

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



"SMOTE VE ETKİLİ VERİ MADENCİLİĞİ TEKNİKLERİ
KULLANILARAK KALP YETERSİZLİĞİ HASTALARININ
SAĞKALIM TAHMİNİNİN İYİLEŞTİRİLMESİ"
MAKALESİNİN İNCELEME RAPORU

20011038 – BÜŞRA MEDİNE GÜRAL

VERİ MADENCİLİĞİ VE BİLGİ KEŞFİ PROJESİ

Ders Yürütücsü
Prof. Dr. Songül VARLI

Aralık, 2024

İÇİNDEKİLER

1	GİRİŞ	1
2	VERİ SETİ VE YÖNTEM	2
2.1	Veri Seti	2
2.2	Öznitelik Seçimi	3
2.3	Sınıflandırmada Kullanılan Algoritmalar	6
2.4	Aşırı Örnekleme	8
2.5	Değerlendirme Ölçütleri	8
3	DENEYSEL ÇALIŞMA VE SONUÇLAR	9
3.1	Tüm Öznitelikler Kullanılarak Yapılan Eğitim	9
3.2	Tüm Öznitelikler ve SMOTE Kullanılarak Yapılan Eğitim	13
3.2.1	Sabit Hiperparametrelerle Eğitim	13
3.2.2	Farklı Hiperparametrelerle Eğitim	18
3.3	Önemli Öznitelikler Kullanılarak Yapılan Eğitim	20
4	SONUÇ	24
	Referanslar	25

1 GİRİŞ

İlgili rapor, Abid Ishaq ve arkadaşlarının Improving the Prediction of Heart Failure Patients' Survival Using SMOTE and Effective Data Mining Techniques (SMOTE ve Etkili Veri Madenciliği Teknikleri Kullanılarak Kalp Yetersizliği Hastalarının Sağkalım Tahmininin İyileştirilmesi) [1] başlıklı makalesinin detaylı bir incelemesini sunmaktadır.

Dünya Sağlık Örgütü'ne (WHO) göre, kardiyovasküler hastalıklar (KVH), dünya genelinde en sık görülen ölüm nedenlerinden biridir [2]. Yüksek tansiyon, yüksek kolesterol, diyabet ve düzensiz kalp atışı gibi pek çok faktör, KVH'nin tespit edilmesini zorlaştırmaktadır. Bununla birlikte, KVH belirtileri cinsiyete göre değişiklik göstermektedir. Örneğin, erkeklerde genellikle göğüs ağrısı öne çıkarken, kadınlarda göğüs rahatsızlığı, mide bulantısı, aşırı bitkinlik ve nefes alma güçlüğü gibi ek belirtiler görülebilir.

Makaledeki çalışma, 299 hastadan oluşan bir veri setindeki kalp hastalıklarından sağ kurtulan bireyleri analiz etmektedir. Çalışmanın amacı, kardiyovasküler hastaların sağkalım tahmin doğruluğunu artırabilecek önemli özelliklerini ve veri madenciliği tekniklerini incelemektir. Bu bağlamda, Karar Ağacı (Decision Tree - DT), Adaptif Artırma Sınıflandırıcısı (Adaptive Boosting - AdaBoost), Lojistik Regresyon (Logistic Regression - LR), Stokastik Gradyan Sınıflandırıcısı (Stochastic Gradient Classifier - SGD), Rastgele Orman (Random Forest - RF), Gradyan Artırma Sınıflandırıcısı (Gradient Boosting - GBM), Ekstra Ağaç Sınıflandırıcısı (Extra Tree Classifier - ETC), Gaussian Naive Bayes Sınıflandırıcısı (G-NB) ve Destek Vektör Makineleri (Support Vector Machine - SVM) olmak üzere dokuz farklı sınıflandırıcı kullanılmıştır. Sentetik Azınlık Aşırı Örnekleme Tekniği (Synthetic Minority Oversampling Technique - SMOTE) kullanılarak sınıf dengesizliği problemi ele alınmıştır. Ek olarak, Random Forest ile en etkili özellikler belirlenmiş ve sınıflandırıcılar, bu özellikler ile eğitilmiştir. Sonuçlar çeşitli açılardan karşılaştırılmıştır. Bu raporda da belirtilen teknikler uygulanmış ve sonuçlar analiz edilmiştir.

2 VERİ SETİ VE YÖNTEM

2.1 Veri Seti

Araştırmada kullanılan veri seti UCI Machine Learning Repository'den elde edilen **Heart-failure-clinical-records-dataset** isimli veri setidir [3]. İçerisinde takip süreci boyunca kalp sorunları yaşayan 194 erkek, 105 kadın olmak üzere 299 hastasının tıbbi kayıtları bulunmaktadır. Toplamda 13 klinik öznitelik vardır. Bu öznitelikler Tablo 2.1'de verilmiştir.

Table 2.1 Veri Setindeki Öznitelikler

	Öznitelik	Null Olmayan Veri Sayısı	Dtype
0	TIME	299	int64
1	Event	299	int64
2	Gender	299	int64
3	Smoking	299	int64
4	Diabetes	299	int64
5	BP	299	int64
6	Anaemia	299	int64
7	Age	299	float64
8	Ejection.Fraction	299	int64
9	Sodium	299	int64
10	Creatinine	299	float64
11	Pletelets	299	float64
12	CPK	299	int64

Veri setindeki özniteliklere ait detaylar Tablo 2.2'de verilmiştir.

Table 2.2 Öznitelik Detayları

No	Öznitelik	Açıklama	Aralık	Ölçü Birimi
1	Time	Takip süresi	4-285	Gün
2	Event (Hedef)	Takip süresi içinde hastanın ölüp ölmemişti.	0,1	Boolean
3	Gender	Erkek ya da kadın	0, 1	Binary
4	Smoking	Hastanın sigara içip içmediği	0, 1	Boolean
5	Diabetes	Hastanın diyabeti olup olmadığı	0, 1	Boolean
6	BP	Hastanın yüksek tansiyon sorunu olup olmadığı	0, 1	Boolean
7	Anemia	Kırmızı kan hücreleri veya hemoglobindeki azalma	0, 1	Boolean
8	Age	Hastanın yaşı	40-95	Yıl
9	Ejection.Fraction	Her kasılmada kalpten çıkan kanın yüzdesi	14-80	Yüzde (%)
10	Sodium	Kanda bulunan sodyum seviyesi	114-148	mEq/L
11	Creatinine	Kanda bulunan kreatinin seviyesi	0.50-9.40	mg/dL
12	Pletelets	Kanda bulunan trombosit miktarı	25.01-850.00	Kilotrombosit/mL
13	CPK	Kanda bulunan CPK enzimi seviyesi	23-7861	Mcg/L

2.2 Öznitelik Seçimi

Orijinal çalışmada hedef değişkeni en çok etkileyen özniteliklerin seçilebilmesi için Random Forest algoritması kullanılmıştır.

RF algoritması, veri setindeki özelliklerini kullanarak çok sayıda karar ağıacı oluşturur. Her ağaç, eğitim verilerinin rastgele bir alt kümesiyle eğitilir ve her bir düğümde veri seti, bir özelliğe göre bölünür. Bölünme sırasında, her bir özellik, sınıflandırmayı veya regresyonu ne kadar iyileştirdiğine göre değerlendirilir. Bu iyileştirme genellikle Gini azalması veya bilgi kazancı gibi metrikler kullanılarak ölçülür. RF, tüm ağaçlar üzerindeki her bir özelliğin bölünme katlarını toplar ve ortalamasını alarak özelliklerin önemini hesaplar. Bu katkılardır, her bir özelliğin hedef sınıfı tahmin etmede ne kadar önemli olduğunu yansıtır.

RF tarafından tahmin edilen özellik önem değerleri ve bu değerlere ait grafik aşağıda verilmiştir.

