GoalZone Fitness: Üye Katılım Tahminleriyle Sınıf Kapasitesi Optimizasyonu

Selen Erdoğan
Selenerdogan2019@gtu.edu.tr
Elektronik Mühendisliği Bölümü, GTÜ

I. GİRİŞ

GoalZone, Kanada'da bir fitness kulübü zinciridir. GoalZone, iki farklı kapasitede - 25 ve 15 kişilik - çeşitli fitness sınıfları sunmaktadır. Bazı sınıflar her zaman tamamen doludur. Tamamen dolu sınıflar genellikle düşük katılım oranına sahiptir.

GoalZone, sınıflar için sunulan yer sayısını artırmak istemektedir. Bunu, üyelerin sınıfa katılıp katılmayacağını tahmin ederek gerçekleştirmek hedefleniyor. Eğer bir üyenin sınıfa katılmayacağı önceden tahmin edilebilirse, bu yer başka bir üye için kullanılabilir hale getirilebilir.

Bu durum, fitness merkezinin mevcut kaynaklarını daha verimli kullanmasına ve daha fazla üyeye hizmet sunmasına olanak tanır. Aynı zamanda, sınıfların tam dolu olmasına rağmen katılım oranlarının düşük olmasının önüne geçilmesi, hem işletme verimliliğini hem de müşteri memnuniyetini artırabilir. GoalZone'un amacı, katılım tahminlerine dayanarak daha etkili bir yer yönetimi yapmak ve böylece hem üyelerin memnuniyetini hem de işletmenin genel performansını iyileştirmektir..

II. DATA PREPROCESSING

Değerlerin uygunluğu kontrol edildi: Her sütun için, verilerin tanımlanan özelliklere (örneğin, veri türü, aralık, format) uygunluğu kontrol edildi.

Uyuşmayan durumlar belirtildi. Eksik değerler tespit edildi:

Her sütunda eksik değerlerin (NaN, boş değerler vb.) sayısı hesaplandı.

Eksik değerlerin toplam veri seti üzerindeki yüzdesi belirlendi.

Düzeltmeler yapıldı. Değerlerin açıklama ile uyuşmaması durumunda, bu değerler düzeltilmek için gerekli adımlar atıldı. Veri formatlarını düzeltmek için yapılan değişiklikler anlatıldı.

	booking_id	months_as_member	weight	days_before	day_of_week	time	category	attended
0	1	17	79.56	8	Wed	PM	Strength	0
1	2	10	79.01	2	Mon	AM	HIIT	0
2	3	16	74.53	14	Sun	AM	Strength	0
3	4	5	86.12	10	Fri	AM	Cycling	0
4	5	15	69.29	8	Thu	AM	HIIT	0

Şekil 1

Şekil 1'de verilerin düzenlenmeden önceki hali gözükmektedir. Veriye yapılan işlemler sırasıyla belirtilmiştir.

```
In [6]: df.days_before.unique()

Out[6]: array(['8', '2', '14', '10', '6', '4', '9', '12', '5', '3', '7', '13',
               '12 days', '20', '1', '15', '6 days', '11', '13 days', '3 days',
               '16', '1 days', '7 days', '8 days', '10 days', '14 days', '17',
               '5 days', '2 days', '4 days', '29'], dtype=object)
```

Şekil 2

Şekil 2'de görüleceği üzere eldeki verinin tüm column yapılarındaki değerler incelendi.

```
In [12]: df.time.unique()

Out[12]: array(['PM', 'AM'], dtype=object)

In [13]: df.category.unique()

Out[13]: array(['Strength', 'HIIT', 'Cycling', 'Yoga', '-', 'Aqua'], dtype=object)
```

Bir kategori olarak "-" bulunuyor, bunu "bilinmiyor" ile değiştirelim.

```
In [14]: df['category']=df['category'].replace('-', 'unknown')

In [15]: df.category.value_counts()

Out[15]: HIIT        667
         Cycling     376
         Strength    233
         Yoga        135
         Aqua         76
         unknown      13
         Name: category, dtype: int64

In [16]: df.attended.value_counts()

Out[16]: 0    1046
         1     454
         Name: attended, dtype: int64
```

Şekil 3

Kategori isimlerinde bulunan veri anlaşılmazlığı giderildi.
Eksik değerler ortalama değer ile dolduruldu.
Veri hakkında detaylı analiz yapıldıktan sonra elde edilen sonuçlar ve yapılan işlemler aşağıdaki gibidir.

Veri çerçevesi toplam 1500 satır ve 8 sütun içeriyor, çeşitli özellikleri temsil ediyor.
Veri çerçevesi, integer (tam sayı), float (ondalık sayı) ve object (nesne) veri türlerini içeriyor.
• Bazı sütunlar eksik değerleri işlemek ve nesne verilerini analiz için daha uygun türlere dönüştürmek için daha fazla ön İşlemeye ihtiyaç duyuyordu. Ayrıntılar aşağıda verilmiştir:
Veri hakkındaki detaylar
• months_as_member: Açıklamayla eşleşiyor.

• days_before Bu, sayısal bir veri olarak açıklanmasına rağmen içinde 'gün' ifadesi olduğu için Nesne (Object) türünde. Bu sorunu çözmek için 'gün' ifadesini kaldırarak düzeltildi.
day_of_week: Haftanın günü nesne (Object) türünde, açıklamada sayısal bir tür olarak belirtilmiş. Ayrıca, bazı anormallikler bulundu ve düzeltildi, sayısal türe dönüştürüldu.

- time: Açıklamayla eşleşiyor.
- category: Kategori, "-" değerine sahipti, bu "-" değeri "Bilinmiyor" ile değiştirildi.
- attended: Açıklamayla eşleşiyor.
- weight: Bu sütunda 20 eksik değer bulunuyordu, bunlar genel ortalama ile değiştirildi.

