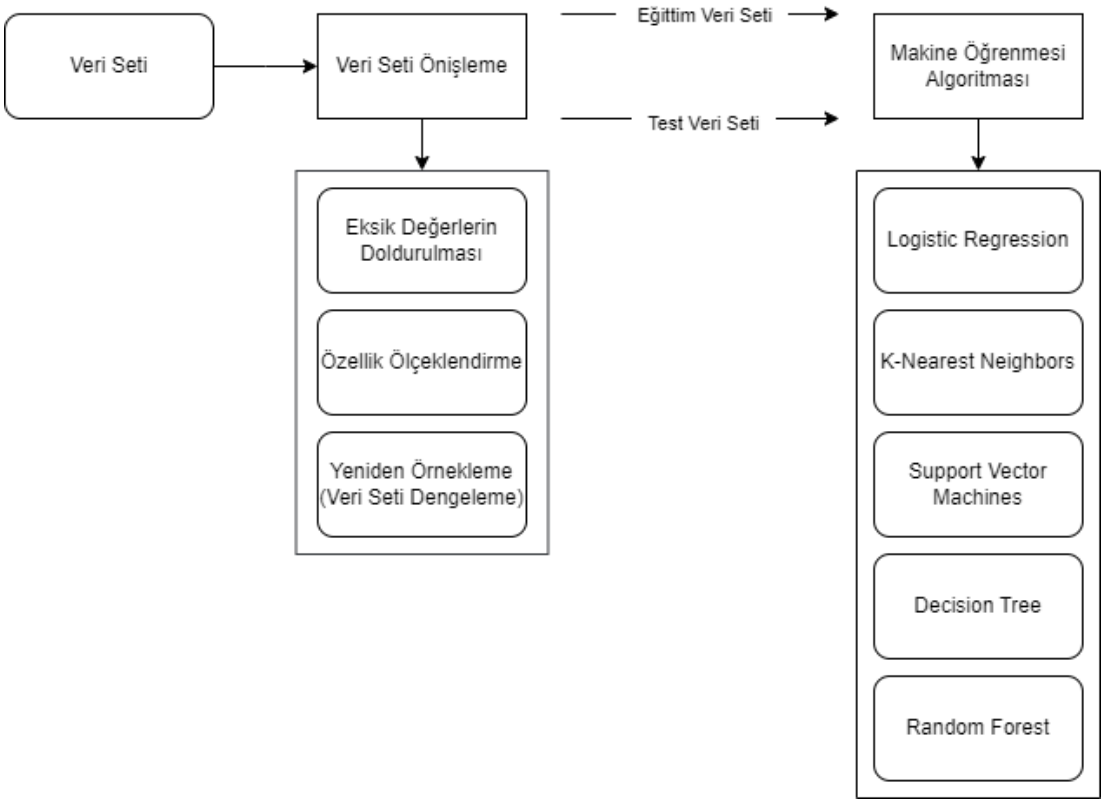
makine öğrenmesi yöntemleri kullanılarak sınıflandırılması için çok farklı algoritma ve yöntemler kullanılmıştır. Bunun yanı sıra farklı veri setleri de kullanılmıştır. En yaygın kullanılanlardan biri de "Pima Indians Diabetes" veri seti kümesidir. Bu veri seti kullanılarak yapılan çalışmalardan biri olan şeker hastalığın tahmin edilmesi için Lojistik Regresyon, K- En Yakın Komşu, Destek Vektör Makineleri, Karar Ağacı, Rastgele Orman makine öğrenmesi sınıflandırma algoritmaları kullanılmıştır. Sınıflandırma algoritmaları karşılaştırıldığında %89 doğruluk oranıyla en iyi, sonucu Rastgele Orman Algoritması vermiştir. Diğer bir çalışmada yine aynı veri seti üzerinde Lojistik Regresyon, K- En Yakın Komşu, Destek Vektör Makineleri, Karar Ağacı, Rastgele Orman olmak üzere beş farklı makine öğrenmesi yöntemi kullanılarak diyabet hastalığı erken evrede teşhis edilmeye çalışılmıştır. Kullanılan algoritmaların performans değerlendirmesi Hassasiyet, Doğruluk, F-Skore ve Eğri Altında Kalan Alan metrikleri kullanarak yapılmıştır. Elde edilen sonuçlara göre % 79 doğruluk oranıyla en iyi performansı Lojistik Regresyon ve Rastgele Orman algoritması vermiştir. Aynı veri seti kullanılarak yapılan çalışmalardan bir diğeri Çok Katmanlı Yapay Sinir Ağları, Radyal Temel Fonksiyonu ve Genel Regresyon Sinir Ağı olmak üzere diyabet hastalığının teşhis edilmesinde üç farklı sinir ağ yapısı kullanmıştır. En iyi sonucu %81 oranıyla Genel Regresyon Sinir Ağları vermiştir. Bir diğer çalışmada Bangladeş Sylhet Diyabet Hastanesin'den elde edilen veriler üzerinde 5 farklı makine öğrenmesi yöntemi kullanılmıştır. Çalışma kapsamında kullanılan Çok Katmanlı Algılayıcı Yapay Sinir Ağları, Destek Vektör Makinaları, Karar Ağaçları, Topluluk Öğrenme Algoritmaları, Doğrusal Ayrımcı Analizi ve k-NN metotları arasında en iyi sonucu %99,81 doğruluk oranı ile k-NN algoritması vermiştir. Bir diğer çalışmada Ulusal Sağlık ve Beslenme İnceleme Anketinden (National Health and Nutrition Examination Survey) elde edilen 2009–2012 yıllarında yürütülen diyabet veri setini kullanılmıştır. Lojistik regresyon (LR), diyabet hastalığı için risk faktörlerini p değeri ve olasılık oranına (OR) dayalı olarak belirlemek için kullanılmıştır. Diyabetik hastaları tahmin etmek için Lojistik Regresyon, K- En Yakın Komşu, Destek Vektör Makineleri, Karar Ağacı, Rastgele Orman algoritmaları sınıflandırma modeli olarak kullanılmıştır. Kullanılan sınıflandırma algoritmalarının performansları, doğruluk oranı ve eğri altındaki alan kullanılarak değerlendirilmiş ve en iyi sonucu %94.25 doğruluk oranı ile Rastgele Orman algoritması vermiştir. Bu çalışmanın temel amacı, diyabet hastalığının teşhis edilmesi için farklı makine öğrenmesi sınıflandırma

algoritmaları yaklaşımları kullanılmasıdır. Çalışmada kullanılan veri setim Ulusal Diyabet ve Sindirim ve Böbrek Enstitüsü'nden (National İnstitute of Diabetes and Digestive and Kidney Diseases) alınan, 21 yaş ve üstü kadınlar için olan Pima Indians Diyabet veri seti kümesidir. Kullanılan veri seti içerisinde diyabet hastalığının teşhisinde yer alan belirli tanısal ölçümlere dayalı veriler bulunmaktadır. 768 kayıtlı ve 9 özniteliğe sahip olan veri seti üzerinde Lojistik Regresyon, K- En Yakın Komşu, Destek Vektör Makineleri, Karar Ağacı, Rastgele Orman sınıflandırma algoritmaları kullanılarak diyabet hastalığının teşhisinde en iyi ve doğru sonucun elde edilmesi amaçlanmıştır. Kullanılan sınıflandırıcıların performanslarını artırmak için veri setinde içerisindeki eksik değerler çarpıklık durumuna göre tekrar yapılandırılmış, veri standardizasyon standart ölçeklendirme kullanılarak yapılmıştır. Aynı zamanda sınıf dengesizlik probleminin sınıflandırma üzerindeki olumsuz etkisini azaltmak için Sentetik Azınlık Aşırı-Örnekleme (SMOTE) tekniği kullanılmıştır. Çalışma kapsamında oluşturulan sınıflandırıcıların değerlendirme kriterleri Doğruluk Oranı (Accuracy Rate), Kesinlik (Precision), Duyarlılık (Recall) ve F1-Skore (F1 Score) değerleri kullanılarak hesaplanmıştır. Bu sonuçlara göre %88 doğruluk oranı ile en iyi sunucu Destek Vektör Makineleri vermiştir. Çalışmanın ilerleyen bölümleri şu şekilde düzenlenmiştir. Bölüm 2, çalışma süresince kullanılan veriler üzerinde uygulanan yöntemler ve analizler hakkında ayrıntılı bilgi verilmiştir. Bölüm 3'te, kullanılan makine öğrenmesi algoritmaları hakkında kısa açıklamalar bulunmaktadır. Bölüm 4, Python kodlama dili kullanarak hazırlanan sınıflandırma algoritmalarının uygulanması ve performans değerlendirme ölçütlerine yer verilmiştir. Çalışmanın son kısmı olan Bölüm 5'te açıklamalar ve sonuçlar bulunmaktadır.

Bölüm 2

Materyal ve Metot

Bu bölüm yapılmış olan çalışmayı gerçekleştirmek için kullanılan yaklaşımları açıklayan metodolojiyi içerir. Çalışma için kullanılan model diyagramı ana hatlarıyla Şekil 2.1'de gösterilmiştir.



Şekil 2.1: Kullanılan model diyagramı.