Table 2.3 Özelliklerin Önemleri

ID	Öznitelik	Önem
0	TIME	0.378150
9	Creatinine	0.128340
6	Age	0.099112
11	CPK	0.088972
7	Ejection.Fraction	0.086521
10	Pletelets	0.083866
8	Sodium	0.066281
3	Diabetes	0.015511
4	BP	0.014744
1	Gender	0.014286
5	Anaemia	0.013150
2	Smoking	0.011067

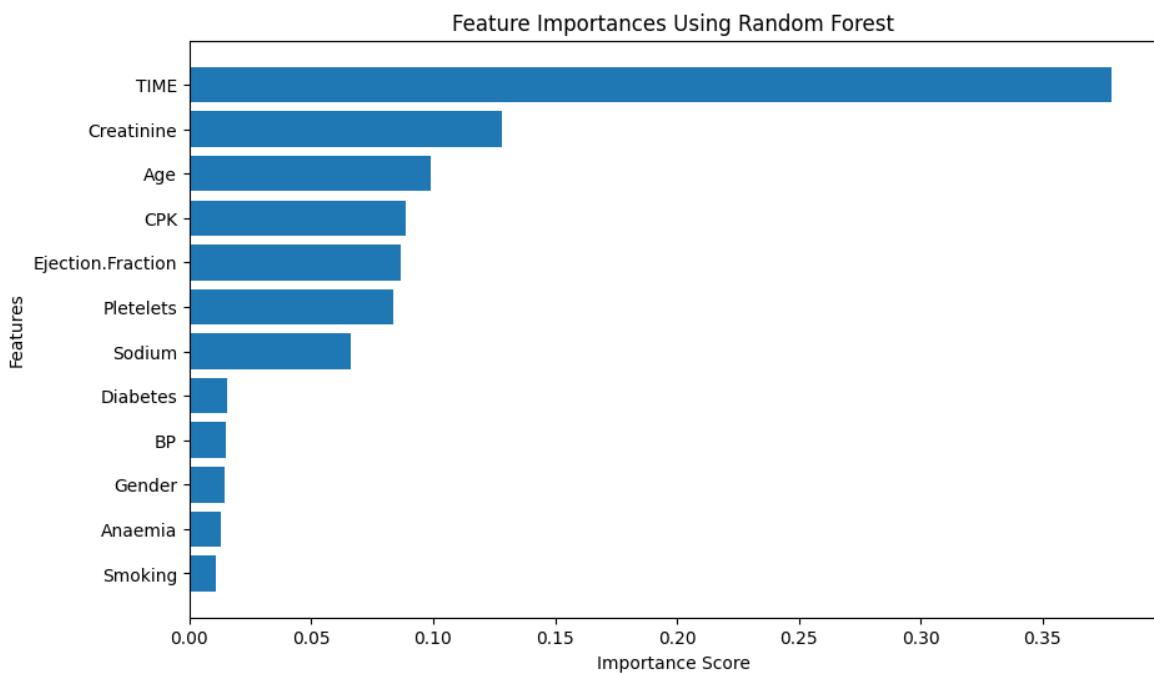


Figure 2.1 Özelliklerin Önem Grafiği

Grafikte, RF algoritması kullanılarak özelliklerin önem skorlarının dağılımı gösterilmektedir. En yüksek önem skoruna sahip özellik "TIME" olarak öne çıkmakta ve hedef değişkeni tahmin etmede diğer özelliklere kıyasla en büyük katkıyı sağlamaktadır. Onu sırasıyla "Creatinine", "Age", ve "CPK" özellikleri takip

etmektedir. Bu, sıralanan özelliklerin model tarafından önemli karar noktaları olarak değerlendirildiğini göstermektedir. Daha düşük önem skorlarına sahip olan "Smoking", "Anaemia", ve "Gender" gibi özelliklerin hedef değişken üzerindeki etkisinin daha sınırlı olduğu gözlemlenmektedir.

Asıl makalede, RF modelinin hiperparametreleri hakkında herhangi bir bilgi verilmediği için modelin oluşturulmasında varsayılan parametreler kullanılmış, rastgelelik için ise random_state parametresi 42 olarak belirlenmiştir. Bu durum, özellikle özellik önem sıralamasında bazı farklılıkların ortayamasına neden olmuştur. Orijinal çalışmada önem grafiği Şekil 2.2'deki gibidir.

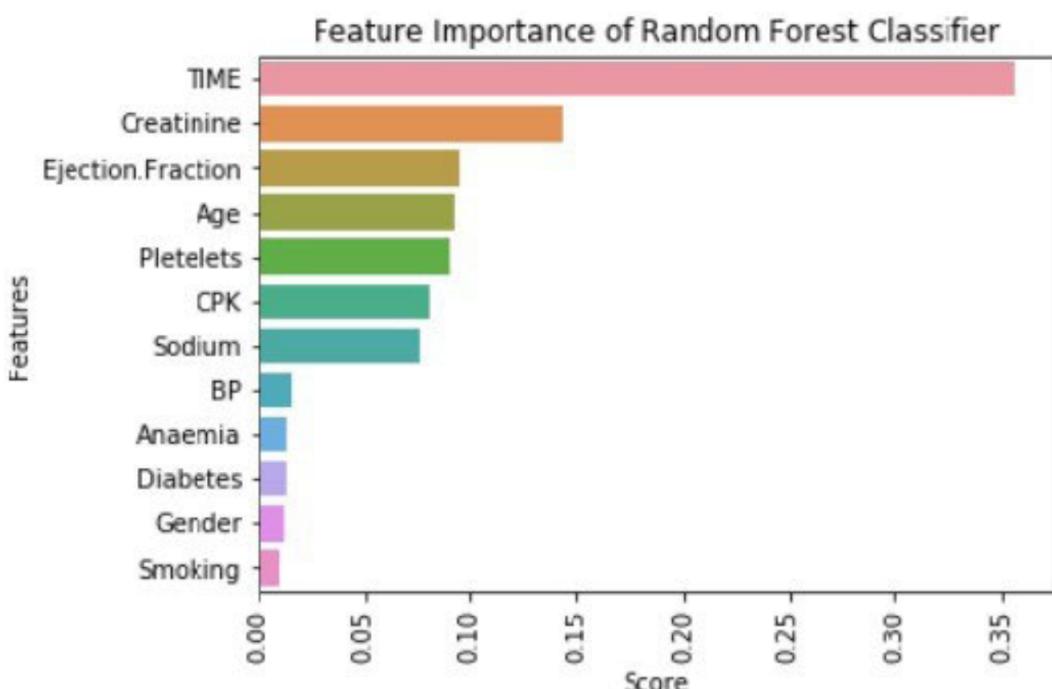


Figure 2.2 Orijinal Çalışmada Özelliklerin Önem Grafiği

Özellik önem grafikleri incelendiğinde, TIME ve Creatinine özelliklerinin her iki çalışmada da en önemli özellikler olarak belirlendiği görülmektedir. Ancak diğer özelliklerin sıralaması ve önem düzeylerinde farklılıklar bulunmaktadır. Orijinal çalışmada Ejection.Fraction özelliği üçüncü en önemli özellik olarak değerlendirilmiştir, bu çalışmada Age ve CPK özellikleri daha öncelikli bir konuma gelmiştir. Bu durum, modelin belirli özelliklere olan duyarlılığının kullanılan parametrelerle değişiklik gösterebileceğini işaret etmektedir. En az öneme sahip 4 özelliğe bakıldığına ise Smoking, Gender ve Anemia ortakken orijinal çalışmada Diabetes'in yerini BP özelliği almıştır.

2.3 Sınıflandırmada Kullanılan Algoritmalar

Orijinal çalışmada, kalp hastalığı tahmini için sınıflandırma yöntemleri önerilmiş ve sınıflandırma doğruluğunu artırmak için topluluk (ensemble) modellerinden faydalanılmıştır. Veri seti, eğitim ve test olarak ikiye bölünmüştür; her bir sınıflandırıcı, eğitim veri seti kullanılarak eğitilmiştir. Sınıflandırıcıların verimliliği ise test veri seti üzerinde değerlendirilmiştir.

Çalışmada kullanılan algoritmalar şunlardır: Karar Ağacı (Decision Tree - DT), Adaptif Artırma Sınıflandırıcısı (Adaptive Boosting - AdaBoost), Lojistik Regresyon (Logistic Regression - LR), Stokastik Gradyan Sınıflandırıcısı (Stochastic Gradient Classifier - SGD), Rastgele Orman (Random Forest - RF), Gradyan Artırma Sınıflandırıcısı (Gradient Boosting - GBM), Ekstra Ağaç Sınıflandırıcısı (Extra Tree Classifier - ETC), Gaussian Naive Bayes Sınıflandırıcısı (G-NB) ve Destek Vektör Makineleri (Support Vector Machine - SVM).

Decision Trees, hem kategorik hem de sayısal verilerle çalışan, anlaşılması ve uygulanması kolay bir makine öğrenimi algoritmasıdır [4]. Veriyi analiz etmek ve sınıflandırma ya da regresyon gibi görevleri gerçekleştirmek için ağaç şeklinde bir yapı kullanır. Karar ağaçları, bir problem üzerinde karar verme sürecini modellemek için dallara ayrılan adımlar oluşturur.

Adaptive Boosting, zayıf sınıflandırıcıları bir araya getirerek güçlü bir model oluşturan bir makine öğrenimi algoritmasıdır [5]. Genellikle "decision stumps" kullanılır ve hatalı sınıflandırılan örneklerde daha fazla ağırlık vererek hataları düzeltmeye odaklanır. Tüm sınıflandırıcıların ağırlıklı oylarıyla nihai karar verilir.

Logistic Regression, ikili sınıflandırma problemleri için yaygın olarak kullanılan istatistiksel bir yöntemdir. Temel amacı, bir olayın meydana gelme olasılığını tahmin etmektir. Model, bağımsız değişkenler ile bağımlı değişken arasında doğrusal bir ilişkiyi öğrenir ve çıktı, sigmoid fonksiyon aracılığıyla 0 ile 1 arasında bir olasılık değeri olarak ifade edilir.

Stochastic Gradient Descent (SGD), büyük veri setleriyle çalışan makine öğrenimi modellerini optimize etmek için kullanılan popüler bir optimizasyon algoritmasıdır. Temel amacı, modelin hata fonksiyonunu minimize ederek en uygun parametreleri bulmaktır. SGD, her yinelemede tüm veri seti yerine yalnızca rastgele bir örnek veya küçük bir veri alt kümesi üzerinde hesaplama yapar, bu da algoritmayı hızlı ve hafif hale getirir.