• Hiçbir tekrarlanan (duplikat) değer bulunmadı.
Veriye ait açıklayıcı istatistikler:
• Üye olarak geçirilen aylar (months_as_member): Ortalama olarak, üyeler yaklaşık 15.63 ay boyunca üye olmuşlardır. Bununla birlikte, bu rakamda oldukça bir varyasyon bulunmaktadır, bunu 12.93'lük standart sapma göstermektedir. Özellikle, 148 ay gibi maksimum bir değerle potansiyel olarak aykın bir durum bulunmaktadır, bu durum daha fazla incelenmelidir.
• Ağırlık Dağılımı: Üyelerin ortalama ağırlığı yaklaşık 82.61 birimdir. Ağırlık değerlerinin dağılımı orta düzeydedir ve ortalama 12.68 standart sapma ile gösterilmektedir. Ağırlık değerleri tipik bir dağılımı takip etmektedir ve çoğunluğu 73.56 ile 89.38 birim arasında bulunmaktadır.
• Önceden Günler (days_before): Rezervasyonların ortalama yapılma süresi etkinlikten önce yaklaşık 8.35 gündür. Bu değişkenin dağılımı pozitif yönde eğilimli, çünkü ortalama, medyandan (9.00) büyüktür.
• Katılım (attended): Katılım özelliği ikili (binary) bir veri türünü temsil eder ve bir etkinliğin katılıp katılmadığını gösterir (katılmadı için 0, katıldı için 1). Ortalama olarak, etkinliklerin yaklaşık %30'u katılmıştır, bu genel katılım oranı hakkında bilgi sağlar.

III. KEŞİFÇİ VERİ ANALİZİ

Veri Görselleştirmesi: Veriyi grafikler, histogramlar, kutu grafikleri, çizgi grafikleri ve dağılım grafikleri gibi görsel araçlar kullanarak görselleştirmektir Bu görsel araçlar, veri setinin dağılımını, yoğunluğunu ve ilişkilerini görmemize yardımcı olur.

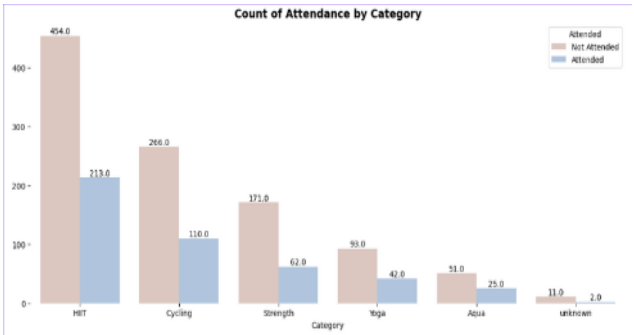
Özet İstatistikler: Verinin temel istatistiksel özelliklerini inceleme işlemine denir. Bu, ortalama, standart sapma, medyan, çeyreklikler gibi istatistiklerin hesaplanmasını içerir. Bu özellikler, verinin merkezi eğilimini, yayılımını ve simetrisini anlamamıza yardımcı olur.

Aykırı Değerlerin Tespiti: Veri setindeki aykırı değerleri (outliers) belirleme işlemidir. Aykırı değerler, genellikle diğer verilerden büyük ölçüde farklı olan veya anormallik gösteren verilerdir ve analizi etkileyebilirler.

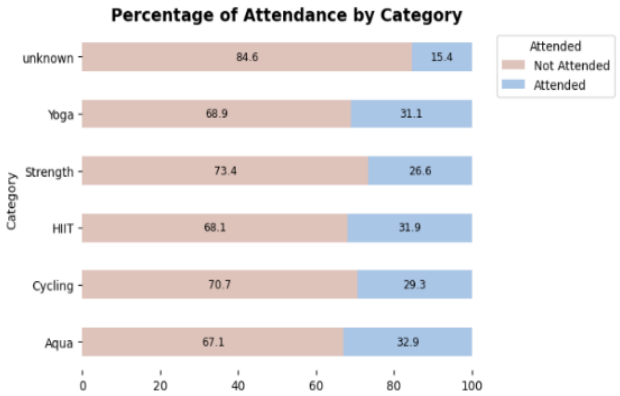
İlişkilerin İncelenmesi: Değişkenler arasındaki ilişkileri anlamaktır. Korelasyon analizi, veri setindeki değişkenler arasındaki ilişkiyi ölçmek için kullanılır. Bu, bir değişkenin diğerine nasıl etki ettiğini veya ilişkili olduğunu görmemize yardımcı olur.

Gruplama ve Kategorizasyon: Veriyi kategorilere veya gruplara ayırma ve bu gruplar arasındaki farkları incelemektir. Bu, veri setinin farklı alt gruplarının özelliklerini anlama ve karşılaştırma amacıyla kullanılır.