2.1 Diyabet Veri Seti

Çalışma kapsamında kullanılan veri seti Ulusal Diyabet ve Sindirim ve Böbrek Enstitüsü'nden (National İnstitute of Diabetes and Digestive and Kidney Diseases) alınan, 21 yaş ve üstü kadınlar için olan Pima Indians diyabet veri setidir. Kullanılmış olan veri setinin amacı, bir hastanın diyabetli olup olmadığını, veri setine dâhil edilen

belirli tanısal ölçütlere dayanarak tahmin etmektir. Kullanılan veri seti 268'i diyabet hastası, 500'ü diyabet hastası olmayan toplam 768 kayıttan ve 9 nitelikten oluşmaktadır. (8 öznitelik ve 1 sınıf değişkeni). Veri setinde bulunan niteliklere ait bilgiler Tablo 2.1'de ayrıntılı olarak verilmiş.

Sayı	Nitelik	Açıklama
X1	Gebelik (Pregnancies)	Hamile kalma sayısı
X2	Glikoz (Glucose)	Plazma glukoz konsantrasyonu (2 saat oral glukoz tolerans testi)
X3	Kan Basıncı (Tansiyon) (Blood Pressure)	Kan Basıncı (mm/Hg)
X4	Cilt Kalınlığı (Skin Thickness)	Deri Kıvrım Kalınlığı (mm)
X5	İnsülin (Insulin)	2 saatlik insülin serum (mu U/ml)
X6	Vücut kitle indeksi (BMI)	Vücut Kitle İndeksi (kg ve m2)
X7	Genetik Diyabet Yatkınlık	Genetik olarak Diyabet hastalığına yatkınlık durumu
X8	Yaş	Kişinin yaşı (yıl)
Y	Sonuç	Sınıf Değişkeni (0-1)

Tablo 2.1: Veri Setinde bulunan özellikler

Veri setinin ilk 5 verisi Tablo 2.2'de gösterilmiştir.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Tablo 2.2: Veri setinde bulunan ilk 5 kayıt

Veri setinin şekil, kolonları ve veri tipleri Tablo 2.3' de gösterilmiştir.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Tablo 2.3: Veri setinin şekil, kolonları ve veri tipleri

Veri ön işleme için tekrar eden verilerin silinmesi, null verilerin 0 a çevrilmesi işlemi yapılmıştır. Değeri 0 olan veriler bulunduğu sütunun medyan değeri ile değiştirilmiştir.

```
df1['Glucose']=df1['Glucose'].replace(0,df1['Glucose'].mean())
df1['BloodPressure']=df1['BloodPressure'].replace(0,df1['BloodPressure'].mean())
df1['SkinThickness']=df1['SkinThickness'].replace(0,df1['SkinThickness'].median())
df1['Insulin']=df1['Insulin'].replace(0,df1['Insulin'].median())
df1['BMI']=df1['BMI'].replace(0,df1['BMI'].median())
```

Şekil 2.2: Değeri sıfır olan verilere median değerinin atanması

Dataframe sütunlarının histogramı Şekil 2.4'de verilmiştir. Bu işlem için matplotlib.pyplot.hist fonkisyonu kullanılmıştır.Parametrelerden

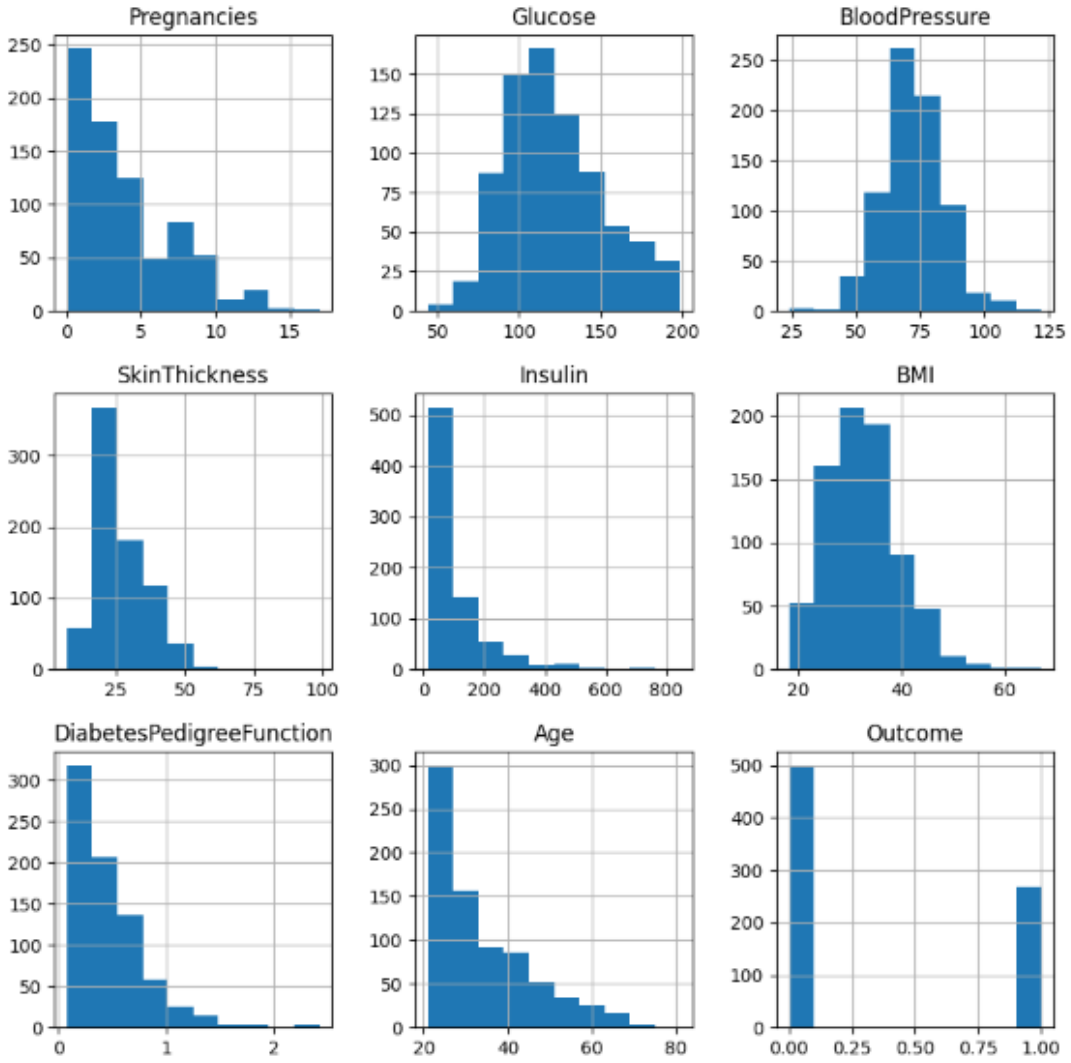
Figsize: oluşturulacak grafiğin inç cinsinden boyutunu belirtir.Parantez içindeki ilk değer grafiğin eni, ikinici değer boyutudur.

Show : Grafik çizdirmeyi ve ekranda gözükmesi sağlıyor.

Bins: Histogramdaki kutu sayısını belirtir.

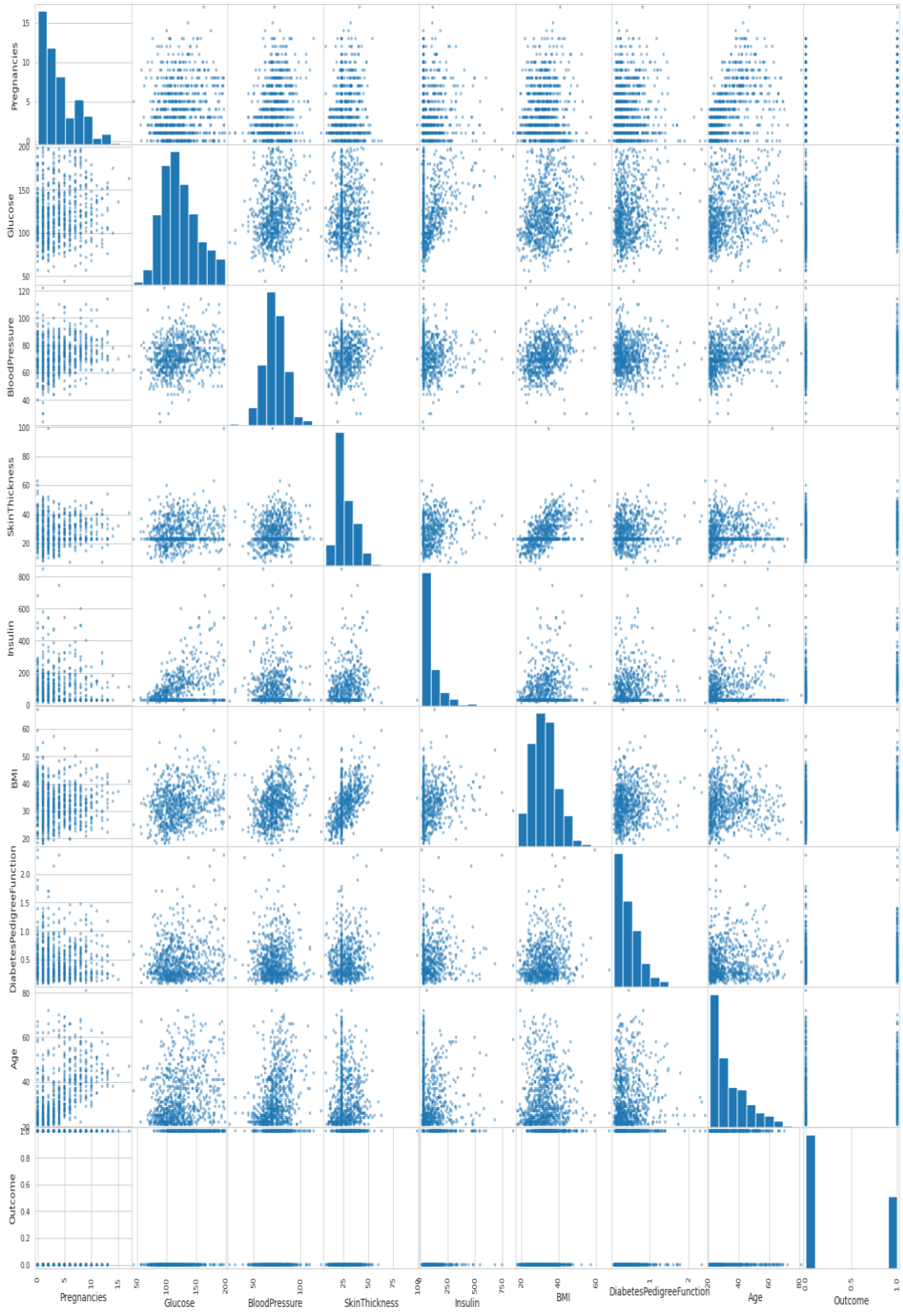
Histtype: Histogramın tipinin belirlememizi sağlar. ('bar', 'barstacked', 'step)

Orientation: Grafiğin yatay ya da dikey çizilmesini belirler.