Random Forest, hem sınıflandırma hem de regresyon problemlerinde kullanılan güçlü ve esnek bir makine öğrenimi algoritmasıdır. Birden fazla karar ağacının birlikte çalışmasıyla oluşturulur ve her ağaç, veri setinin rastgele bir alt kümesiyle eğitilir. Sonuçlar, sınıflandırma için oylama, regresyon için ise ortalama alma yoluyla birleştirilir. Bu yöntem, aşırı öğrenmeyi (overfitting) azaltarak genelleme yeteneğini artırır ve veri setindeki özelliklerin önemini değerlendirmeye olanak tanır.

Gradient Boosting Machine, hem sınıflandırma hem de regresyon problemlerinde etkili sonuçlar sunan, güçlü bir artırma (boosting) algoritmasıdır. GBM, zayıf öğreniciler olarak genellikle karar ağaçlarını kullanır ve her bir ardışık ağaç, önceki ağaçın hatalarını düzeltmek için eğitilir. Model, hataları minimize etmek için gradyan iniş yöntemini kullanır ve bu sayede tahmin doğruluğunu kademeli olarak artırır [6].

Extra Tree Classifier (Extremely Randomized Trees), hem sınıflandırma hem de regresyon problemlerinde kullanılan, karar ağaçlarına dayalı bir topluluk yöntemidir [7]. Random Forest'a benzer şekilde çalışır ancak her ağaç, veri setinin rastgele bir alt kümesiyle eğitilir. Farklı olarak, dallanma sırasında en iyi bölünmeyi bulmak yerine bölünme noktalarını tamamen rastgele seçer.

Gaussian Naive Bayes, özellikle sürekli özelliklere sahip veri setlerinde kullanılan bir olasılıksal sınıflandırma algoritmasıdır. Naive Bayes algoritmasının bir türü olan bu yöntem, özelliklerin birbirinden bağımsız olduğunu varsayar ve her bir özelliğin normal (Gaussian) dağılıma sahip olduğunu kabul eder. Model, sınıf olasılıklarını Bayes teoremi kullanarak hesaplar ve bir örneğin hangi sınıfa ait olduğunu tahmin eder.

Support Vector Machines, sınıflandırma ve regresyon problemlerinde kullanılan güçlü bir makine öğrenimi algoritmasıdır. SVM, veri noktalarını farklı sınıflara ayırmak için en geniş marjinal aralığı sağlayan en iyi hiper düzlemi bulmayı hedefler. Çizilebilir sınıfların lineer olmadığı durumlarda, SVM, çekirdek (kernel) fonksiyonlarını kullanarak veriyi daha yüksek boyutlu bir uzaya dönüştürür ve burada ayrimı mümkün kılar.

2.4 Aşırı Örnekleme

Var olan veri setindeki sınıf dengesizliği problemi SMOTE yöntemi ile ele alınmıştır. Bu teknik, azınlık sınıfına ait verilerin kopyalanması yerine, sentetik örnekler oluşturarak veri setini dengeler. SMOTE, her azınlık sınıfı örneği için Öklid mesafesi kullanarak yakın komşularını analiz eder ve bu komşularla arasında bulunan doğrular üzerinde rastgele yeni veri noktaları oluşturur. Bu sayede, azınlık sınıfı örneklerinin sayısı artırılırken, veri setinin çeşitliliği korunur.

2.5 Değerlendirme Ölçütleri

Makine öğrenimi modellerinin performansını değerlendirmek için çeşitli yöntemler bulunmaktadır. Farklı değerlendirme araçlarının birlikte kullanılması, analitik çalışmalarında daha kapsamlı bir anlayış elde edilmesine olanak tanır. Bu çalışmada, dört temel ölçüt (doğruluk, kesinlik, geri çağırma ve F-Skoru) kullanılarak makine öğrenimi algoritmalarının performansları arasındaki farklar analiz edilmiştir. Değerlendirmek için kullanılan ölçütler ve tanımları aşağıda verilmiştir.

- **Doğruluk (Accuracy):** Modelin doğru yaptığı tahminlerin, toplam tahmin sayısına oranını ifade eder.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Kesinlik (Precision):** Pozitif olarak tahmin edilen örneklerin ne kadarının gerçekten pozitif olduğunu gösterir.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Geri Çağırma (Recall):** Gerçek pozitif örneklerin, toplam gerçek pozitiflerin (TP ve FN) içinde ne kadarını doğru tahmin ettiğini ifade eder.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F-Skoru (F-Score):** Kesinlik ve geri çağırmanın harmonik ortalamasıdır.

$$F\text{-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

3

DENEYSEL ÇALIŞMA VE SONUÇLAR

Bu bölümde, orijinal çalışmada adımlar uygulanarak kalp hastalarının sağkalım tahmini için yapılan deneylerin tasarımları ve elde edilen sonuçlar tartışılmıştır.

299 veriden oluşan veri seti %30 test ve %70 eğitim için kullanılacak şekilde bölünmüştür. Çalışmanın ilk aşamasında tüm özellik seti kullanılarak sınıflandırıcıların performansları kaydedilmiştir. Ardından SMOTE kullanılarak aşırı örnekleme yapılmış ve bu veriler dahil edilerek eğitim tekrardan gerçekleştirilmiştir. Son aşamada ise 2. bölümde Random Forest ile seçilen özellikler SMOTE kullanılarak makine öğrenimi sınıflandırıcıları üzerinde incelenmiştir. Orihinal çalışmada model hiperparametrelerine ait detay verilmediğinden bu bölümdeki sonuçlar makaledeki sonuçlardan genel anlamda farklılaşmıştır.

3.1 Tüm Öznitelikler Kullanılarak Yapılan Eğitim

Veri setindeki 'Event' kolonu sınıf etiketi olarak seçilmiş ve kalan 12 öznitelik kullanılarak eğitim gerçekleştirilmiştir. Tablo 3.1'de kullanılan modellerin hiperparametreleri verilmiştir.

Table 3.1 Kullanılan Hiperparametreler

Algorithm	Parameters
DT	{'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 10}
AdaBoost	{'algorithm': 'SAMME.R', 'learning_rate': 0.1, 'n_estimators': 100}
LR	{'C': 1.0, 'max_iter': 1000, 'penalty': 'l2', 'solver': 'lbfgs', 'tol': 0.0001}
SGD	{'alpha': 0.0001, 'learning_rate': 'optimal', 'loss': 'hinge', 'max_iter': 1000, 'penalty': 'l2', 'tol': 0.001}
RF	{'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50}
GBM	{'learning_rate': 0.1, 'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 100}
ETC	{'max_depth': None, 'min_samples_split': 2, 'n_estimators': 100, 'min_samples_leaf': 1}
GNB	-
SVM	{'C': 0.1, 'kernel': 'linear'}

Tablo 3.2'de yapılan eğitimler sonucu oluşan çıktılar verilmiştir.

Table 3.2 Tüm Öznitelikler Kullanılarak Yapılan Sınıflandırmanın Sonucu

Algorithm	Accuracy	Precision	Recall	F1
DT	0.7777	0.7241	0.6363	0.6774
AdaBoost	0.8222	0.9473	0.5454	0.6923
LR	0.8222	0.84	0.6363	0.7241
SGD	0.3666	0.366	1.0	0.5365
RF	0.8666	0.92	0.6969	0.7931
GBM	0.8555	0.8571	0.7272	0.7868
ETC	0.8222	0.9047	0.5757	0.7037
GNB	0.7555	0.7391	0.5151	0.6071
SVM	0.8	0.8947	0.5151	0.6538

Elde edilen sonuçlar incelendiğinde, farklı algoritmaların performans metriklerinde belirgin farklılıklar olduğu görülmektedir. Özellikle Random Forest (RF) algoritması, hem accuracy hem de F1-skora göre en yüksek performansı sergilemiştir. Bu, RF'nin hem doğru sınıflandırma yapabilme kabiliyeti hem de sınıf dengesizliğine karşı dayanıklılığını göstermektedir. Benzer şekilde, Gradient Boosting Machine de yüksek bir accuracy ve F1-skora sahip olarak güçlü bir alternatif olarak öne çıkmıştır.

Decision Tree (DT) algoritması, accuracy değeri %77.77 ile orta düzeyde bir performans göstermiştir. Bununla birlikte, precision ve recall metrikleri arasındaki denge, bu algoritmanın dengesiz sınıflar üzerinde bazı sınıflandırma zorluklarını yaşadığını işaret etmektedir. AdaBoost algoritması ise %82.22 accuracy ile DT'ye kıyasla daha başarılı olmuştur; ancak recall değeri %54.54, modelin azınlık sınıfını tanımda zorluk yaşadığı göstermektedir. Bu durum, AdaBoost'un ağırlıklı olarak çoğunluk sınıfına odaklanma eğiliminden kaynaklanabilir.