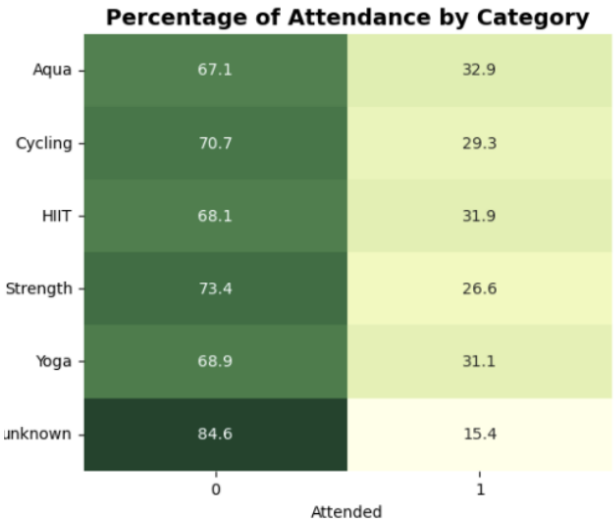
A. Veri Görselleştirilmesi



Şekil 4

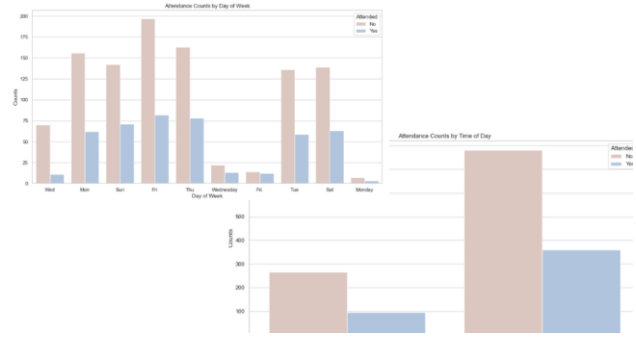


Şekil 5

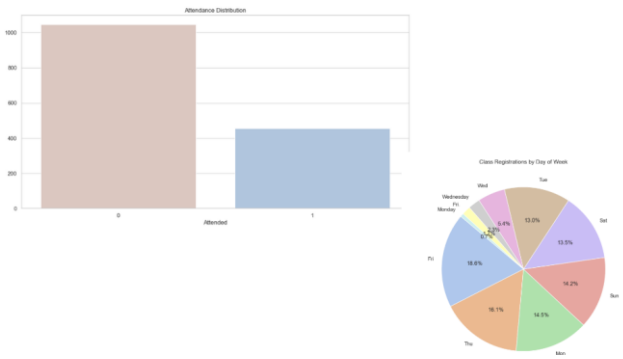


Şekil 6

Grafiklerden elde edilen bilgilere göre:
Farklı fitness sınıfı kategorilerindeki katılım yüzdelerinin değişkenliği, gözlemlerin dengeli olmadığını göstermektedir. Bazı kategoriler daha yüksek katılım oranlarına sahipken, diğerleri daha düşük katılım oranlarına sahiptir, bu da katılım dağılımında bir dengesizlik olduğunu göstermektedir.



Şekil 7

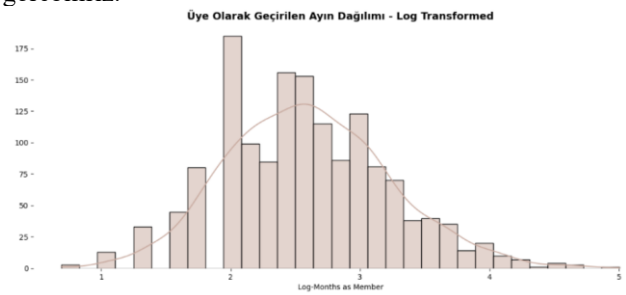


Şekil 8

Şekil 7 ve 8' de veri analizi yapılmak için görselleştirme kullanıldı.

Şekil 9

Şekil 9'de görüleceği üzere dağılım sağa çarpık (çarpık) bir şekilde eğilmiştir, bu durum aykırı değerlerin varlığını göstermektedir.
Veri küçük olduğundan, bir log dönüşümü uygulandıktan sonra dağılımın normal bir dağılıma yakın olduğunu görebiliriz.

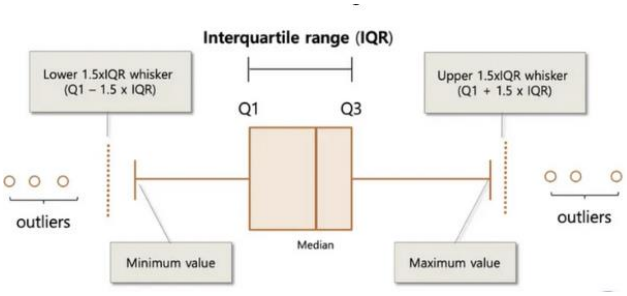


Şekil 10

Log dönüşümü veri setindeki çarpıklığı azaltmak için yapılmıştır. Büyük değerleri küçülterek, küçük değerleri daha az etkileyerek veri setindeki ölçek farklılıklarını azaltır.

B. Aykırı Değerlerin Tespiti

Aykırı değer, bir veri setindeki gözlemlerin geri kalanından büyük ölçüde farklı olan herhangi bir veri noktasıdır. Diğer bir tanımla genel eğilimin oldukça dışına çıkan gözlemdir.



Şekil 11

Q1: %25lik çeyrek
Median: Q2
Q3: %75lik çeyrek

IQR = Q3 — Q1 ‘dir. Bu aralık, kabul edilebilirlik sınırlarını yani Üst ve Alt sınır belirlemede önemli bir faktördür.

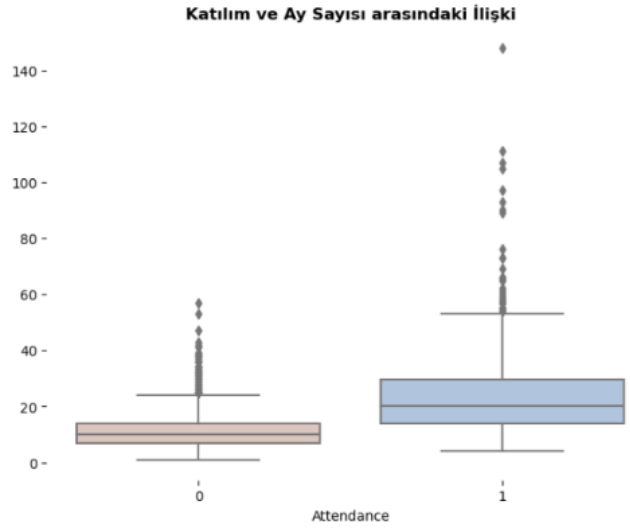
Üst Sınır için; Q3 değerine, IQR değerinin 1.5 katını ekliyoruz:

(Q3 + 1.5 x IQR)

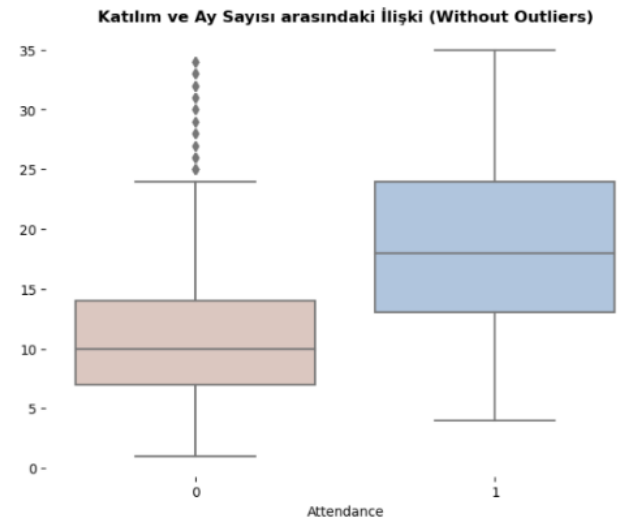
Alt Sınır için; Q3 değerinden, IQR değerinin 1.5 katını çıkarıyoruz:

(Q1 — 1.5 x IQR)

Belirlediğimiz bu alt ve üst limitler dışında yer alan değerler, Aykırı Değer olarak alınacaktır.



