04.06.2025

Bursa Teknik Üniversitesi – Bilgisayar Mühendisliği

BLM463 Veri Madenciliğine Giriş dersi Proje Ödevi

21360859054 – İbrahim Semih Temiz

Shill Bidding Tespiti: Veri Madenciliği Yaklaşımıyla Sınıflandırma Analizi

1. Giriş

Çevrim içi açık artırma sistemlerinde karşılaşılan en önemli güvenlik sorunlarından biri olan sahte teklif (shill bidding) problemi, bu projenin temel odak noktasını oluşturmaktadır. Sahte teklif, bir açık artırmada satıcının kendisi veya işbirliği yaptığı kişiler tarafından yapılan yapay teklifler olarak tanımlanabilir. Bu durum, hem alıcıların yüksek fiyatlar ödemesine hem de açık artırma sisteminin güvenilirliğinin zedelenmesine neden olmaktadır.

Bu projede, Shill Bidding Dataset üzerinde kapsamlı bir veri madenciliği çalışması gerçekleştirilmiştir. Çalışma, veri ön işleme adımlarından başlayarak, detaylı veri analizi ve görselleştirme ile devam etmiş, makine öğrenmesi modellerinin uygulanması ve performans değerlendirmesi ile sonuçlanmıştır. Özellikle Random Forest, SVM ve KNN gibi farklı sınıflandırma algoritmaları kullanılarak, sahte teklif veren kullanıcıların tespiti için etkili bir model geliştirilmiştir.

Projenin temel hedefleri:

- Sahte teklif veren kullanıcıların tespiti için etkili bir sınıflandırma modeli geliştirmek
- Farklı makine öğrenmesi algoritmalarının performanslarını karşılaştırmak
- Hiperparametre optimizasyonu ile model performansını artırmak
- Veri madenciliği sürecinin tüm aşamalarını detaylı olarak belgelemek

2. Veri Hazırlığı ve Ön İşleme

2.1 Kütüphaneler ve Araçlar

Projenin başarılı bir şekilde gerçekleştirilmesi için gerekli olan kütüphaneler ve araçlar seçilirken, veri analizi ve makine öğrenmesi süreçlerinin her aşaması göz önünde bulundurulmuştur:

- **pandas:** Veri manipülasyonu ve analizi için kullanılan temel kütüphane
- **numpy:** Sayısal işlemler ve matris işlemleri için
- **matplotlib ve seaborn:** Veri görselleştirme ve analiz için
- **scikit-learn:** Makine öğrenmesi modelleri ve metrikler için
- **imbalanced-learn:** Sınıf dengesizliği yönetimi için

2.2 Veri Seti ve İlk İnceleme

Çalışmada kullanılan veri seti, çevrim içi açık artırma sistemlerinden toplanan gerçek kullanıcı davranışlarını içermektedir. Veri setinin temel özellikleri:

- **Kaynak:** shill_bidding_dataset.csv
- **Gözlem sayısı:** 6321 (toplam kullanıcı sayısı)
- **Değişken sayısı:** 38 (37 özellik + 1 hedef değişken)
- **Hedef değişken:** Class (0: Normal kullanıcı, 1: Sahte teklif veren)

2.3 Veri Temizliği ve Dönüşüm

Veri setinin kalitesini artırmak ve modellerin daha iyi performans göstermesini sağlamak için kapsamlı bir veri temizliği süreci uygulanmıştır:

Eksik Değer Analizi:

Veri setinde eksik değer bulunmaması, analiz sürecini olumlu yönde etkilemiştir. Bu durum, veri setinin kaliteli olduğunu ve ek bir doldurma işlemine gerek olmadığını göstermektedir.

Aykırı Değer Tespiti ve İşleme:

Boxplot analizi kullanılarak aykırı değerler tespit edilmiş ve Winsorize yöntemi ile işlenmiştir. Bu yöntemde, değerler %1 ve %99 yüzdelik dilimler kullanılarak sınırlandırılmıştır.