Daha basit bir model olan Gaussian Naive Bayes (GNB), düşük recall ve F1 skorları ile sınıf dengesizliği karşısında yetersiz kalmıştır. Bu sonuçlar, GNB'nin güçlü varsayımlarına dayalı yapısının karmaşık veri setlerinde sınırlı olabileceğini göstermektedir. Logistic Regression (LR), %82.22 accuracy ve %72.41 F1-skora sahip olup, doğrusal bir model olarak veri setine uygunluk göstermiştir. Bununla birlikte, recall değeri %63.63, modelin bazı azınlık sınıf örneklerini doğru şekilde sınıflandıramadığına işaret etmektedir.

Stochastic Gradient Descent (SGD), %36.66 gibi oldukça düşük bir accuracy ile dikkat

çekmektedir. Bu performans, özellikle SGD'nin hiperparametre optimizasyonuna veya sınıflandırma görevinde uygun olmayan bir kayıp fonksiyonu seçimine duyarlı olduğunu göstermektedir. SGD'nin recall değeri %100 gibi yüksek bir değerle azınlık sınıfını tamamen yakalamasına rağmen, precision %36.6 skorunun düşük olması, modelin azınlık sınıfına ait olmayan örnekleri yanlış şekilde sınıflandırdığını göstermektedir.

Son olarak, Support Vector Machine (SVM), %80 accuracy ile güçlü bir performans sergilemiş olsa da, recall ve F1 skorlarının nispeten düşük olması, modelin sınıf dengesizliği sorununa duyarlı olabileceğini göstermektedir. Extra Tree Classifier (ETC), %82.22 accuracy ile güçlü bir performans sergilemiştir; ancak precision ve recall arasındaki dengesizlik, modelin karar sınırlarının optimize edilmesi gerektiğini göstermektedir.

Genel olarak, sonuçlar Random Forest ve Gradient Boosting gibi ağaç tabanlı algoritmaların hem doğruluk hem de genel performans açısından öne çıktılığını göstermektedir. Buna karşılık, SGD gibi doğrusal yöntemlerin sınıf dengesizliği bulunan veri setlerinde daha düşük performans sergilediği gözlemlenmiştir. Bu sonuçlar, model seçiminin veri setinin özelliklerine ve problem türüne göre dikkatlice yapılması gerektiğini göstermektedir.

Tablo 3.3'te orijinal çalışmada sonuçlar görülmektedir.

Table 3.3 Tüm Öznitelikler Kullanılarak Yapılan Sınıflandırmanın Sonucu (Orijinal)

Algorithm	Accuracy	Precision	Recall	F1
DT	0.7889	0.80	0.79	0.79
AdaBoost	0.8223	0.83	0.82	0.82
LR	0.8556	0.85	0.86	0.85
SGD	0.6667	0.62	0.67	0.63
RF	0.8889	0.89	0.89	0.89
GBM	0.8444	0.84	0.84	0.84
ETC	0.8334	0.83	0.83	0.83
GNB	0.8667	0.86	0.87	0.86
SVM	0.8667	0.87	0.87	0.86

Her iki çalışmada da Random Forest (RF) algoritması, en yüksek accuracy ve F1 değerlerini elde etmiş olup, bu algoritmanın veriye uygunluğunu ortaya koymaktadır. Orijinal makalede RF'nin accuracy değeri %88.89 iken, bu çalışmada %86.66 olarak hesaplanmıştır. Bu fark, rastgelelikten veya hiperparametre farklılıklarından

kaynaklanıyor olabilmektedir. Ancak, her iki çalışmada da RF'nin en iyi performansı sergilemesi, algoritmanın güvenilir bir seçim olduğunu desteklemektedir.

AdaBoost, Logistic Regression (LR), Extra Tree Classifier (ETC) ve Gradient Boosting Machine (GBM) algoritmalarında birbirine yakın başarı seviyeleri gözlemlenmiştir. Ancak, Gaussian Naive Bayes (GNB) algoritmasında orijinal makalede %86.67 accuracy raporlanmış olmasına karşın, bu çalışmada değer %75.55 olarak hesaplanmıştır. Bu fark, özellikle GNB'nin veri dağılımına ve sınıf dengesine olan duyarlılığından kaynaklanmış olabilir.

Stochastic Gradient Descent (SGD) algoritması, her iki çalışmada da diğer algoritmala kiyasla düşük performans sergilemiştir. Orijinal makalede accuracy %66.67 olarak hesaplanmışken, bu çalışmada değer %36.66 ile önemli ölçüde düşmüştür. Bu durum, SGD'nin hiperparametre optimizasyonundaki farklılıklar veya rastgelelige olan yüksek duyarlılığı ile açıklanabilir.

Tablo 3.2'ye ait grafik Şekil 3.1'de ve orijinal çalışmadaki grafik Şekil 3.2'de verilmiştir.

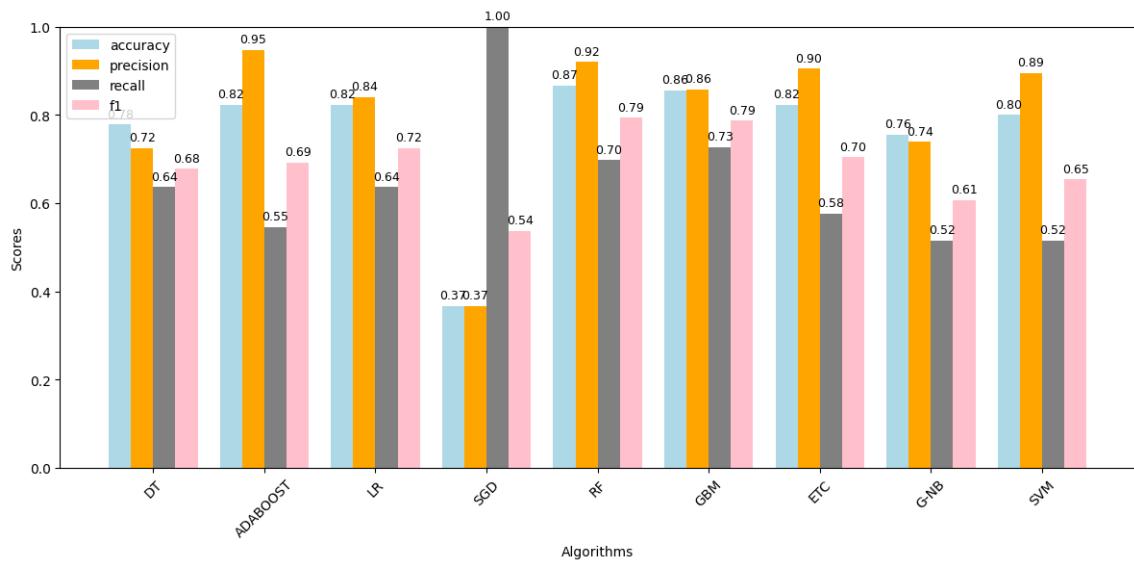


Figure 3.1 Tüm Öznitelikler Kullanılarak Yapılan Sınıflandırmanın Sonuç Grafiği

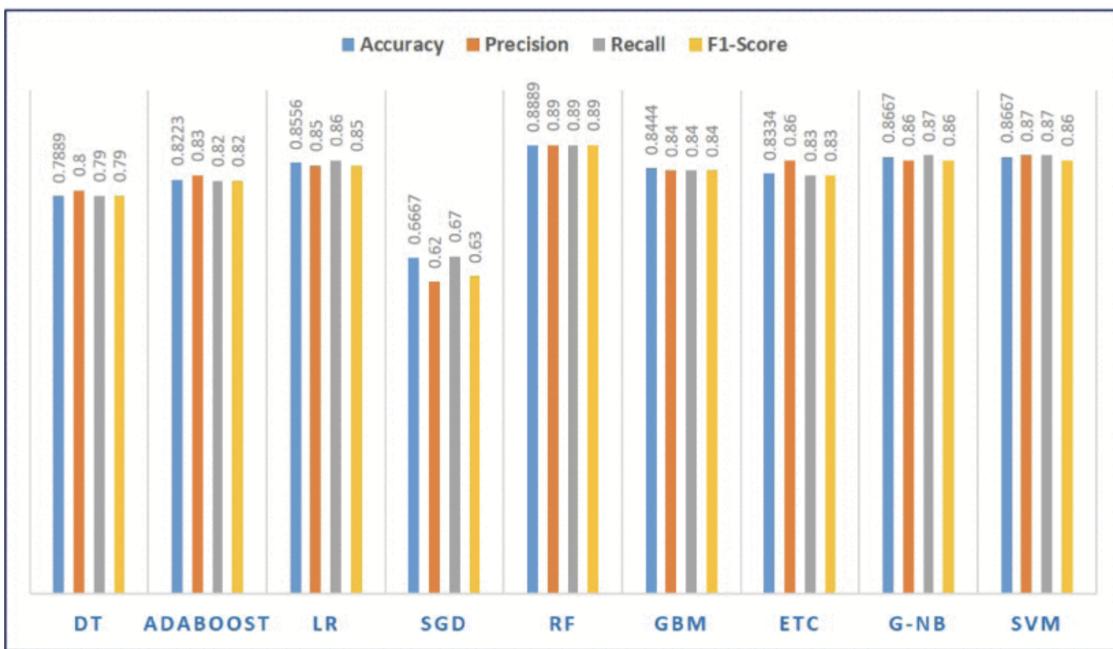


Figure 3.2 Tüm Öznitelikler Kullanılarak Yapılan Sınıflandırmanın Sonuç Grafiği (Orijinal)

3.2 Tüm Öznitelikler ve SMOTE Kullanılarak Yapılan Eğitim

Bu bölümde SMOTE yöntemi kullanılarak tüm özellikler üzerinde deneyel çalışma yapılmıştır. Makalede hiperparametre detaylarına ilişkin bilgi verilmemişinden SMOTE işlemi sonrasında ilk olarak hiperparametreler sabit tutularak sonuç alınmış, sonrasında yeni hiperparametrelerle tekrar eğitim gerçekleştirilmiştir.