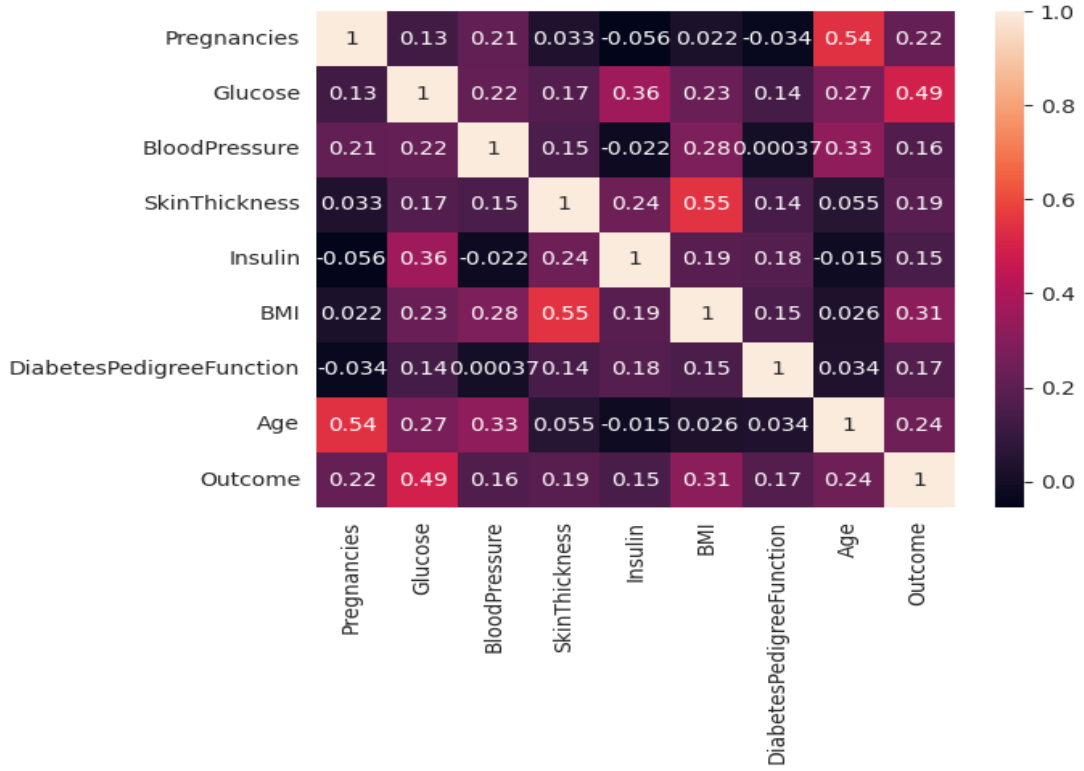
Şekil 2.3: Veri seti histogramı

Dataframe sütunlarının dağılım grafiği Şekil 2.4'de verilmiştir. Bu işlem için scatter_matrix fonkisyonu kullanılmıştır.



Şekil 2.4: Veri seti dağılım grafiği

Dataframe de yer alan sütunların ikili ilişkileri hesaplatılıp, dikdörtgen verileri renk kodlu bir matris Şekil 2.5'de verilmiştir. (ısı haritası)



Şekil 2.5: Veri seti ısı haritası

2.2 Veri Seti Önişleme

Verileri seti üzerinde yapılacak olan ön işlenme, çalışma kapsamında kullanılan makine öğrenmesi algoritmalarının sınıflandırma performans değerlerini artırmak ve en iyi sonucu elde etmek için uygulanan önemli adımlardan biridir. Makine öğrenmesi yöntemlerinde kullanılan algoritmanın performansı, kullanılan veriler arasındaki korelasyona, veri seti içerisinde eksik veya aykırı değere bağlı olarak değişmektedir. Bu çalışmada kullanılan veri seti düzenlenme işlemi üç aşamadan oluşmaktadır.

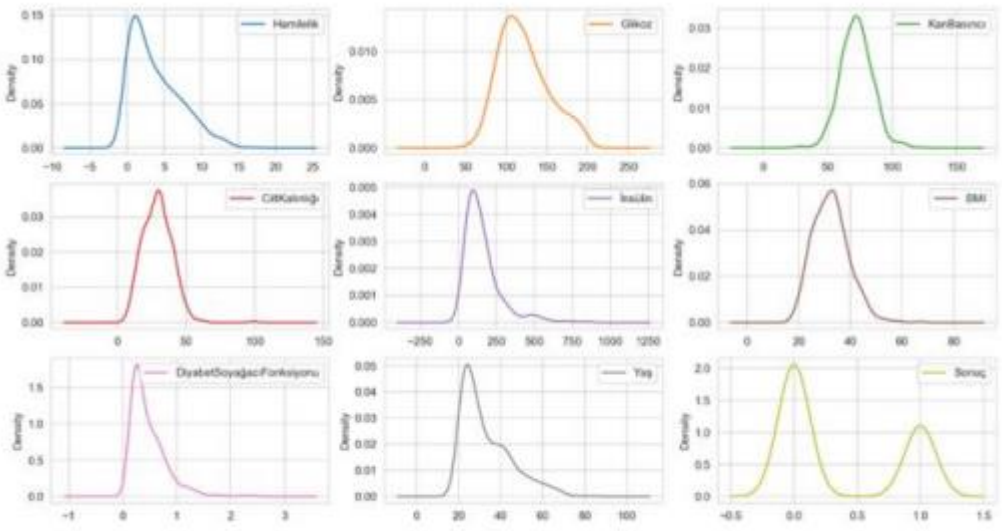
1) NaN yani eksik değerlerin doldurulması.

2) Özellik Ölçeklendirme (Feature Scaling)

3) Yeniden Örnekleme (Veri Seti Dengeleme)

Kullanılan veriler içerisinde eksik değerler (NaN) bulunmamasına rağmen Tablo 2.4 ‘ye baktığımızda Glikoz, İnsülin, Vücut Kitle İndeksi, Kan Basıncı ve Deri Kalınlığı gibi özelliklerinin min () değerinin “0” olduğu görülmektedir. İnsan vücudu ve

anatomisi göz önüne alındığında ve düşünüldüğünde bu değerlerin "0" olamayacağı, aslında bu değerlerin eksik değerler olduğu görülmektedir. Eksik değerlerin doldurulması için kullanılan çeşitli yöntemler bulunmaktadır. Bunlardan biri de eksik değerlerin Çarpıklık (Skewness) durumuna göre değiştirilmesidir. Veri setinde bulunan her bir özelliğe ait yoğunluk grafikleri Şekil 2.6'da gösterilmiştir.



Şekil 2.6: Veri yoğunluk grafiği

Çarpıklık bir dağılımın asimetri dağılımını, yani verilerin dağılımının simetrik olmama derecesini ölçer. Diğer bir ifade ile normal dağılımdan sapma miktarı hakkında bilgi verir. Bu çalışmada, pozitif çarpıklığa sahip özniteliklerin eksik değerleri o sütunun medyanı yani ortanca değeriyle (median), normal dağılıma ait özniteliklerin kayıp değerleri ise o sütunun ortalama (mean) değeriyle değiştirilmiştir. Tablo 2.4'de belirtilen özelliklere ait çarpıklık değerleri verilmiştir.