Şekil 12



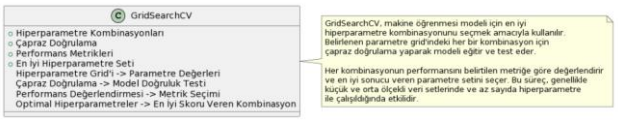
Şekil 13

Şekil 12 ve Şekil 13’te görüldüğü üzere IQR hesaplaması ve değer çıkarılmasından sonra veride aykırı değer azalması gözlemlenmiştir.
Aykırı değerlerin kaldırılmasından sonra şunu çıkarabiliriz:
- Katılanlar, katılmayanlara göre daha yüksek Q1, medyan ve Q3'e sahiptir.
- Veri sağa çarpık (çarpık) olarak işaret ediyor.

C. GridSearchCV

GridSearchCV (Grid Search Cross-Validation), makine öğrenimi modellerinin hiperparametrelerini ayarlamak için yaygın olarak kullanılan bir tekniktir. Hiperparametreler, bir makine öğrenimi modelinin performansını etkileyen, ancak veri tarafından öğrenilmeyen parametrelerdir. Örnek olarak, bir destek vektör makinesi (SVM) modelinin C değeri veya bir karar ağacı modelinin maksimum derinliği gibi hiperparametreler verilebilir.

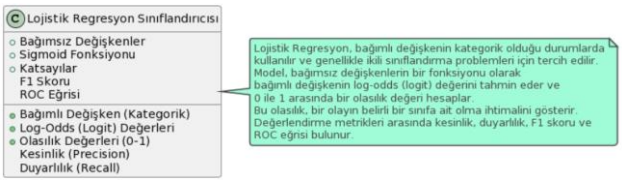
GridSearchCV, belirli bir model için hiperparametre kombinasyonlarını belirlemek ve en iyi hiperparametreleri seçmek için kullanılır.



Şekil 14 GridSearchCV

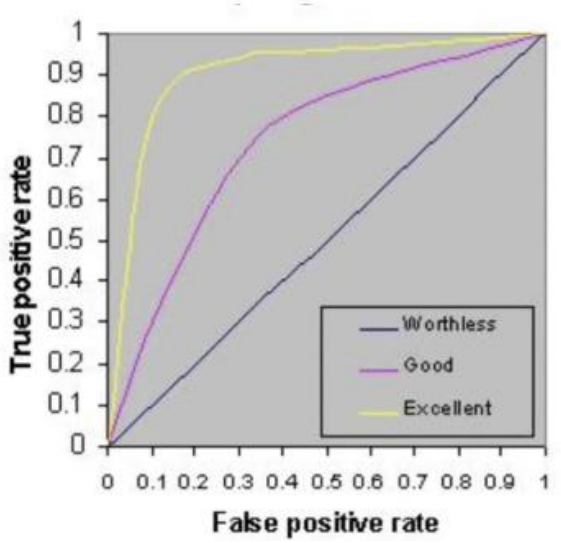
D. Lojistik Regresyon

Logistic Regression Classifier, ikili sınıflandırma problemlerinde kullanılan bir algoritmadır. Bu algoritma, girdi verilerini analiz ederek bir veri öğesinin belirli bir sınıfa ait olma olasılığını tahmin eder. Tahminler, sigmoid fonksiyonu aracılığıyla 0 ile 1 arasında bir olasılık değeri olarak ifade edilir. Logistic regression, sınıflandırma için basit, yorumlanabilir ve hızlı bir yaklaşım sunar. Ancak çoklu sınıflı sınıflandırma problemleri veya non-lineer ilişkiler gerektiren görevler için daha karmaşık modellere ihtiyaç duyulan durumlarda tercih edilmeyebilir.



Şekil 15 Lojistik Regresyon

E. ROC – AUC



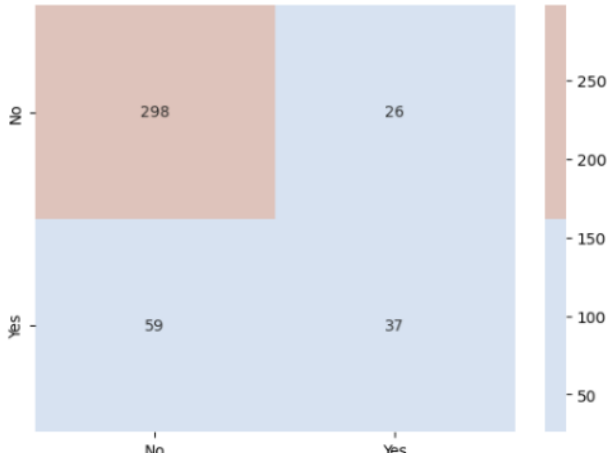
Şekil 16

ROC eğrisi sınıflandırma problemleri için çok önemli bir performans ölçümüdür. ROC bir olasılık eğrisidir ve altında kalan alan olan AUC ayrılabilirliğin derecesini veya ölçüsünü temsil eder.

ROC eğrisinde X ekseninde FPR(Yanlış Pozitif Oran) ve Y ekseninde ise TPR (Gerçek Pozitif Oranı) bulunmaktadır.