3.2.1 Sabit Hiperparametrelerle Eğitim

SMOTE yöntemi ile azınlık sınıfındaki eğitim örnekleri artırılmış ve eğitim veri seti dengeli hale getirilmiştir. Model parametreleri sabit tutularak eğitim gerçekleştirilmiştir ve sonuçlar Tablo 3.4'te verilmiştir.

Table 3.4 SMOTE Yöntemi Kullanılarak Yapılan Eğitimin Sonuçları

Algorithm	Accuracy	Precision	Recall	F1
DT	0.7777	0.6857	0.7272	0.7058
AdaBoost	0.8111	0.8333	0.6060	0.7017
LR	0.8333	0.8461	0.6666	0.7457
SGD	0.6333	0.0	0.0	0.0
RF	0.8222	0.7931	0.6969	0.7419
GBM	0.8444	0.8275	0.7272	0.7741
ETC	0.8111	0.8076	0.6363	0.7118
GNB	0.7444	0.6562	0.6363	0.6461
SVM	0.7666	0.7	0.6363	0.6666

Sınıf dengesizliğini gidermeye yönelik yapılan bu veri artırma işleminin bazı algoritmalarında performansı artırdığı, bazlarında ise sınırlı bir etkisi olduğu gözlemlenmiştir.

Decision Tree (DT) algoritmasında accuracy değeri sabit kalırken, precision ve recall değerlerinde bir değişim gözlenmiştir. Precision %72.41'den %68.57'ye, recall ise %63.63'ten %72.72'ye yükselmiştir. Bu durum, DT'nin SMOTE ile sınıf dengesizliğini daha iyi ele aldığı ancak karar sınırlarının hala daha az etkili olduğunu göstermektedir.

AdaBoost algoritmasında SMOTE sonrası accuracy değeri %82.22'den %81.11'e düşerken, precision değeri aynı seviyede kalmıştır. Recall ise %54.54'ten %60.60'a yükselmiştir. Bu, SMOTE'un AdaBoost'un azınlık sınıfı üzerinde daha iyi bir performans göstermesini sağladığını, ancak genel doğrulukta küçük bir kayıp yaşandığını ortaya koymaktadır.

Logistic Regression (LR) algoritmasında accuracy, precision, ve recall değerlerinde sınırlı bir değişiklik olmuştur. Accuracy %82.22'den %83.33'e yükselmiş, precision %84'ten %84.61'e çıkmıştır. Bu durum, LR'nin SMOTE sonrası veriyi daha iyi deneleyebildiğini göstermektedir. Recall ise %63.63'ten %66.66'ya yükselerek azınlık sınıfı tahminlerinde iyileşme sağlandığını göstermektedir.

Stochastic Gradient Descent (SGD) algoritmasında performans tamamen başarısız hale gelmiştir. Önceki tabloda %36.66 olan accuracy, SMOTE sonrası %0.0 olarak hesaplanmıştır. Precision, recall ve F1 değerleri de %0.0 seviyesinde kalmıştır. Bu, SMOTE'un SGD gibi doğrusal modeller için uygun bir çözüm olmayacağı ve bu algoritmanın sınıf dengesizliğini ele almak için ek düzenlemelere ihtiyaç duyduğunu göstermektedir.

Random Forest (RF) algoritması SMOTE sonrası accuracy değerini düşürmüştür olsa da, recall değerinde bir değişim gözlenmemiştir. F1 skorunda ve precision'da ise minimal değişiklikler gözlemlenmiştir. Bu, RF'nin sınıf dengesizliği karşısında zaten güçlü bir algoritma olduğunu ve SMOTE'un bu performansa sınırlı bir katkı sağladığını göstermektedir.

Gradient Boosting Machine (GBM) algoritması, SMOTE sonrası accuracy değerini düşürmüştür. Ancak F1 skoru önceki çalışmaya kıyasla nispeten korunmuş ve recall değeri %72.72 olarak kaydedilmiştir. Bu, GBM'nin SMOTE ile azınlık sınıfı üzerinde biraz daha dengeli bir performans sağladığını göstermektedir.

Extra Tree Classifier (ETC) algoritması, SMOTE sonrasında accuracy'de bir miktar düşüş göstermiştir. Precision ve recall değerleri de sırasıyla %90.47'den %80.76'ya ve %57.57'den %63.63'e değişmiştir. Bu sonuçlar, SMOTE'un ETC'nin azınlık sınıfı üzerindeki performansını iyileştirdiğini, ancak genel doğrulukta küçük bir kayba yol açtığını göstermektedir.

Gaussian Naive Bayes (GNB) algoritmasında accuracy %75.55'ten %74.44'e düşmüştür. Precision değeri azalırken recall değerinde artış gözlemlenmiş, bu durum SMOTE'un GNB'nin genel performansını pek de iyileştiremediğini göstermiştir.

Son olarak, Support Vector Machine (SVM) algoritmasında SMOTE sonrası accuracy %80'den %76.66'ya düşmüştür, precision ve recall değerleri sırasıyla %89.47'den %70'e ve %51.51'den %63.63'e değişmiştir. Bu, SVM'nin SMOTE sonrası sınıf dengesizliğini bir miktar daha iyi ele aldığına ancak genel doğruluk seviyesinde kayıp yaşadığını göstermektedir.

Orijinal çalışmada, SMOTE kullanılarak veri artırıldıktan sonra oluşan sonuçlar Tablo 3.5'teki gibidir.

Table 3.5 SMOTE Yöntemi Kullanılarak Yapılan Eğitimin Sonuçları (Orijinal)

Models	Accuracy	Precision	Recall	F-Score
DT	0.8278	0.83	0.83	0.83
AdaBoost	0.8852	0.89	0.89	0.89
LR	0.8360	0.84	0.84	0.85
SGD	0.5409	0.54	0.54	0.53
RF	0.9180	0.92	0.92	0.92
GBM	0.8442	0.84	0.84	0.84
ETC	0.9262	0.93	0.93	0.93
GNB	0.7459	0.75	0.75	0.75
SVM	0.7622	0.76	0.76	0.76

SMOTE kullanımı, orijinal makaledeki SMOTE'suz sonuçlarla karşılaştırıldığında, özellikle ağaç tabanlı algoritmalar (RF, GBM, ETC) ve ensemble yöntemleri (AdaBoost) üzerinde performans artışına yol açmıştır. Ancak, doğrusal modeller ve probabilistik yaklaşımlar (SGD ve GNB gibi) üzerinde bu iyileştirmenin sınırlı kaldığı gözlemlenmektedir.

Orijinal çalışmada SMOTE uygulanmadan önce, Random Forest (RF) %88.89 accuracy ile en yüksek performansı sergilemiştir. SMOTE sonrasında RF'nin accuracy değeri %91.80'e yükselmiştir. Benzer şekilde, Extra Trees Classifier (ETC) %83.34'ten %92.62'ye önemli bir artış göstermiştir. Bu durum, ağaç tabanlı modellerin SMOTE ile dengelenmiş veri setlerinde daha etkili olduğunu ve bu yöntemin azınlık sınıfına ait verilerin temsil edilme oranını artırarak genel doğruluğu iyileştirdiğini göstermektedir. Logistic Regression (LR) ve Gradient Boosting Machine (GBM) gibi yöntemler de benzer şekilde, daha dengeli performans sergilemiştir. AdaBoost'ta ise hem doğruluk hem de F1 skoru, SMOTE sonrası %88.52 gibi yüksek değerlere ulaşmıştır.

Öte yandan, Stochastic Gradient Descent (SGD) algoritmasında SMOTE sonrasında da düşük bir performans gözlemlenmiştir. Bu sonuç, SGD'nin sınıf dengesizliği giderildiğinde bile azınlık sınıfını etkili bir şekilde tahmin edemediğini ve doğrusal modellerin SMOTE ile veri artırımadan yeterince fayda sağlamayabileceğini ortaya koymaktadır. Gaussian Naive Bayes (GNB) algoritmasında da SMOTE sonrası accuracy %74.59'ye düşüş göstermiştir.