Kategorik Değişken Dönüşümü:

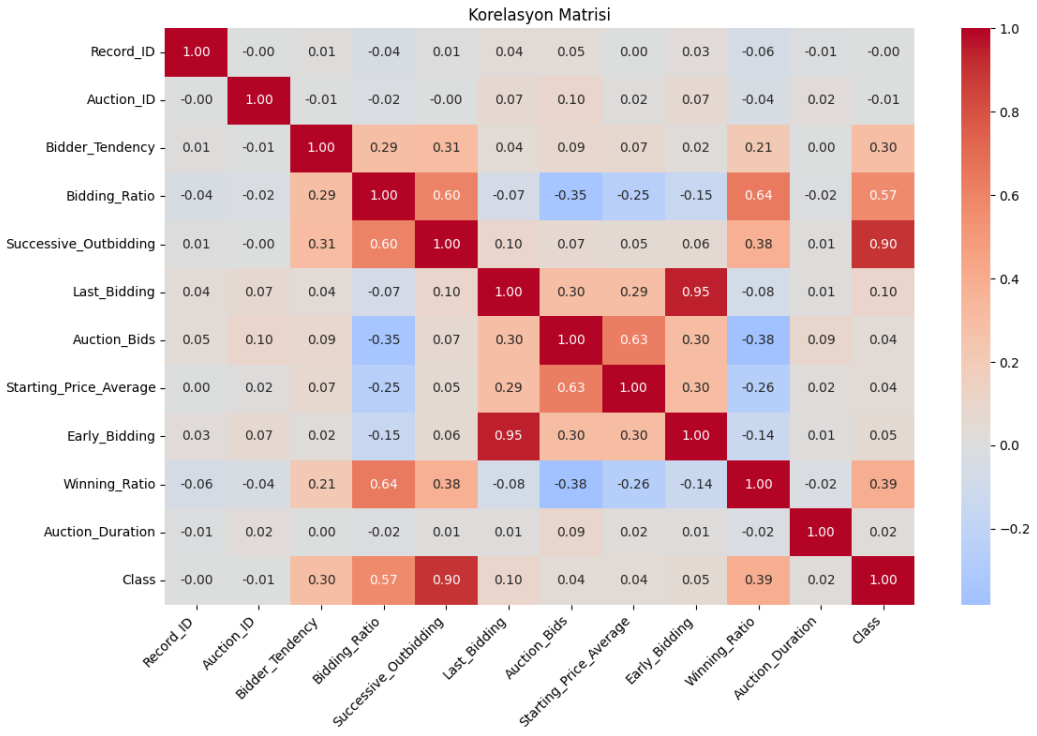
Bidder_ID ve Auction_ID gibi kategorik değişkenler, LabelEncoder kullanılarak sayısal değerlere dönüştürülmüştür.

Ölçeklendirme:

StandardScaler kullanılarak tüm sayısal özellikler z-skora dönüştürülmüştür. Bu işlem, özellikle SVM ve KNN gibi ölçek duyarlı modellerin performansını artırmıştır.

3. Veri Görselleştirme ve Analiz

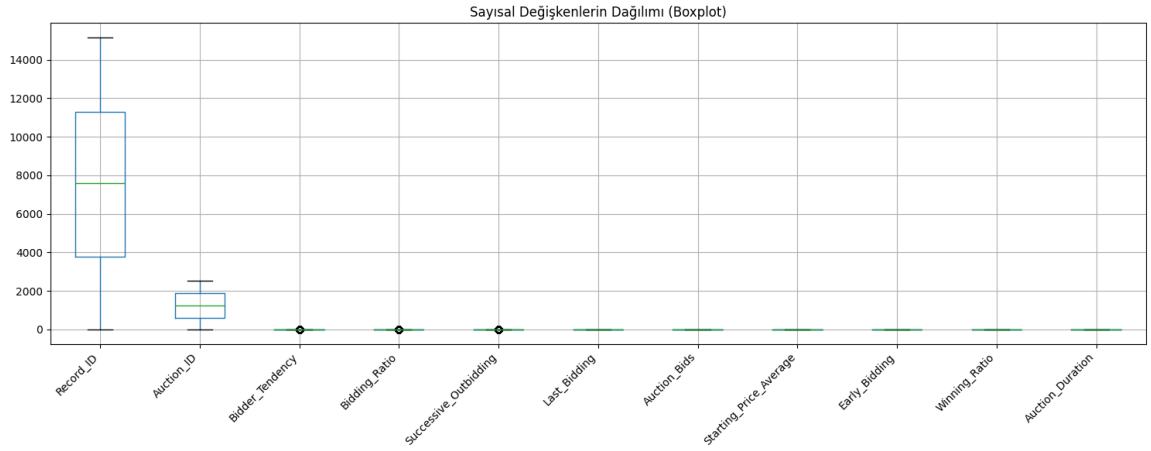
3.1]