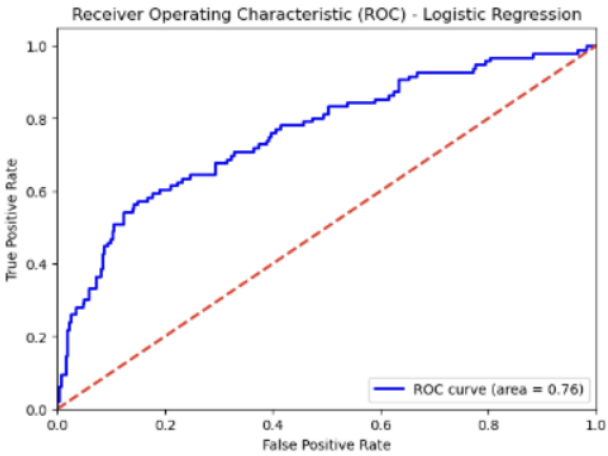
IV. YÖNTEM SONUÇLARI

A. Lojistik Regresyon



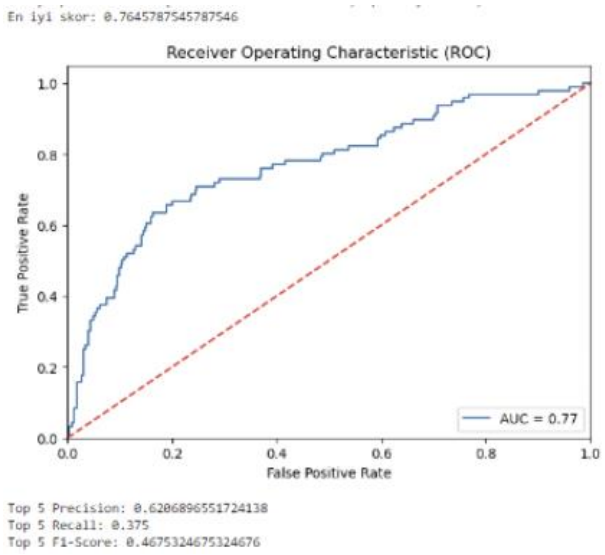
Şekil 17 Confusion Matrix

Lojistik regresyon sonucunda elde edilen sonuçlar aşağıdaki gibidir.
Accuracy: 0.78
Precision: 0.65
Recall: 0.41
F1-Score: 0.51



Şekil 18

GridSearchCV ile hiperparametre optimizasyonu yapıldıktan sonra elde edilen sonuçlar Şekil 19'deki gibidir.

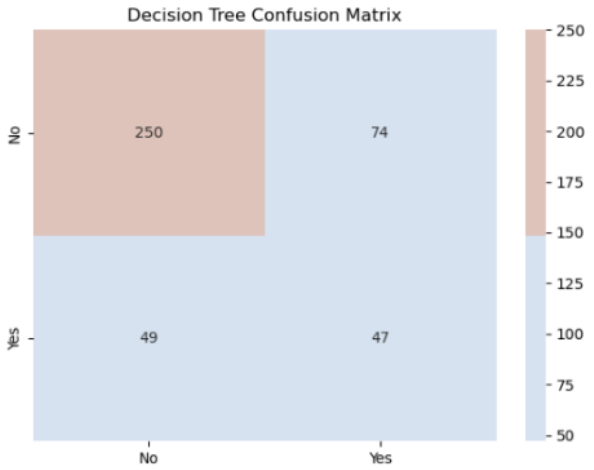


Şekil 19 Hiperparametre Optimizasyonu Sonrası

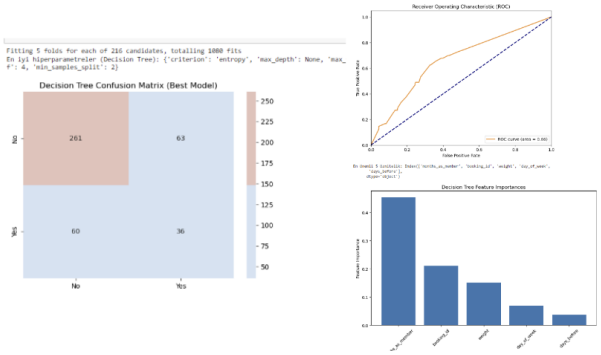
B. Decision Tree

Decision Tree Classifier, bir sınıflandırma algoritmasıdır ve veri madenciliği ve makine öğrenimi uygulamalarında yaygın olarak kullanılır. Bu algoritma, bir veri kümesini sınıflandırmak veya tahmin etmek için ağaç benzeri bir yapı kullanır. Veriyi sınıflandırmak için, ağaç yapısı içinde karar düğümleri (decision nodes) ve yaprak düğümleri (leaf nodes) bulunur. Karar düğümleri, verinin belirli özelliklerine dayalı olarak sınıflandırma kararlarını alır, yaprak düğümleri ise sonuçları temsil eder. Her karar düğümü, veriyi belirli bir özellik veya nitelik üzerinde böler ve bu şekilde ağaç yapısı, veriye uygulandığında sınıflandırma yapar. Decision Tree Classifier, kolay

anlaşılabilir, yorumlanabilir ve görsel olarak temsil edilebilir olması nedeniyle tercih edilir. Ayrıca, veri madenciliği ve sınıflandırma problemlerinde kullanılabilirken, aşırı öğrenmeye ve veriye hassas olma riskini azaltmak için çeşitli tekniklerle (örneğin, kesme derinliği) geliştirilebilir.



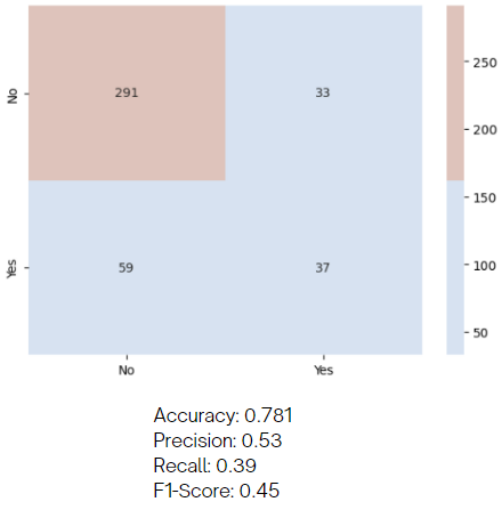
Şekil 20 Decision Tree - Confusion Matrix