Bu çalışmada ise SMOTE kullanımı farklı algoritmalar için hem benzer hem de farklı etkiler göstermiştir. Örneğin, RF algoritması %86.66'dan %82.22'ye düşerek orijinal

çalışmanın aksine doğrulukta bir kayıp yaşamıştır. AdaBoost da ise %82.22'den %81.11'e düşüş, bu çalışmada da sınırlı bir etkisinin olduğunu göstermektedir. ETC, orijinal çalışmada gibi iyi bir performans göstermiş ancak bu sonuçlarda artış sınırlı kalmıştır. SGD, bu çalışmada SMOTE sonrası tamamen başarısız olmuş, bu durum orijinal makalede gözlemlenen sınırlı başarıyla uyum göstermektedir.

Genel olarak, orijinal makaledeki SMOTE kullanımı, özellikle ağaç tabanlı ve ensemble yöntemlerin azınlık sınıfına olan duyarlığını artırmış ve daha yüksek doğruluk sağlamıştır. Bu çalışmada ise SMOTE, doğruluk açısından bazı algoritmalarla kayıplara neden olmuş, ancak recall ve F1 gibi metriklerde sınırlı da olsa iyileşmeler sağlamıştır. Bu farkların temelinde, hiperparametre ayarları ve SMOTE'un uygulanma biçimindeki farklılıklar yer almaktadır. Orijinal çalışma daha genel bir iyileşme sağlarken, bu çalışma bazı algoritmalarla dengesiz sonuçlar üretmiştir. Bu durum, her modelin SMOTE gibi bir yönteme farklı tepki verdiği ve bu yöntemin etkisinin modelin yapısına göre değiştiğini açıkça ortaya koymaktadır.

Sırasıyla Şekil 3.3'te ve Şekil 3.4'te bu çalışmada ve orijinal çalışmada SMOTE uygulandıktan sonraki sonuçlara ait grafikler verilmiştir.

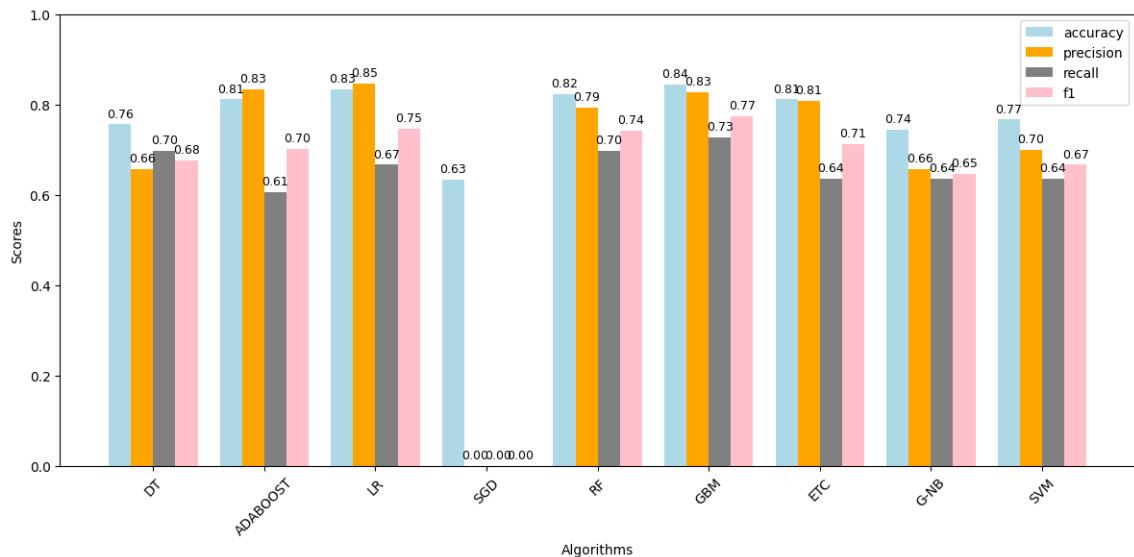


Figure 3.3 SMOTE Yöntemi Kullananlarak Yapılan Eğitimin Sonuç Grafiği

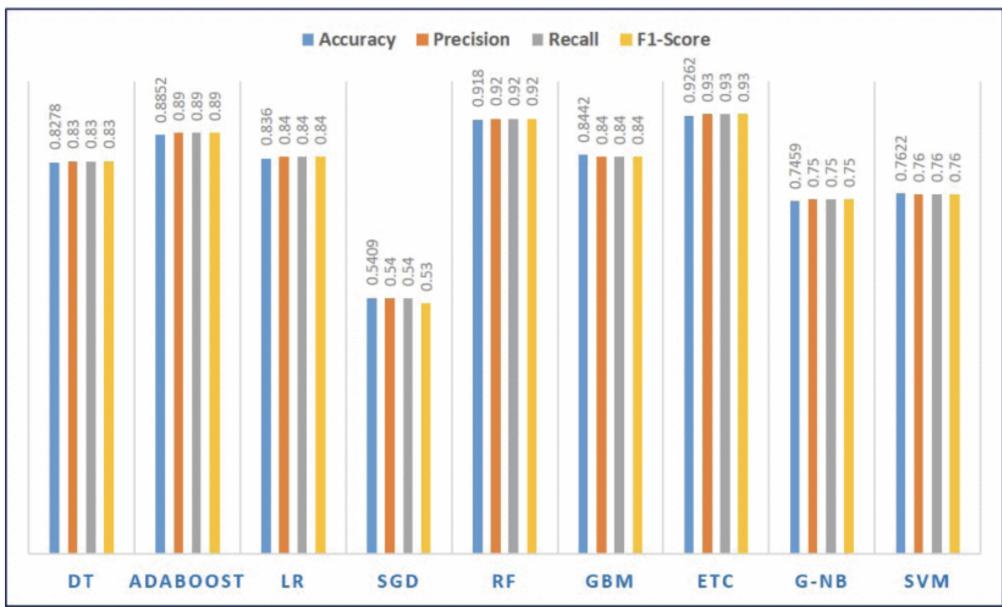


Figure 3.4 SMOTE Yöntemi Kullanılarak Yapılan Eğitimin Sonuç Grafiği (Orijinal)

3.2.2 Farklı Hiperparametrelerle Eğitim

SMOTE işlemininin uygulanmasının ardından modeldeki hiperparametreler güncellenmiştir. Bu şekilde eğitimler yeniden gerçekleştirilmiş ve sonuçları aşağıda görülmektedir.

Table 3.6 SMOTE Yöntemi Kullanılarak Farklı Parametrelerle Yapılan Eğitimde Kullanılan Hiperparametreler

Algorithm	Parameters
DT	{'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 8, 'min_samples_split': 2}
AdaBoost	{'algorithm': 'SAMME.R', 'learning_rate': 1.0, 'n_estimators': 100}
LR	{'C': 0.1, 'max_iter': 50, 'penalty': 'l2', 'solver': 'liblinear', 'tol': 1e-05}
SGD	{'alpha': 0.0001, 'class_weight': None, 'learning_rate': 'optimal', 'loss': 'squared_hinge', 'max_iter': 1000, 'penalty': 'elasticnet'}
RF	{'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 100}
GBM	{'learning_rate': 0.1, 'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50}
ETC	{'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 150}
GNB	-
SVM	{'C': 0.1, 'kernel': 'linear'}

Table 3.7 SMOTE Yöntemi Kullanılarak Farklı Hiperparametrelerle Yapılan Eğitimin Sonucu

Algorithm	Accuracy	Precision	Recall	F1
DT	0.8111	0.8333	0.6060	0.7017
AdaBoost	0.8444	0.8518	0.6969	0.7666
LR	0.8333	0.8461	0.6666	0.7457
SGD	0.6333	0.0	0.0	0.0
RF	0.8555	0.8333	0.7575	0.7936
GBM	0.8555	0.8571	0.7272	0.7868
ETC	0.8111	0.8076	0.6363	0.7118
GNB	-	-	-	-
SVM	0.7666	0.7	0.6363	0.6666

SMOTE işlemi sonrası modellerin hiperparametreleri optimize edilerek yeniden eğitilmesi, performans metriklerinde genel bir iyileşmeye yol açmıştır. Bu iyileşmeler, SMOTE ile dengelenmiş veri setinin, doğru hiperparametreler seçildiğinde, özellikle ensemble ve ağaç tabanlı modellerin performansını artırabildiğini göstermektedir. Ancak, bu durum tüm algoritmalar için tutarlı olmamış ve bazı modellerde sınırlı etkiler gözlemlenmiştir.

En belirgin iyileşme, ensemble ve ağaç tabanlı modellerde görülmüştür. Random Forest (RF) ve Gradient Boosting Machine (GBM), SMOTE sonrası optimize edilmiş parametrelerle önceki sonuçlarına kıyasla daha yüksek recall ve F1 skorlarına ulaşmıştır. RF'nin recall değeri %69.69'dan %75.75'e, F1 skoru ise %74.19'dan %79.36'ya yükselmiştir. GBM için de benzer bir artış gözlemlenmiş, recall%72.72 seviyesinde sabit kalmış ancak F1 skoru iyileşerek %78.68 olmuştur. Bu, ağaç tabanlı algoritmaların, sınıf dengesizliğini gidermeye yönelik SMOTE işlemiyle daha verimli çalıştığını ve hiperparametre optimizasyonunun etkili olduğunu göstermektedir.