Öznitelik	**Çarpıklık**
Glikoz	0.5309
Kan Basıncı	0.1341
Cilt Kalınlığı	0.6906
İnsülin	2.166
Vücut kitle indeksi	0.5939

Tablo 2.4: Belirtilen özelliklerin çarpıklık değeri

Glikoz, Kan Basıncı, Cilt Kalınlığı ve Vücut Kitle İndeksi gibi sütunlar o kadar çarpık değildir. Bu sütunlar için değerler ortalama ile İnsülin sütunu çarpıklığın etkisinden dolayı medyan değeri olacak şekilde değiştirilmiştir. Veri seti üzerinde yapılacak olan ikinci işlem veri (özellik) ölçeklemedir (Feature Scaling). Özellik Ölçeklendirme, bir veri kümesinde bulunan özelliklerin aralığını normalleştirme işlemidir. Makine öğrenmesinde kullanılan, özellikle uzaklık temelli algoritmaların performansını etkilemekle birlikte Gradyan Mesafesi (Gradient Distance) kullanılan algoritmaların da hız performansını etkilemektedir. Uzaklık temelli algoritmalar, benzerlikleri belirtmek veya bulmak için noktalar arasında bulunan mesafeyi kullanır. Daha büyük büyüklüğe sahip özellikler oluşturulan model tarafından daha yüksek ağırlıklı olarak belirecektir veya derecelendirilecektir ve oluşturulan model bir özelliğe aşırı derecede bağımlı olacaktır. Oluşturulan modelde tüm özelliklerin tahmin sonucuna eşit olarak katkı sağlaması için özellik ölçeklendirme uygulamamız önemlidir. Bu çalışma da standart ölçülendirme (Standart Scaler) yöntemi kullanılmıştır. Standart ölçülendirme, bir veri kümesinde bulunan özellikleri değerleri benzer ölçeği paylaşacak şekilde özellikleri dönüştürme işlemidir. Veri setinde bulunan değişkenleri ortalaması '0' standart sapması '1' olan bir dağılıma çevirir. Herhangi bir sütunda bulunan x özelliğine sahip sütunun standart ölçeklendirme yöntemi Denklem 1' de gösterilmiştir.

$$z = \frac{x-\mu}{\sigma} \qquad 1$$

$$\sigma = standart\ sapma$$

$$\mu = ortalama$$

2.3 Veri Seti Yeniden Örnekleme

Yeniden örnekleme, sınıf dağılımının eşit olmadığı durumlarda yapılan eğitim veri kümesine örnek eklemek veya çıkarmak için tasarlanmış, veri seti dengeleme işlemidir. Veri Seti dengeleme kavramı yani Dengesiz Veri Seti en basit haliyle bir grupta bulunan gözlem sayısının diğer gruba kıyasla daha az olması olarak tanımlanmaktadır.

Makine öğrenmesinde kullanılan algoritmaların performans ölçümünü belirleyen etkenlerden biri de veri içerisinde eşit veya eşite yakın sayı sınıf etiket örneğinin bulunmasıdır. Veri Seti Dengelemek için farklı yaklaşım ve teknikler kullanılmaktadır. Bu çalışmada sınıf dengesizlik probleminin sınıflandırma üzerindeki olumsuz etkisini azaltmak için Sentetik Azınlık AşırıÖrnekleme (SMOTE) tekniği kullanılmıştır. Kullanılan algoritmanın asıl amacı azınlık sınıfında bulunan verilerin çoğunluk sınıfında bulunan veri miktarına yaklaştırılarak çoğaltılmasıdır. Yani örnekler oluşturmak için enterpolasyon tekniği kullanan sentetik aşırı örnekleme (over sampling) tekniğidir.

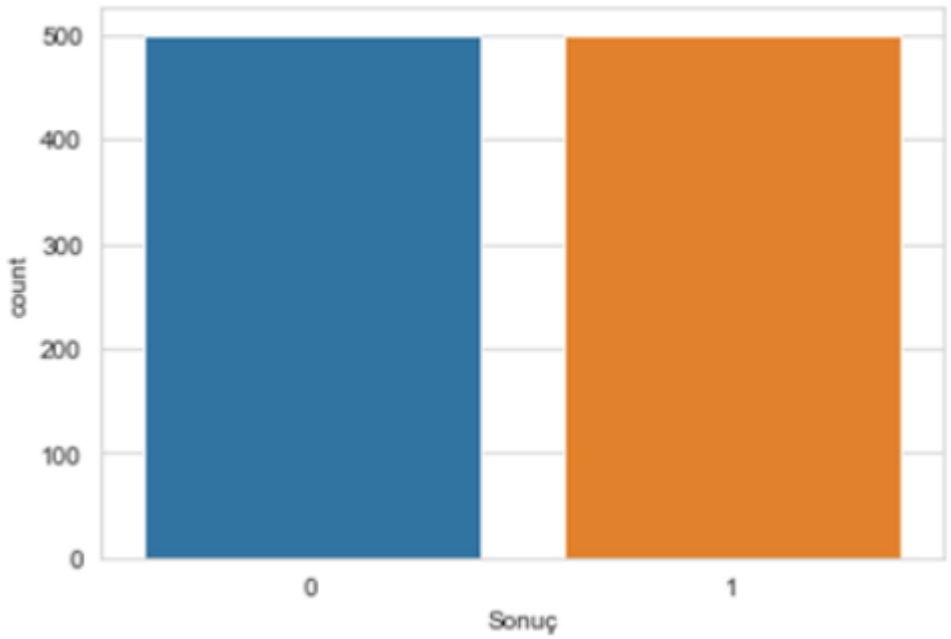
SMOTE, en yakın komşu algoritması (k-NN) fikrine dayanır ve sentetik bir veri örneğinin orijinal ve en yakın komşulardan biri arasında enterpolasyon yapılabileceğini varsayar. SMOTE algoritması, azınlık sınıfından her veri örneğinin komşu ortamını hesaplar, komşularından birini rastgele seçer ve her örnek ile seçilen en yakın komşu arasındaki verilerin interpolasyonu yoluyla sentetik veri yapar. Yapılacak sentetik veri örneklerinin sayısı orijinal veri kümesinin boyutundan küçük olduğunda, algoritma rastgele seçilir ve sentetik veri örnekleri oluşturmak için orijinal bir veri örneği kullanılır. Tersine, yapılacak sentetik veri örneklerinin sayısı orijinal veri kümesinin boyutundan büyük olduğunda, algoritma önceden belirlenmiş aşırı örnekleme oranını kullanarak yinelemeli olarak sentetik veri örnekleri oluşturur.

Sentetik Örneklerin oluşturulması Denklem 2' de gösterildiği gibi kısaca incelenen özellik vektörü Ei ile en yakın komşusu arasındaki alınır, daha sonra bu fark 0 ile 1 arasında rastgele bir sayı δ ile çarpılır. Sonuç incelenen özellik vektörüne eklenir ve yeni örnek oluşturulmuş olur.

$$E_{yeni} = E_i + \left(E_i - E_j\right) \delta \qquad 2$$

Çalışmama kapsamında kullanılan veri setinde bulunan sınıf etiketlerine baktığımızda veri setinde bulunan 768 kayıttan 268'i diyabet hastası, geriye kalan 500'ü diyabet hastası olmayan kişilere aittir. Verilerin %70 i eğitim, %30 kısmı test olarak ayrılmıştır. Kullanılacak sınıflandırma algoritmalarında 400 diyabet hastası olmayan sınıftan, 214 diyabet hastası olan grup olarak ayrılmıştır. SMOTE tekniği kullanılarak

veri seti 500 pozitif, 500 negatif olarak düzenlenmiştir. Şekil 2.7 yeniden örnekleme işlemi yapıldıktan sonra elde edilen veri sayılarını göstermektedir.



Şekil 2.7: Veri seti sayısı

	Gebelik	Glikoz	Tansiyon	Deri Kalınlığı	İnsülin	Vücut Kitle İndeksi	Diyabet geçmişi	Yaş	Sonuç
Ortalama	3.84	120.89	69.10	20.53	79.79	31.99	0.47	33.24	0.34
Standart Sapma	3.36	31.97	19.35	15.95	115.24	7.88	0.33	11.76	0.47
Minimum Değer	0.00	0.00	0.00	0.00	0.00	0.00	0.078	21.00	0.00
25%	1.00	99.00	62.00	0.00	0.00	27.30	0.24	24.00	0.00
50%	3.00	117.00	72.00	23.00	30.50	32.00	0.37	29.00	0.00
75%	6.0	140.25	80.00	32.00	127.25	36.60	0.62	41.00	1.00
Maximum Değer	17.00	199.00	122.00	99.00	846.00	67.10	2.42	81.00	1.00

Tablo 2.5: Veri kümesi değerleri

Bölüm 3

Sınıflandırma

Bu çalışma kapsamında kullanılan sınıflandırma algoritmaları hakkında bilgi verilmiştir.

3.1 Logistic Regression Classifier (Lojistik Regresyon)

Lojistik regresyon, Denetimli Öğrenme tekniği kapsamında yer alan en popüler Makine Öğrenimi algoritmalarından biridir. Belirli bir bağımsız değişken kümesini kullanarak kategorik bağımlı değişkeni tahmin etmek için kullanılır. Lojistik regresyon, kategorik bir bağımlı değişkenin çıktısını tahmin eder. Bu nedenle, sonuç kategorik veya ayrık bir değer olmalıdır. Evet veya Hayır, 0 veya 1, doğru veya Yanlış vb. olabilir, ancak 0 ve 1 olarak kesin değeri vermek yerine, 0 ile 1 arasında kalan olasılıksal değerleri verir. Bu çalışmada kullanılan veri setinde kategorik bağımlı değişken Outcome kolonudur ve 0-1 değerlerine sahiptir. Sınıflandırma sklearn.linear_model kütüphanesinden LogisticRegression kullanılarak yapılmıştır. Random state rastgeleliği kontrol etmek için kullanılan bir model hiperparametresidir. Farklı yürütmelerde aynı sonuç alınabilsin diye random state 0 verilmiştir. Lojistik regresyon algoritması, veri setinin belirtilen özellikleriyle test edilmiştir.

```
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
logr = LogisticRegression(random_state=0)
logr.fit(X_train, y_train)
```