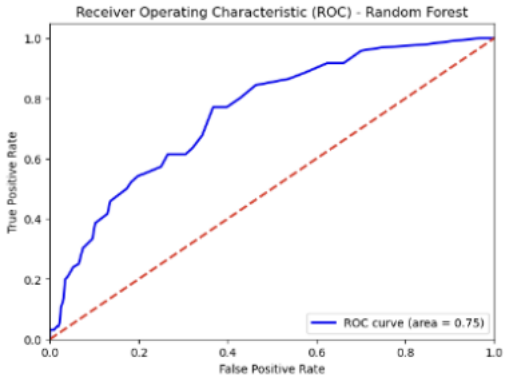
Şekil 21 Hiperparametre Optimizasyon Sonrası Decision Tree Sonuçları

C. *Random Forest*

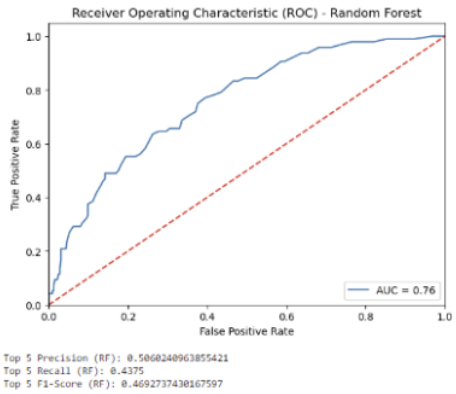
Random Forest Classifier, makine öğrenimi alanında kullanılan bir sınıflandırma algoritmasıdır. Bu algoritma, birden fazla karar ağacını bir araya getirerek daha güçlü bir sınıflandırma modeli oluşturur. Random Forest, veri kümesini rastgele alt örneklemlere böler ve bu alt örneklemler üzerinde ayrı ayrı karar ağaçları oluşturur. Her bir ağaç, farklı özelliklerle ve alt örneklemlerle eğitilir. Ardından, bu ağaçların sonuçları bir araya getirilerek bir tahmin yapılır. Random Forest, aşırı uydurmayı önler, yüksek doğruluk sağlar ve farklı türde veri setlerinde etkili bir şekilde çalışabilir. Bu nedenle sınıflandırma problemlerinde yaygın olarak kullanılır.



Şekil 22 Hiperparametre Optimizasyonu Yapılmadan Önce Elde Edilen Sonuçlar



Şekil 23 Hiperparametre Optimizasyonu Yapılmadan Önce Elde Edilen Sonuçlar

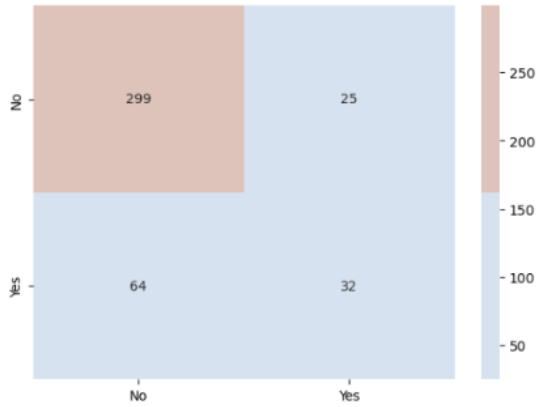


Şekil 24 Hiperparametre OptimizasyonuSonrasında Elde Edilen Sonuçlar

Şekil 24'te görüldüğü üzere AUC değeri 0.76 değerine yükselmiştir.

D. SVM

SVM veya Destek Vektör Makineleri (Support Vector Machines), makine öğrenimi alanında kullanılan bir sınıflandırma ve regresyon algoritmasıdır. SVM, iki veya daha fazla sınıf arasında sınıflandırma yapmak için kullanılır ve özellikle veri noktalarını sınıflar arasında bir hiper düzlemle ayırmaya odaklanır. Bu hiper düzlem, sınıflar arasındaki marjinal mesafeyi maksimize etmek ve yanlış sınıflandırılan veri noktalarını minimize etmek için hesaplanır. SVM, doğrusal veya non-lineer veri setlerinde etkili bir şekilde çalışabilir ve yüksek boyutlu veri analizinde kullanışlıdır. Ayrıca, çekirdek fonksiyonları kullanarak veri setlerini yüksek boyutlu uzaylara dönüştürme yeteneği sayesinde karmaşık veri yapılarını ele alabilir. SVM, sınıflandırma ve regresyon problemlerinde kullanılırken, hiperparametrelerin uygun şekilde ayarlanması, modelin performansını etkileyebilir.

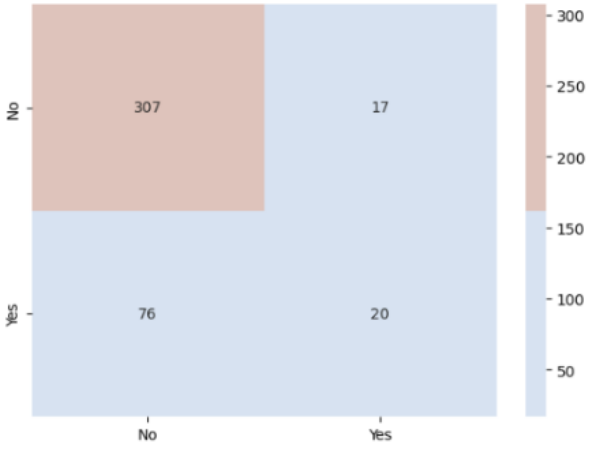


Accuracy: 0.79
Precision: 0.5614035087719298
Recall: 0.3333333333333333
F1-Score: 0.41830065359477125

Şekil 25 SVM sonucunda elde edilen metrik değerleri

E. KNN

K-Nearest Neighbors (KNN), bir gözlem biriminin sınıflandırılmasında veya tahmin edilmesinde kullanılan bir makine öğrenimi algoritmasıdır. Temel fikir, bir gözlem biriminin sınıfını veya değerini belirlemek için çevresindeki en yakın k komşuyu kullanmaktır. K, kullanıcının belirlediği bir parametredir ve komşuların sayısını temsil eder. KNN algoritması, veri noktalarının benzerlik ölçüleri (genellikle Euclidean mesafesi) kullanılarak sınıflandırma veya tahmin yapar. Örneğin, bir veri noktasını sınıflandırmak istediğinizde, bu noktaya en yakın k veri noktasının sınıfını gözlemleyerek tahmin yaparsınız. KNN basit ve anlaşılması kolay bir algoritmadır, ancak büyük veri setleri veya yüksek boyutlu verilerde hesaplama maliyeti yüksek olabilir.