AdaBoost algoritmasında ise dikkat çekici bir performans artışı gözlenmiştir. Recall değeri %60.60'tan %69.69'a yükselmiş ve F1 skoru %70.17'den %76.66'ya çıkmıştır. Bu durum, AdaBoost'un SMOTE ile dengelenmiş veri setinde daha etkili bir şekilde öğrenim gerçekleştirdiğini ve optimize edilen parametrelerin model performansına olumlu katkı sağladığını göstermektedir.

Diğer yandan, Logistic Regression (LR) algoritması, SMOTE sonrası optimize edilmiş parametrelerle nispeten dengeli bir performans göstermiştir. Ancak, doğrusal modellerden Stochastic Gradient Descent (SGD) algoritması, optimize edilen parametrelere rağmen önceki başarısızlığını sürdürmektedir. SMOTE sonrası recall

ve precision değerlerinin sıfır kalması, bu algoritmanın dengesiz veri setlerinde ve SMOTE sonrası dengelenmiş veri setinde bile etkili bir öğrenim gerçekleştiremediğini göstermektedir.

SVM algoritması için optimize edilmiş parametreler ile recall değeri %63.63 seviyesinde sabit kalırken, F1 skoru da %66.66'dan yukarıya çıkamamıştır.

Bu değerlere ait grafik Şekil 3.5'te verilmiştir.

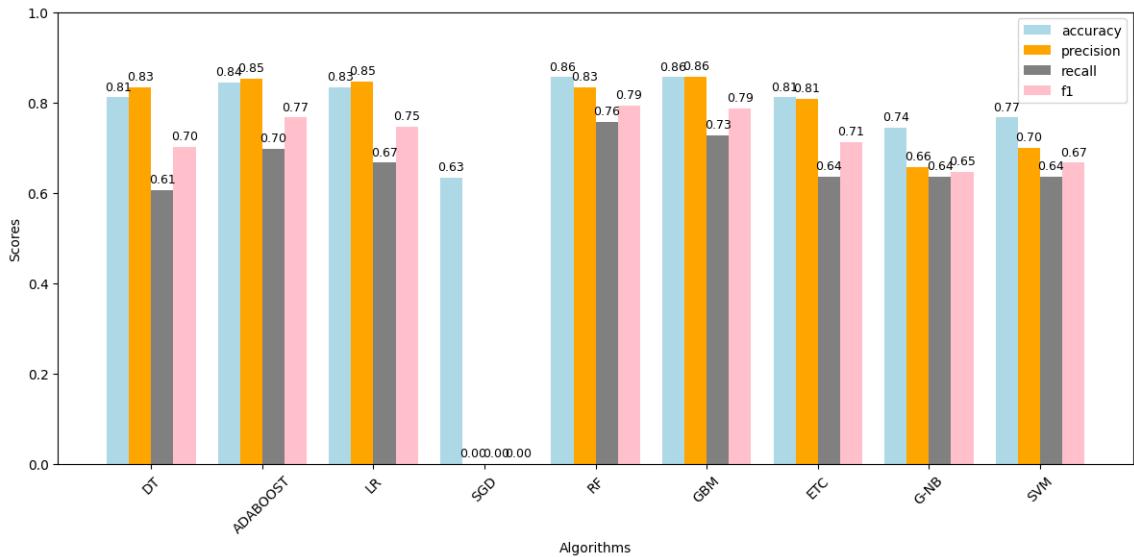


Figure 3.5 SMOTE Yöntemi Kullanılarak Farklı Hiperparametrelerle Yapılan Eğitimin Sonuç Grafiği

3.3 Önemli Öznitelikler Kullanılarak Yapılan Eğitim

Bölüm 2'de belirlenen öznitelik önem sırasındaki son 4 öznitelik (Smoking, Anemia, Gender, BP) çıkarılmış ve bu şekilde eğitimler yeniden gerçekleştirilmiştir. Eğitim aşamasında kullanılan hiperparametreler Tablo 3.8'te verilmiştir.

Table 3.8 Önemli Öznitelikler Kullanılarak Eğitilen Modellerin Hiperparametreleri

Algorithm	Parameters
DT	{'criterion': 'entropy', 'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 10}
AdaBoost	{'algorithm': 'SAMME.R', 'learning_rate': 1.0, 'n_estimators': 100}
LR	{'C': 0.1, 'max_iter': 1000, 'penalty': 'l2', 'solver': 'liblinear', 'tol': 0.0001}
SGD	{'alpha': 0.0001, 'class_weight': None, 'learning_rate': 'optimal', 'loss': 'squared_hinge', 'max_iter': 1000, 'penalty': 'elasticnet'}
RF	{'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 150}
GBM	{'learning_rate': 0.1, 'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 100}
ETC	{'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 150}
GNB	-
SVM	{'C': 0.1, 'kernel': 'linear'}

Eğitim sonrası oluşan sonuçlar Tablo 3.9'de verilmiştir.

Table 3.9 Önemli Öznitelikler Kullanarak Eğitilen Modellerin Sonucu

Algorithm	Accuracy	Precision	Recall	F1
DT	0.7666	0.6666	0.7272	0.6956
AdaBoost	0.8444	0.8518	0.6969	0.7666
LR	0.8222	0.7931	0.6969	0.7419
SGD	0.6333	0.0	0.0	0.0
RF	0.8333	0.8	0.7272	0.7619
GBM	0.8444	0.8064	0.7575	0.7812
ETC	0.8555	0.8571	0.7272	0.7868
GNB	0.7444	0.6562	0.6363	0.6461
SVM	0.7666	0.6875	0.6666	0.6769

Son dört özelliğin veri setinden çıkarılması ve SMOTE kullanılarak modellerin belirlenen parametrelerle yeniden eğitilmesi, önceki SMOTE ve grid search sonrası eğitim sonuçlarına kıyasla performansta hem olumlu hem de olumsuz yönde etkiler yaratmıştır. Özellikle, veri setindeki daha az önemli olduğu düşünülen özniteliklerin çıkarılması, modellerin genelleme kapasitesine etki etmiş ve bazı algoritmaların performansında iyileşme sağlarken diğerlerinde performans kayıplarına yol açmıştır.

Tablo 3.4 ile karşılaştırıldığında, RF'nin accuracy değeri %82.22'den %83.33'e hafif bir artış göstermiştir ve precision değeri %79.31'den %80'e yükselmiştir. GBM de benzer şekilde, accuracy değerlerini korumuş ya da hafif iyileştirmeler göstermiştir. ETC için accuracy değeri kayda değer bir yükseliş göstermemiştir. Bunlar, ağaç tabanlı algoritmaların daha az önemli öznitelikler çıkarıldığından performansını koruyabildiğini, hatta SMOTE ile dengelenmiş veri setinde daha etkili hale geldiğini göstermektedir.

Logistic Regression (LR) gibi doğrusal modellerde ise bazı metriklerde küçük kayıplar gözlemlenmiştir. Örneğin, LR'nin accuracy değeri %83.33'ten %82.22'ye düşmüştür, recall %66.66 seviyesinde sabit kalmış ancak F1 skoru bir miktar gerilemiştir. Bu, doğrusal modellerin tüm özniteliklerden faydalanan eğiliminde olduğunu ve bazı özniteliklerin çıkarılmasının modeli sınırlayabileceğini göstermektedir.

Stochastic Gradient Descent (SGD) algoritması, özniteliklerin çıkarılması sonrasında yine düşük performans sergilemiştir. Recall ve precision değerleri sıfırda sabit kalmış, bu da algoritmanın veri setindeki değişikliklerden olumlu etkilenmediğini ortaya koymaktadır. Bu sonuç, SGD'nin sınıf dengesizliği ve öznitelik seçimi gibi durumlara karşı duyarlığını bir kez daha vurgulamaktadır.

Gaussian Naive Bayes (GNB) ve Support Vector Machine (SVM) algoritmaları ise öznitelik çıkarımı sonrası performanslarında belirgin bir değişim göstermemiştir. GNB'nin accuracy değeri %74.44'te sabit kalırken, SVM'nin doğruluğu %76.66'da korunmuştur. Bu durum, bu algoritmaların özniteliklerin çıkarılmasından sınırlı şekilde etkilendiğini ve bu özelliklerin modelin genelleme kapasitesine çok fazla katkı sağlamadığını düşündürmektedir. Ensemble yöntemlerden AdaBoost'un performansı ise öznitelik çıkarımından sonra gözle görülür bir artış göstermiştir.

Aşağıda ise orijinal çalışmada öznitelik çıkarımı sonrası sınıflandırıcıların performansları verilmiştir.