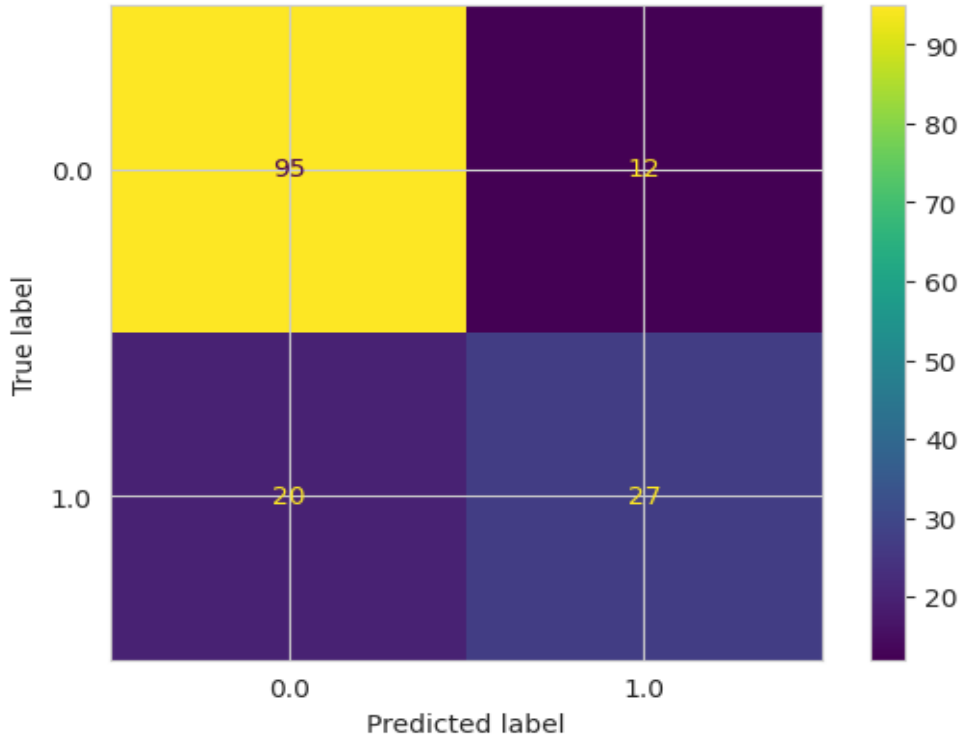
```
LogisticRegression
LogisticRegression(random_state=0)
```

Şekil 3.1: Logistic regression algoritması tanımı

Şekil 3.3'de gösterilen Lojistik regresyon modelinin karmaşıklık matrisi verilmiştir. Oluşturulan model 107 tane hastalığı olmayan kişinin 95 tanesini doğru bilmekte iken hasta olan 47 hastanın da 27 tanesinin hasta olduğunu doğru bir şekilde tahmin edebilmiştir. Modelin accuracy değeri sınıflandırılan örneklerin yüzdesi 0.79, pozitif tahmin edilen değerlerin gerçekte kaç adedinin pozitif olduğunu gösteren precision: 0.69 ve fl skoru: 0.62'dir. Sonuçlar Şekil 3.4'de gösterilmiştir.

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cmLGR = confusion_matrix(y_test, y_predLgr)
disp = ConfusionMatrixDisplay(confusion_matrix= cmLGR, display_labels=logr.classes_)
disp.plot()
plt.show()
```

Şekil 3.2: Logistic regression algoritması karışıklık matrisi tanımı



Şekil 3.3: Logistic regression algoritması karışıklık matrisi

```
print("AccuracyLR:", accuracyLR)
print("precisionLR:", precisionLR)
print("F1LR:", F1LR)
print("Accuracy (mean) %",scores1.mean() * 100)
```

```
AccuracyLR: 0.7922077922077922
precisionLR: 0.6923076923076923
F1LR: 0.627906976744186
Accuracy (mean) % 100.0
```

Şekil 3.4: Logistic regression algoritması doğruluk oranı

3.2 K-Nearest Neighbors Classifier, KNN (K- En Yakın Komşu)

K-En Yakın Komşu, Denetimli Öğrenme tekniğine dayalı en basit Makine Öğrenimi algoritmalarından biridir. K-NN algoritması, yeni durum/veriler ile mevcut durumlar arasındaki benzerliği varsayar ve yeni durumu mevcut kategorilere en çok benzeyen kategoriye koyar. K-NN algoritması mevcut tüm verileri depolar ve benzerliğe göre yeni bir veri noktasını sınıflandırır. Bu, yeni veriler ortaya çıktığında, K-NN algoritması kullanılarak kolayca uygun bir kategoride sınıflandırılabileceği anlamına gelir. Tembel öğrenen algoritması olarak da adlandırılır çünkü eğitim setinden hemen öğrenmez, bunun yerine veri setini depolar ve sınıflandırma anında veri seti üzerinde bir işlem gerçekleştirir. Diyelim ki iki kategori var, yani Kategori A ve Kategori B ve yeni bir x1 veri noktamız var, yani bu veri noktası bu kategorilerden hangisinin içinde olacak. Bu tür bir problemi çözmek için bir K-NN algoritmasına ihtiyacımız var. K-NN'nin yardımıyla, belirli bir veri kümesinin kategorisini veya sınıfını kolayca belirleyebiliriz. Bu çalışmada kullanılan veri setindeki veriler ile hastanın son 5 yıl içinde diyabet geçiren ve geçirmeyen şeklinde kategorize etmek için KNN sınıflandırması kullanılmıştır. metric='minkowski': Bu varsayılan parametredir ve noktalar arasındaki mesafeyi belirler. n_neighbors: Algoritmanın gerekli komşularını tanımlamak için. Bu çalışmada komşu sayısı 5 verilmiştir. K-NN algoritması, veri setinin belirtilen özellikleriyle test edilmiştir.

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5, metric='minkowski')
knn.fit(X_train, y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier()
```

Şekil 3.5: K-Nearest neighbors algoritması tanımı

Şekil 3.8'de gösterilen KNN modelinin karmaşıklık matrisi verilmiştir. Görülmektedir ki oluşturulan model 107 tane hastalığı olmayan kişinin 92 tanesini doğru bilmekte iken hasta olan 48 hastanın da 28 tanesinin hasta olduğunu doğru bir şekilde tahmin edebilmiştir. Modelin accuracy değeri yani doğru olarak sınıflandırılan örneklerin yüzdesi 0.77, pozitif tahmin edilen değerlerin gerçekte kaç adedinin pozitif olduğunu gösteren değer yani precision: 0.65 ve fl skoru: 0.62'dir. Modelin lojistik regreyona göre fl skoru değişmemiştir ancak accuracy ve precision metriklerinde azalma bulunmaktadır. Bu veri seti için KNN modeli lojistik regresyondan daha az doğruluk ve kesinliğe sahiptir. Sonuçlar Şekil 3.7'de gösterilmiştir.

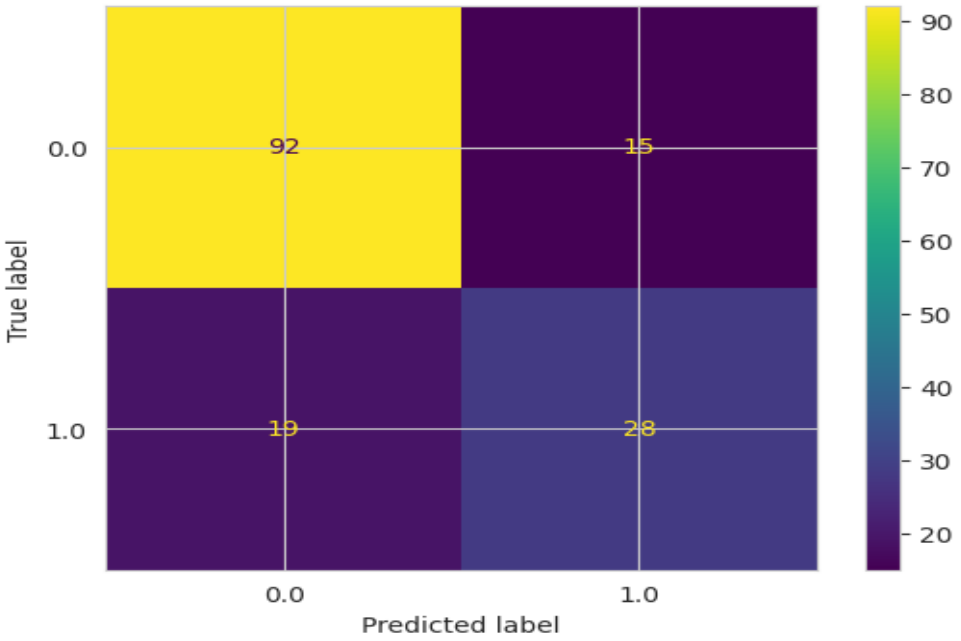
```
cmKNN = confusion_matrix(y_test, ypredKNN)
disp = ConfusionMatrixDisplay(confusion_matrix= cmKNN, display_labels=logr.classes_)
disp.plot()
plt.show()
```

Şekil 3.6: K-Nearest neighbors algoritması karışıklık matrisi tanımı

```
print("AccuracyKNN:", accuracyKNN)
print("precisionKNN:", precisionKNN)
print("F1KNN:", F1KNN)
print("Accuracy (mean) %",scores2.mean() * 100)
```

```
AccuracyKNN: 0.7792207792207793
precisionKNN: 0.6511627906976745
F1KNN: 0.6222222222222222
Accuracy (mean) % 100.0
```