KNN Accuracy: 0.79
KNN Precision: 0.54
KNN Recall: 0.21
KNN F1-Score: 0.30

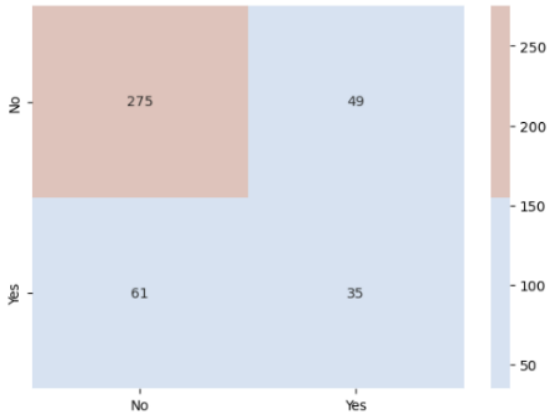
Şekil 26 KNN sonucunda elde edilen metrik değerleri

```
KNN Model Performance (Best Model):
KNN Accuracy: 0.7714285714285715
KNN Confusion Matrix:
 [[303  21]
 [ 75  21]]
KNN Precision: 0.5
KNN Recall: 0.21875
KNN F1-Score: 0.30434782608695654
```

Şekil 27 KNN Hiperparametre Optimizasyonu Sonrasında Elde Edilen Değerler

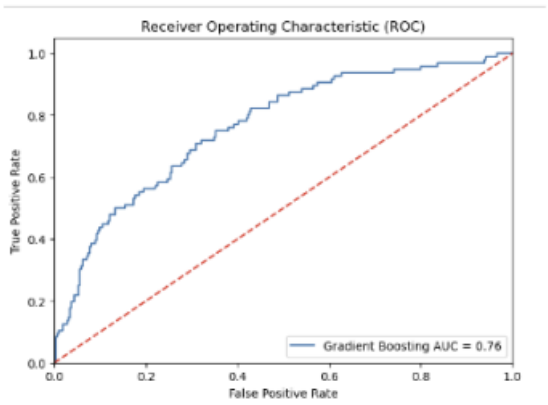
F. XGBoost

XGBoost, Extreme Gradient Boosting'in kısaltması olan bir makine öğrenimi algoritmasıdır ve özellikle sınıflandırma ve regresyon problemlerinde yüksek performanslı tahminler elde etmek için kullanılır. Bu algoritma, birçok zayıf tahmin modelini birleştirerek güçlü bir tahmin modeli oluşturur. XGBoost, ağaç tabanlı bir yöntemdir ve bu ağaçlar, veri setini sınıflandırmak veya regresyon tahminleri yapmak için kullanılır. Algoritma, aşırı uyum (overfitting) problemlerini azaltmak ve doğruluk oranını artırmak için çeşitli düzenleme teknikleri kullanır. XGBoost, büyük veri setleri ve karmaşık veri yapıları ile başa çıkabilme yeteneğine sahiptir ve çeşitli hiperparametrelerin ayarlanmasına olanak tanır, bu nedenle geniş bir uygulama yelpazesi için tercih edilir.

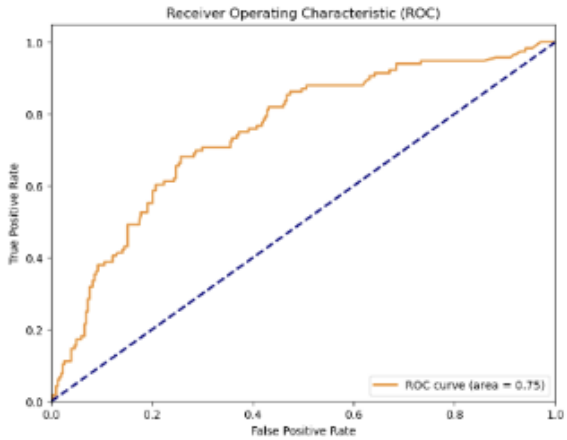


Accuracy: 0.72
Precision: 0.49
Recall: 0.37
F1-Score: 0.42

Şekil 28 XGBoost sonrasında Elde edilen metrik değerleri



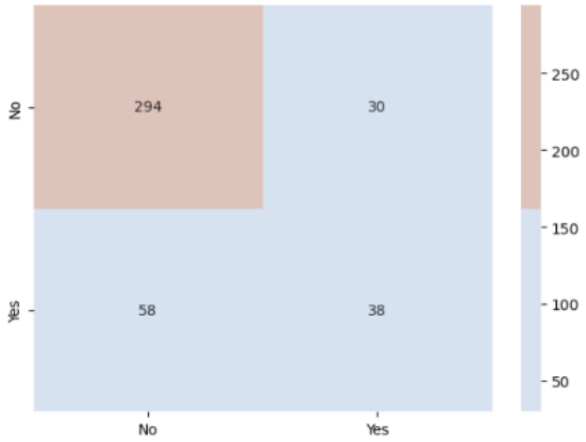
Şekil 29 XGBoost sonrasında Elde edilen metrik değerleri



Şekil 30 XGBoost HiperParametre optimizasyonu sonrasında Elde edilen metrik değerleri