Table 3.10 Önemli Öznitelikler Kullanarak Eğitilen Modellerin Sonucu (Orijinal)

Models	Accuracy
DT	0.8778
AdaBoost	0.8852
LR	0.8442
SGD	0.5491
RF	0.9188
GBM	0.8852
ETC	0.9262
GNB	0.7540
SVM	0.7622

Orijinal çalışmada önemsiz özniteliklerin çıkarılması ve SMOTE uygulanması sonrası elde edilen sonuçlar, özellikle ağaç tabanlı ve ensemble modellerde (RF, GBM, ETC) daha yüksek performans sergilediğini göstermektedir. RF %91.88 ve ETC %92.62 accuracy ile öne çıkarken, bu çalışmada RF %83.33 ve ETC %85.55 ile daha düşük performans göstermiştir. Bu farklar, orijinal çalışmada daha etkili hiperparametre optimizasyonu uygulanmış olmasından kaynaklanabilir.

AdaBoost ve GBM gibi ensemble yöntemler her iki çalışmada da güçlü performans sergilemiştir; ancak bu çalışmada accuracy değerleri (%84.44) orijinal çalışmaya (%88.52) göre bir miktar gerilemiştir. Logistic Regression (LR) algoritması da benzer şekilde %84.42'den %82.22'ye düşmüştür, bu da doğrusal modellerin SMOTE ve öznitelik seçimi sırasında daha hassas olduğunu göstermektedir. SVM ve GNB gibi modellerde ise performans farkı minimal kalmıştır, bu durum bu modellerin öznitelik çıkarımından daha az etkilendiğini ortaya koymaktadır.

Stochastic Gradient Descent (SGD) algoritması, bu çalışmada başarısız olurken, orijinal çalışmada %54.91 accuracy ile sınırlı bir başarı göstermiştir. Bu durum, SGD'nin SMOTE sonrası dengelenmiş veri setlerinde dahi performans sorunları yaşayabileceğini ortaya koymaktadır. Genel olarak, orijinal çalışmada hiperparametre optimizasyonu ve veri işleme adımlarının daha etkili kullanıldığı ve bu nedenle daha yüksek doğruluk elde edildiği söylenebilir.

Tablo 3.9 ve Tablo 3.10'a ait grafikler sırasıyla aşağıda verilmiştir.

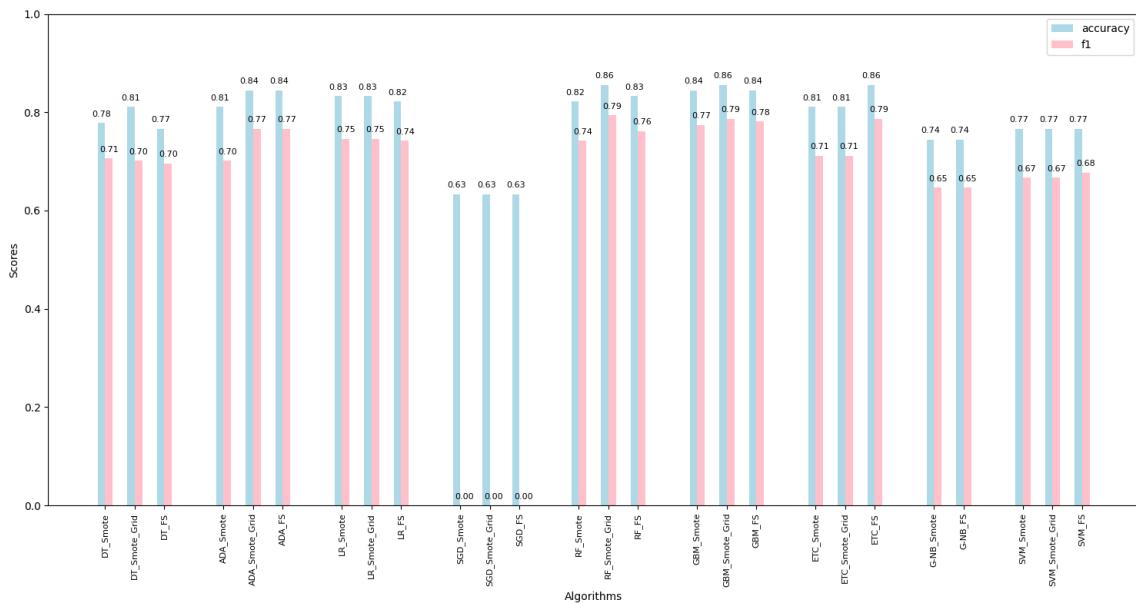


Figure 3.6 Önemli Öznitelikler Kullanarak Eğitilen Modellerin Sonuç Grafiği

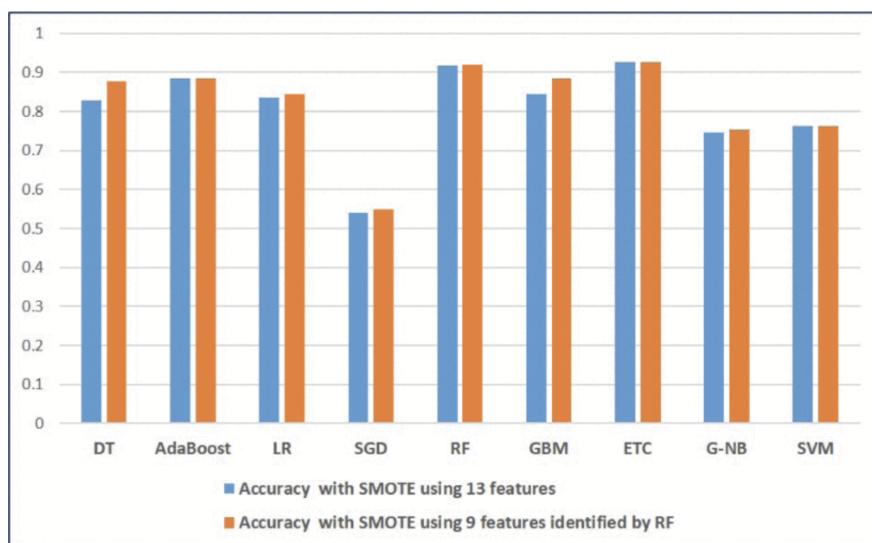


Figure 3.7 Önemli Öznitelikler Kullanarak Eğitilen Modellerin Sonuç Grafiği (Orijinal)

4 **SONUÇ**

Bu çalışmada, Improving the Prediction of Heart Failure Patients' Survival Using SMOTE and Effective Data Mining Techniques makalesinin detaylı bir raporu sunulmuştur. Çalışmada LR, AdaBoost, RF, GBM, G-NB ve SVM gibi makine öğrenimi teknikleri kullanılmış, sınıf dengesizliği problemini çözmek için SMOTE yöntemi uygulanmıştır. Ayrıca, RF algoritması kullanılarak öznitelik sıralaması gerçekleştirilmiştir. RF'ye göre en önemli öznitelikler: Time, Creatinine, Ejection Fraction, Age, Platelets, CPK ve Sodium olarak belirlenmiştir.

Deneysel sonuçlar, öznitelik seçimi ile birlikte kullanılan ağaç tabanlı yöntemlerin en yüksek doğruluk değerlerini elde ettiğini göstermiştir. SMOTE tekniği, özellikle ağaç tabanlı sınıflandırıcıların kalp hastalarının sağkalımını tahmin etme performansını önemli ölçüde artırmıştır. Ancak, SMOTE'un tüm algoritmalar için her zaman uygun bir çözüm olmadığı gözlemlenmiştir; bazı algoritmalarla performans artışı sağlanırken, bazlarında ise sınırlı etki veya performans kayipları gözlenmiştir. Benzer şekilde, öznitelik seçimi de kullanılan algoritmayla bağlı olarak farklı sonuçlar üretmekte, bazı modeller seçilen özelliklerden olumlu etkilenirken, diğerleri bu değişikliklerden daha az fayda sağlayabilmektedir.

Referanslar

- [1] A. Ishaq *et al.*, “Improving the prediction of heart failure patients’ survival using smote and effective data mining techniques,” *IEEE Access*, vol. 9, pp. 39 707–39 716, 2021. DOI: 10.1109/ACCESS.2021.3064084.
- [2] *The top 10 causes of death — who.int*, <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>, [Accessed 02-12-2024].
- [3] *UCI Machine Learning Repository — archive.ics.uci.edu*, <https://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records>, [Accessed 29-11-2024].
- [4] *Decision Tree - GeeksforGeeks — geeksforgeeks.org*, <https://www.geeksforgeeks.org/decision-tree/>, [Accessed 02-12-2024].
- [5] mljourney, *Understanding the AdaBoost Algorithm in Machine Learning - ML Journey — mljourney.com*, https://mljourney.com/understanding-the-adaboost-algorithm-in-machine-learning/?utm_source=chatgpt.com, [Accessed 08-12-2024].
- [6] E. R. Muratlar, *Gradient Boosted Regresyon Ağacı - Veri Bilimi Okulu — veribilimiokulu.com*, <https://www.veribilimiokulu.com/gradient-boosted-regresyon-agaci/>, [Accessed 08-12-2024].
- [7] A. Sharaff and H. Gupta, “Extra-tree classifier with metaheuristics approach for email classification,” in *Advances in Computer Communication and Computational Sciences*, S. K. Bhatia, S. Tiwari, K. K. Mishra, and M. C. Trivedi, Eds., Singapore: Springer Singapore, 2019, pp. 189–197, ISBN: 978-981-13-6861-5.