Şekil 3.7: K-Nearest neighbors algoritması doğruluk oranı



Şekil 3.8: K-Nearest neighbors algoritması karışıklık matrisi

3.3 Support Vector Machines Classifier, SVM (Destek Vektör Makineleri)

Destek Vektör Makinesi veya SVM, Sınıflandırma ve Regresyon problemleri için kullanılan en popüler Denetimli Öğrenme algoritmalarından biridir. amacı, gelecekte yeni veri noktasını doğru kategoriye kolayca yerleştirebilmemiz için n-boyutlu uzayı sınıflara ayırabilecek en iyi çizgiyi veya karar sınırını oluşturmaktır. Bu en iyi karar sınırına hiperdüzlem denir. Köpeklerin bazı özelliklerini de taşıyan garip bir kedi gördüğümüzü varsayalım, bu yüzden onun kedi mi yoksa köpek mi olduğunu doğru bir şekilde tanımlayabilen bir model istiyorsak, böyle bir model SVM algoritması kullanılarak oluşturulabilir. Kedi ve köpeklerin farklı özelliklerini öğrenmesi için önce bol kedi köpek görseli olan modelimizi eğiteceğiz ve ardından bu garip canlı ile test edeceğiz. Yani destek vektörü bu iki veri (kedi ve köpek) arasında bir karar sınırı oluşturduğundan ve uç durumları (destek vektörleri) seçtiğinden, kedi ve köpeğin uç durumunu görecektir. Bu çalışmada Lineer SVM kullanılmıştır. Çünkü veri setinin çıktısı lineerdir. = ve 1 olarak düz bir çizgi ile ayrılabilir. Lineer SVM, lineer olarak ayrılabilir veriler için kullanılır, yani bir veri seti tek bir düz çizgi kullanılarak iki sınıfa sınıflandırılabiliyorsa, bu tür veriler lineer olarak ayrılabilir veriler olarak adlandırılır

ve Lineer SVM sınıflandırıcı olarak adlandırılan sınıflandırıcı kullanılır. Destek Vektör Makinesi algoritması, veri setinin belirtilen özellikleriyle test edilmiştir.

```
#3. SVC(SVM classifier)
from sklearn.svm import SVC
svc = SVC(kernel='linear')
svc.fit(X_train, y_train)
```

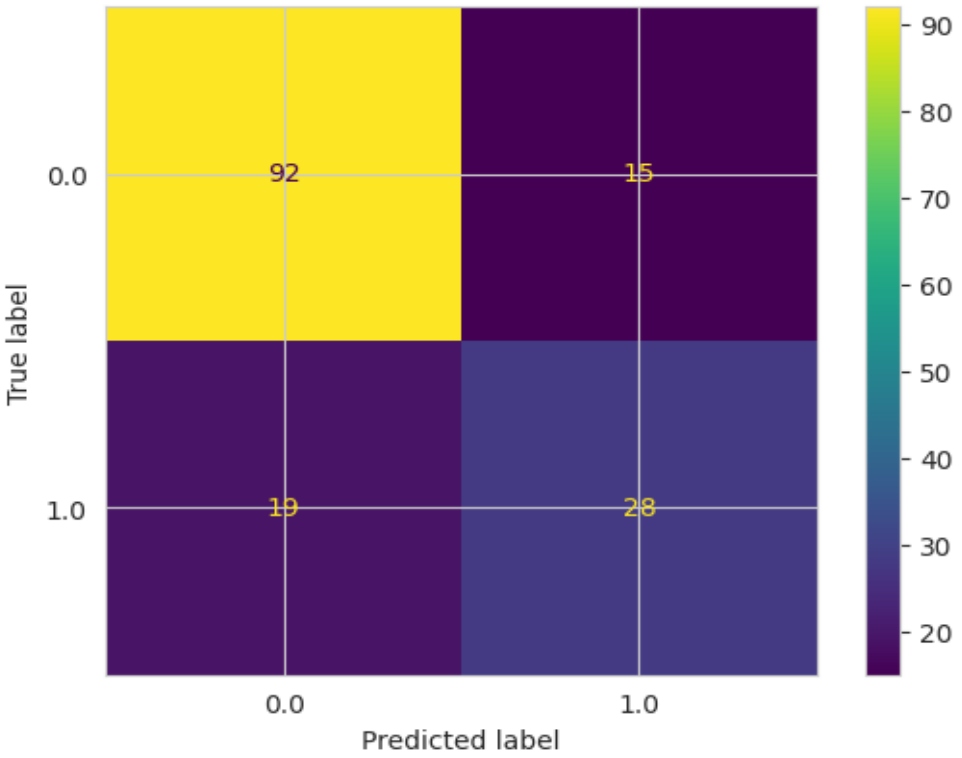
```
SVC(kernel='linear')
```

Şekil 3.9: K-Nearest neighbors algoritması tanımı

Şekil 3.11'de gösterilen SVM modelinin karmaşıklık matrisi verilmiştir. Görülmektedir ki oluşturulan model 107 tane hastalığı olmayan kişinin 92 tanesini doğru bilmekte iken hasta olan 48 hastanın da 28 tanesinin hasta olduğunu doğru bir şekilde tahmin edebilmiştir. Karmaşıklık matrisi KNN model ile aynıdır. Modelin accuracy değeri yani doğru olarak sınıflandırılan örneklerin yüzdesi 0.77, pozitif tahmin edilen değerlerin gerçekte kaç adedinin pozitif olduğunu gösteren değer yani precision: 0.65 ve f1 skoru: 0.62'dir. Modelin metrikleri yine KNN ile aynıdır. Yaklaşık olarak eşit performans göstermişlerdir. Sonuçlar Şekil 3.12'da gösterilmiştir.

```
cmSVC = confusion_matrix(y_test, y_predSVC)
disp = ConfusionMatrixDisplay(confusion_matrix= cmSVC, display_labels=logr.classes_)
disp.plot()
plt.show()
```

Şekil 3.10: Destek vektör makinası algoritması karışıklık matrisi tanımı



Şekil 3.11: Destek vektör makinası algoritması karışıklık matrisi

```
print("AccuracySVC:", accuracySVC)
print("precisionSVC:", precisionSVC)
print("F1SVC:", F1SVC)
print("Accuracy (mean) %",scores3.mean() * 100)
```

```
AccuracySVC: 0.7792207792207793
precisionSVC: 0.6511627906976745
F1SVC: 0.6222222222222222
Accuracy (mean) % 100.0
```

Şekil 3.12: Destek vektör makinası algoritması doğruluk oranı

3.4 Decision Tree Classifier (Karar Ağacı)

İç düğümlerin bir veri kümesinin özelliklerini temsil ettiği, dalların karar kurallarını temsil ettiği ve her yaprak düğümün sonucu temsil ettiği ağaç yapılı bir sınıflandırıcıdır. ir Karar ağacında, Karar Düğümü ve Yaprak Düğümü olmak üzere iki düğüm vardır. Karar düğümleri herhangi bir karar vermek için kullanılır ve birden

fazla dala sahiptir, Yaprak düğümleri ise bu kararların çıktısıdır ve başka dallar içermez. Kararlar veya test, verilen veri setinin özelliklerine göre gerçekleştirilir. Bir karar ağacı basitçe bir soru sorar ve cevaba göre (Evet/Hayır), ağacı alt ağaçlara ayırır. Karar Ağaçları, karar verirken genellikle insanın düşünme yeteneğini taklit eder, bu nedenle anlaşılması kolaydır. Entropi: Entropi, belirli bir öznitelikteki safsızlığı ölçmek için kullanılan bir ölçümdür. Verilerdeki rastgeleliği belirtir. Bu çalışmada bilgi kazanımı niteliği seçilmiştir. Bilgi kazanımı, bir özniteliğe dayalı olarak bir veri kümesinin bölümlenmesinden sonra entropideki değişikliklerin ölçümüdür. Bir özelliğin bize bir sınıf hakkında ne kadar bilgi sağladığını hesaplar. Bilgi kazancının değerine göre düğüm bölünüp karar ağacı oluşturulur. Karar Ağaçları algoritması, veri setinin belirtilen özellikleriyle test edilmiştir.

```
#4. Desicion Tree
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(criterion='entropy')

dtc.fit(X_train, y_train)
```

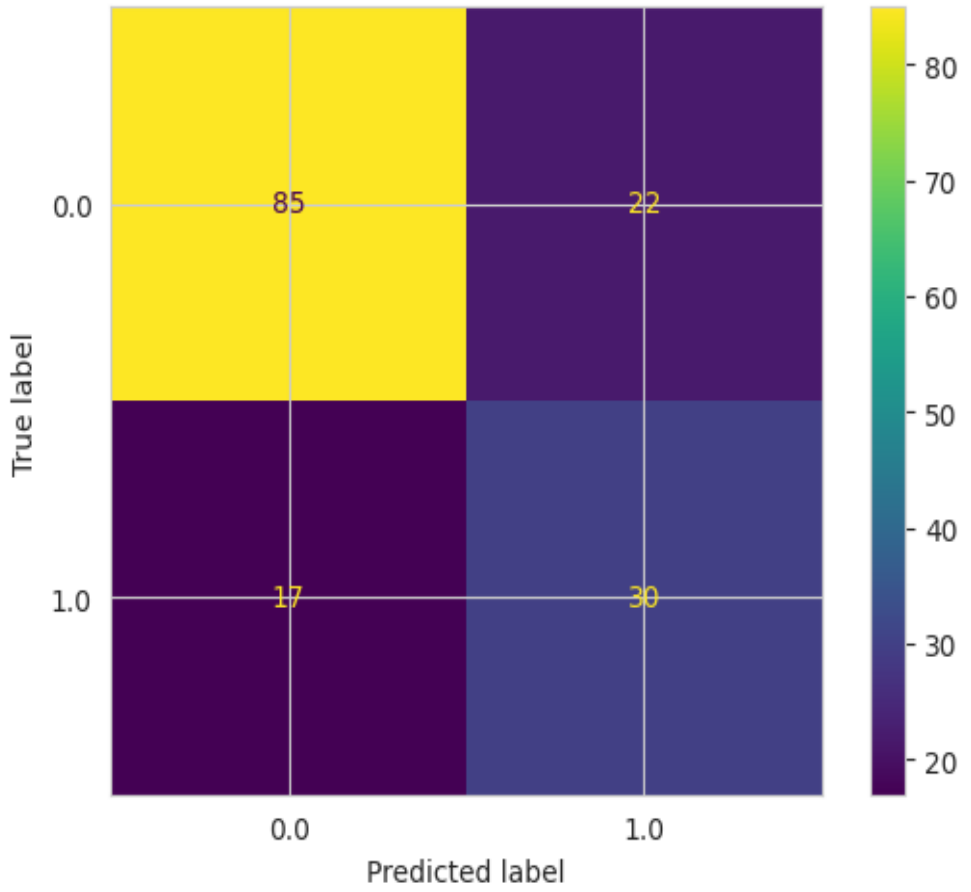
```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy')
```