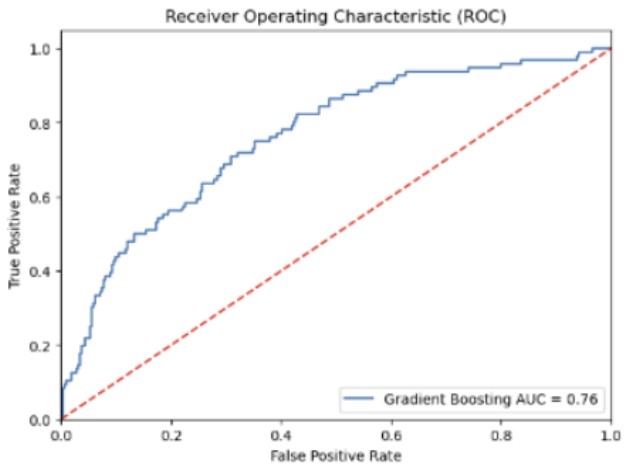
G. Gradient Boosting

Gradient Boosting, makine öğrenimi alanında yaygın olarak kullanılan bir topluluk öğrenme (ensemble learning) tekniğidir. Temel fikir, zayıf öğrenicileri (genellikle karar ağaçları) kullanarak güçlü bir tahmin modeli oluşturmaktır. Başlangıçta, bir veri kümesini tahmin eden bir zayıf öğrenici oluşturulur ve bu öğrenici hataları tespit edilir. Ardından, bu hatalar üzerine odaklanarak bir sonraki öğrenici oluşturulur ve böylece devam eder. Her bir öğrenici, önceki öğrencinin hatalarını telafi etmeye çalışır, bu da modelin tahmin yeteneğini artırır. Gradient Boosting, regresyon ve sınıflandırma problemleri için kullanılır, aşırı uyum sorununu azaltmak için çeşitli düzenleme teknikleri içerir ve genellikle yüksek doğruluk sağlayan güçlü tahmin modelleri oluşturur.

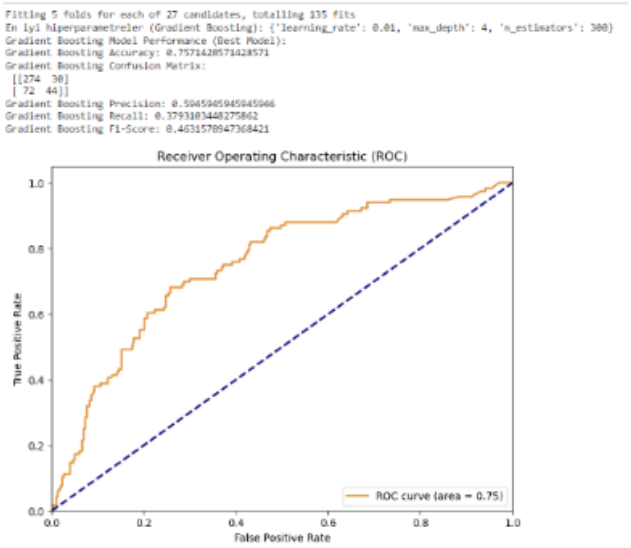


Accuracy:0.79
Precision: 0.56
Recall: 0.40
F1-Score: 0.46

Şekil 31 Gradient Boosting sonrası elde edilen metrik değerleri



Şekil 32 Gradient Boosting sonrası elde edilen ROC ve AUC



Şekil 33 Hiperparametre Optimizasyonu sonrası elde dilen sonuçlar

V. SONUÇLAR

Bu çalışmada, çeşitli makine öğrenmesi algoritmalarının performansları karşılaştırmalı olarak değerlendirilmiştir. Değerlendirilen modeller arasında Lojistik Regresyon, Destek Vektör Makineleri (SVM), k-En Yakın Komşu (kNN), Gradyan Artırma, Rastgele Orman ve XGBoost algoritmaları bulunmaktadır. Her bir modelin performansı, doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1 skoru gibi metrikler kullanılarak ölçülmüştür.

Lojistik Regresyon modeli %79.8 doğruluk ile en yüksek performansı sergileyen model olmuştur. Bu model aynı zamanda %58.7 kesinlik ve %38.5 duyarlılık değerleri ile dikkate değer sonuçlar sunmuştur, ancak F1 skoru %46.5 ile nispeten düşük kalmıştır.

Gradient Boosting modeli, benzer bir doğruluk oranı (%79) ile ikinci en iyi performansı göstermiştir. Kesinlik ve duyarlılık oranları sırasıyla %56 ve %40 iken, F1 skoru Lojistik Regresyon modeli ile aynı %46 olarak ölçülmüştür.

Üçüncü sırada, Rastgele Orman modeli %78.1 doğruluk ile yer almıştır. Bu model, %52.9 kesinlik ve %38.5 duyarlılık değerleri ile dengeli bir performans göstermiş ve %44.6 F1 skoru elde etmiştir.

SVM modelinin doğruluk oranı %79 olmasına rağmen, kesinlik (%56), duyarlılık (%33) ve F1 skoru (%42) bakımından daha düşük sonuçlar vermiştir.

kNN ve XGBoost modelleri sırasıyla %79 ve %72 doğruluk oranları ile benzer sonuçlar göstermişlerdir. Ancak, kNN modelinin düşük duyarlılık (%21) ve F1 skoru (%30) göz önünde bulundurulduğunda, bu modelin diğer metriklerde daha az etkili olduğu görülmüştür.

Karar Ağacı Sınıflandırıcısı, %71 doğruluk oranı ile en düşük performansı sergileyen model olmuştur. %39 kesinlik ve %49 duyarlılık oranlarına sahip olan bu modelin F1 skoru %43 olarak hesaplanmıştır.

Sonuç olarak, bu çalışmada değerlendirilen modeller arasında Lojistik Regresyon ve Gradyan Artırma modelleri, diğer algoritmalarla karşılaştırıldığında daha yüksek doğruluk oranları sunmuşlardır. Ancak, yüksek doğruluk değerleri her zaman diğer performans metrikleriyle doğru orantılı olmayabilir. Bu nedenle, bir modelin seçilmesi sırasında, belirli bir uygulamanın gereksinimlerine göre tüm performans metrikleri dikkate alınmalıdır.

VI. KAYNAKÇA

https://www.veribilimiokulu.com/gradient-boosted-regresyon-agaclari/
https://ecehanyildirim.medium.com/temel-topluluk-%C3%B6%C4%9Frenimi-ve-random-forest-gradient-boosting-algoritmalar%C4%B1-e105f93b33f0
https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e
https://arslanev.medium.com/makine-%C3%B6%C4%9Frenmesi-knn-k-nearest-neighbors-algoritmas%C4%B1-bdfb688d7c5f
https://aws.amazon.com/tr/what-is/logistic-regression/