Şekil 3.13: Karar ağaçları algoritması tanımı

Şekil 3.15'de gösterilen Desicion Tree modelinin karmaşıklık matrisi verilmiştir. Görülmektedir ki oluşturulan model 107 tane hastalığı olmayan kişinin 85 tanesini doğru bilmekte iken hasta olan 48 hastanın da 30tanesinin hasta olduğunu doğru bir şekilde tahmin edebilmiştir. Burada diğer modellere göre True Negative sayısında azalma görülmektedir. (Hasta olmayana hasta değil demek.) Ancak True pozitif sayısı da 30'a yükselmiştir. (Hastaya hasta demek.) Modelin accuracy değeri yani doğru olarak sınıflandırılan örneklerin yüzdesi 0.74, pozitif tahmin edilen değerlerin gerçekte kaç adedinin pozitif olduğunu gösteren değer yani precision: 0.57 ve f1 skoru: 0.60'dir. Karmaşıklık matrisinde de görülen True negatif sayısındaki azalma accuracy ve precision değerlerinde de düşüş olarak gözlemlenmektedir. Modelin performansı diğer modellere göre oldukça düşüktür. Sonuçlar Şekil 3.16'de gösterilmiştir.

```
cmDTC = confusion_matrix(y_test, y_predDtc)
disp = ConfusionMatrixDisplay(confusion_matrix= cmDTC, display_labels=logr.classes_)
disp.plot()
plt.show()
```

Şekil 3.14: Karar ağaçları algoritması karışıklık matrisi tanımı



Şekil 3.15: Karar ağaçları algoritması karışıklık matrisi

```
print("AccuracyDTC:", accuracyDTC)
print("precisionDTC:", precisionDTC)
print("F1DTC:", F1DTC)
print("Accuracy (mean) %",scores4.mean() * 100)
```

```
AccuracyDTC: 0.7597402597402597
precisionDTC: 0.5961538461538461
F1DTC: 0.6262626262626262
Accuracy (mean) % 100.0
```

Şekil 3.16: Karar ağaçları algoritması doğruluk oranı

3.5 Random Forest Classifier (Rastgele Orman)

Random Forest, denetimli öğrenme tekniğine ait popüler bir makine öğrenme algoritmasıdır. ML'de hem Sınıflandırma hem de Regresyon problemleri için kullanılabilir. Karmaşık bir sorunu çözmek ve modelin performansını iyileştirmek için birden çok sınıflandırıcıyı birleştirme süreci olan topluluk öğrenme kavramına dayanır. Adından da anlaşılacağı gibi, "Random Forest, belirli bir veri kümesinin çeşitli alt kümelerinde bir dizi karar ağacı içeren ve bu veri kümesinin tahmin doğruluğunu iyileştirmek için ortalamayı alan bir sınıflandırıcıdır." Rastgele orman, tek bir karar ağacına güvenmek yerine, her ağaçtan tahmin alır ve tahminlerin çoğunluk oylarını temel alır ve nihai çıktıyı tahmin eder. kriter= Bölmenin doğruluğunu analiz eden bir fonksiyondur. Bu çalışmada karar ağacında olduğu gibi bilgi kazancı için "entropi" kullanıldı. n_estimators= Rastgele Ormandaki gerekli ağaç sayısı. Varsayılan değer 10'dur. Herhangi bir sayı seçilebilir ancak ezberleme sorununa dikkat edilmesi gerekir. Bu çalışmada varsayılan değer kullanılmıştır Rastgele orman algoritması, veri setinin belirtilen özellikleriyle test edilmiştir.

```
#5.Random Forest
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(n_estimators=10, criterion='entropy')
rfc.fit(X_train, y_train)
```

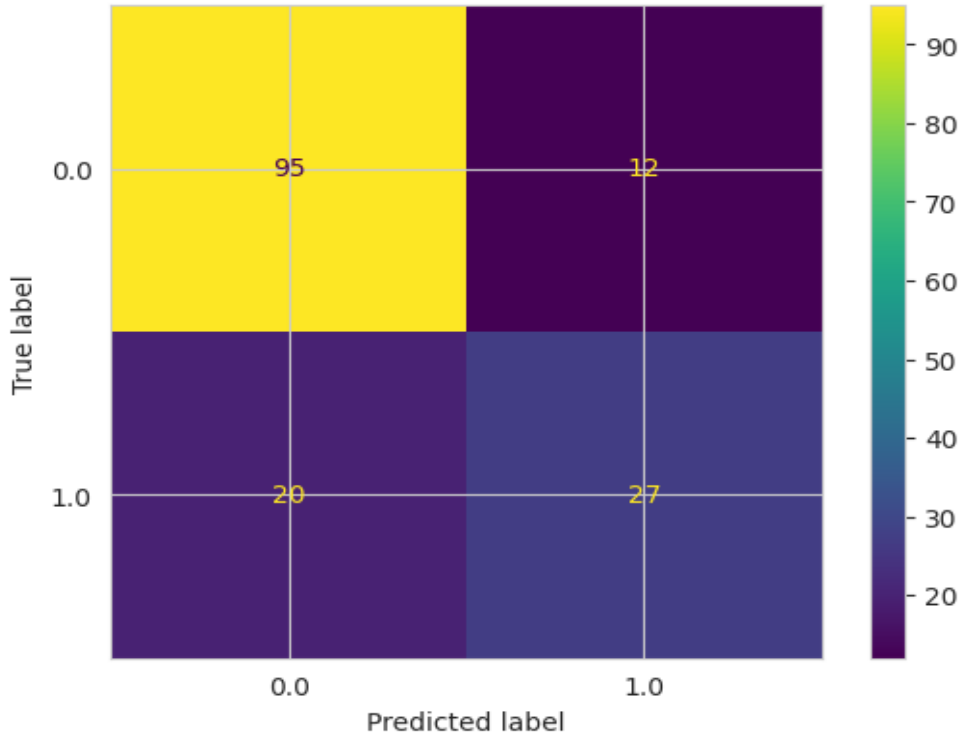
```
RandomForestClassifier
RandomForestClassifier(criterion='entropy', n_estimators=10)
```

Şekil 3.17: Rastgele orman algoritması tanımı

Şekil 3.19'de gösterilen Random Forest modelinin karmaşıklık matrisi verilmiştir. Görülmektedir ki oluşturulan model 107 tane hastalığı olmayan kişinin 95 tanesini doğru bilmekte iken hasta olan 47 hastanın da 27 tanesinin hasta olduğunu doğru bir şekilde tahmin edebilmiştir. Modelin accuracy değeri yani doğru olarak sınıflandırılan örneklerin yüzdesi 0.79, pozitif tahmin edilen değerlerin gerçekte kaç adedinin pozitif olduğunu gösteren değer yani precision: 0.69 ve f1 skoru: 0.62'dir. Sonuçlar Şekil 3.20'de gösterilmiştir.

```
cmRFC = confusion_matrix(y_test, y_predRFC)
disp = ConfusionMatrixDisplay(confusion_matrix= cmRFC, display_labels=logr.classes_)
disp.plot()
plt.show()
```

Şekil 3.18: Rastgele orman algoritması karışıklık matrisi tanımı



Şekil 3.19: Rastgele orman algoritması karışıklık matrisi

```
print("AccuracyRFC:", accuracyRFC)
print("precisionRFC:", precisionRFC)
print("F1RFC:", F1RFC)
print("Accuracy (mean) %",scores5.mean() * 100)
```

```
AccuracyRFC: 0.8246753246753247
precisionRFC: 0.717391304347826
F1RFC: 0.7096774193548387
Accuracy (mean) % 100.0
```

Şekil 3.20: Rastgele orman algoritması doğruluk oranı

Bölüm 4

Sonuç

Çalışma kapsamında yapılan analizlerde Python kodlama dili kullanılmıştır. Diyabet hastalığının teşhis edilmesi için Lojistik regresyon ve DVM makine öğrenmesi sınıflandırma algoritmaları kullanılmıştır. Makine öğrenmesinin en önemli parçalarından biri de sınıflandırma işlemidir. Sınıflandırma işleminde elimizde bulunan veriler eğitim seti ve test seti olmak üzere iki ana aşamaya ayrılır. Oluşturulan model eğitim seti kullanılarak, model performansı test seti kullanılarak yapılır. Buradan anlaşılacağı gibi veri seti makine öğrenmesi algoritmalarında çok önemlidir. Oluşturulan modelin değerlendirme kriterleri Tablo 4.1'de açıklamaları ve formülleri verilmiş olan Doğruluk Oranı (Accuracy Rate), Kesinlik (Precision), Duyarlılık (Recall) ve F1-Skore (F1 Score) değerleri kullanılarak hesaplanmıştır. Performans ölçüm değerlerin hesaplanması için Tablo 4.2'de gösterilen karmaşıklık (hata) matrisi kullanarak yapılmıştır.

Değerlendirme Kriter	Açıklama	Formül
Doğruluk Oranı	Oluştular modelin hedef sınıfları tahmin başarısı.	(DP+DN)/ N
Kesinlik (Precision)	Oluşturulan model de sonucun ne kadar doğru olduğunu gösterir.	DP/(DP+YP)
Duyarlılık (Recall)	Oluşturulan model de doğru örnekleri bulma yeteneğini gösterir.	DN/(YP+DN)
F1- Skore (F1 Score)	Kesinlik ve duyarlılığın harmonik ortalamasıdır.	2*(Kesinlik*Duyarlılık)/ (Kesinlik+Duyarlılık)/

Tablo 4.1: Model değerlendirme kriterleri

	Pozitif	Negatif
Pozitif	DP (Doğru Pozitif / TP)	YN (Yanlış Negatif/ FN)
Negatif	YP (Yanlış Pozitif / FP)	DN (Doğru Negatif / TN)

Tablo 4.2: Hata matrisi

* DP (TP): Gerçek sınıfın değeri pozitiftir (yani diyabet hastası) ve kullanılan yöntemle pozitif (diyabet hastası) olarak tahmin edilmiştir.

* DN (TN): Gerçek sınıfın değeri negatiftir (diyabet hastası olmayan) ve kullanılan yöntemle negatif (diyabet hastası olmayan) olarak tahmin edilmiştir.

* YN (FN): Gerçek sınıfın değeri pozitiftir (yani diyabet hastası) fakat kullanılan yöntemle negatif (diyabet hastası olmayan) olarak tahmin edilmiştir.

* YP (FP): Gerçek sınıfın değeri negatiftir (yani diyabet hastası olmayan) fakat kullanılan yöntemle pozitif (diyabet hastası) olarak tahmin edilmiştir.

	Doğruluk Oranı	**Kesinlik**	**Duyarlılık**	**F1-Score**
Logistic Regression	%80	%79	%69	%62
K-Nearest Neighbors	%78	%77	%65	%62
Support Vector Machines	%78	%77	%65	%62
Decision Tree	%76	%75	%59	%62
Random Forest	%83	%82	%71	%70

Tablo 4.3: Değerlendirme sonuçları

Modeller arasında metrikler üzerinde değerlendirme yapıldığında en başarılı modeller Random forest ve Logistic regression olmuştur. Veri seti lineer olduğundan Logistic regression başarılı olması beklenen bir durumdur. Bu iki model içinde değerlendirmeye alınan tüm metrikler neredeyse aynıdır. En başarısız model ise Decision Tree olmuştur.

Kaynaklar

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

[2] Diabetes mellitus ve komplikasyonlarının tanı, tedavi ve izlem kılavuzu, https://file.temd.org.tr/Uploads/publications/guides/documents/diabetesmellitus_2022.pdf, Accessed 09 May, 2022.

[3] N. Eroğlu, Diabetes Mellitus'un komplikasyonları.İzmir Demokrasi Üniversitesi Sağlık Bilimleri Dergisi,1(2), 6-12, 2018.

[4] K. Akyol ve A. Karacı, Diyabet hastalığının erken aşamada tahmin edilmesi için makine öğrenme algoritmalarının performanslarının karşılaştırılması. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 9(6),123–134, 2021, https://doi.org/10.1016/10.29130/dubited.1014508.

[5] F. Al-Areqi and M. Z. Konyar, Effectiveness evaluation of different feature extraction methods for classification of Covid-19 from computed tomography NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci. 2023; 12(1), 064-071 M. Korkmaz, K. Kaplan 71 images: A high accuracy classification study.Biomedical Signal Processing and Control, 76, 2022, https://doi.org/10.1016/j.bspc.2022.103662.

[6] F. Al-Areqi and M. Z. Konyar, transfer öğrenme mimarileri kullanılarak bilgisayarlı tomografi görüntülerinden Covid-19'un yüksek doğrulukla sınıflandırılması. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 13(3), 457-466, 2022, https://doi.org/10.24012/dumf.1129870.

[7] Ü. Veranyurt, A. F. Deveci, ve M. F. Esen, Makine öğrenmesi teknikleriyle hastalık sınıflandırması:Random Forest, K-Nearest Neighbour ve Adaboost algoritmaları uygulaması. Uluslararası Sağlık Yönetimi ve Araştırmaları Dergisi, 6(2), 275–286, 2020.

[8] Y. Özkan, B. S. Yürekli, ve A. Suner, Diyabet tanısının tahminlenmesinde denetimli makine öğrenme algoritmalarının performans karşılaştırması.Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 12(1), 211–226, 2021, https://doi.org/10.17714/gumusfenbil.820882.

[9] G. Bilgin, Makine öğrenmesi algoritmaları kullanarak erken dönemde diyabet hastalığı riskinin araştırılması. Zeki Sistemler Teori ve Uygulamaları Dergisi, 4(1),55–64, 2021, https://doi.org/10.38016/jista.877292.

[10] N. P. Tigga and S. Garg, Prediction of type 2 diabetes using machine learning classification methods. Procedia Computer Science., 167, 706–716, 2020, https://doi.org/10.1016/j.procs.2020.03.336.

[11] S. Nahzat ve M. Yağanoğlu, Diabetes prediction using machine learning classification algorithms. European Journal of Science and Technology, 24, 53–59, 2021, https://doi.org/10.31590/ejosat.899716.

[12] V. Chang, J. Bailey, Qianwen, A. Xu, and Z. Sun, Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms, https://doi.org/10.1007/s00521-022-07049-z.

[13] S. Joshi and S. R. P. Shetty, Performance analysis of different classification methods in data mining for diabetes dataset using WEKA tool. International Journal on Recent and Innovation Trends in Computing and Communication, 3(3), 1168–1173, 2015, https://doi.org/10/10.17762/ijritcc2321-8169.150361.

[15] M. B. ER ve İ. Işık, LSTM tabanlı derin ağlar kullanılarak diyabet hastalığı tahmini. Türk Doğa ve Fen Dergisi, 10(1), 68–74, 2021, https://doi.org/10.46810/tdfd.818528.

[16] G. Yangın, XGboost ve Karar Ağacı tabanlı algoritmaların diyabet veri setleri üzerine uygulaması.Yüksek Lisans Tezi, Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü, Türkiye, 2019.

[17] A. G. Karegowda, V. Punya,, M. A. Jayaram, and A. S. Manjunath, Rule based classification for diabetic patients using Cascaded K-Means and Decision Tree C4 . 5. International Journal of Computer Applications, 45(12), 45–50, 2012, https://doi.org/10.5120/6836-9460.

[18] M. Maniruzzaman, N. Kumar, M. M. Abedin, M. S. Islam, H. S. Suri, A.s El-Baz, J. S. Suri, Comparative approaches for classification of Diabetes Mellitus data: Machine learning paradigm. Computer Methods and Programs in Biomedicine. 152, 23–34, 2017,https://doi.org/10.1016/J.CMPB.2017.09.004.

[19] Ö. Deperlioğlu ve U. Köse, Derin Sinir Ağları kullanarak diabet teşhisi., 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 1–4, Ankara, Türkiye, 2018.

[20] P. Cihan and H. Coskun, Performance comparison of machine learning models for diabetes prediction, 29th Signal Processing and Communications Applications Conference (SIU'2021), pp. 26–30, İstanbul, Türkiye, 2021,

[21] P. B. M. Kumar, R. S. Perumal, R. K. Nadesh, and K. Arivuselvan, Type 2: Diabetes Mellitus prediction using Deep Neural Networks classifier. International Journal of Cognitive Computing in Engineering, 1, 55–61, 2020, https://doi.org/10.1016/j.ijcce.2020.10.002.

[22] E. Yıldırım and A. Çalhan, Machine learning supported diabetes prediction with Apache Spark. Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 10(3), 1107–1117, 2022, https://doi.org/10.29130/dubited.999048.

[23] Pima Indians Diabetes Database | Kaggle, https://www.kaggle.com/datasets/uciml/pima-indiansdiabetes-database/, Accessed 09 May, 2022. [25] Python. [Online]. Available: https://www.python.org/ (visited on 2022).

[24] Visualize programming language popularity using tiobeindexpy. [Online].Available: https : / / towardsdatascience . com / visualize -programming - language - popularity - using - tiobeindexpy -f82c5a96400d (visited on 2024).

[25] Numpy. [Online]. Available: https://numpy.org/ (visited on 2024).

[26] Pandas. [Online]. Available: https : / / pandas . pydata . org/ (visited on 2024).

[27] Scikit-learn. [Online]. Available: https://scikit- learn.org/stable/(visited on 2024).

[28] Matplotlib: Visualization with python. [Online]. Available: https : / /matplotlib.org/ (visited on 2024).

[29] Tensorflow. [Online]. Available: https://www.tensorflow.org/ (visited on 2024).

[30] Brain team. [Online]. Available: https : / / research . google / teams /brain/ (visited on 2022).

[31] Keras. [Online]. Available: https://keras.io/ (visited on 2022).