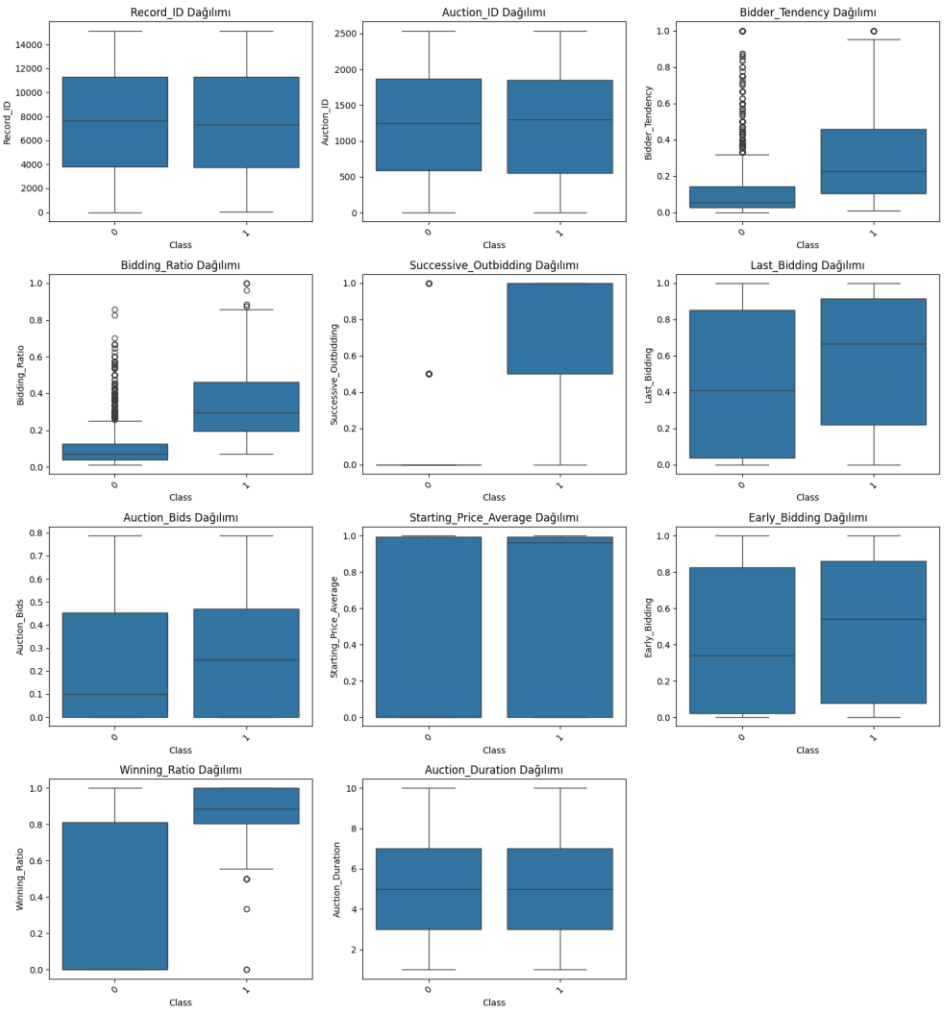
Korelasyon Matrisi Heatmap

Korelasyon matrisi, değişkenler arasındaki ilişkileri anlamak için güçlü bir araçtır. Heatmap görselleştirmesi, korelasyon değerlerini -1 ile 1 arasında renk kodlaması ile göstermiştir. 0.95 üzerindeki korelasyona sahip öznitelikler tespit edilmiş ve bu öznitelikler veri setinden çıkarılmıştır.

3.2 Dağılım Analizi



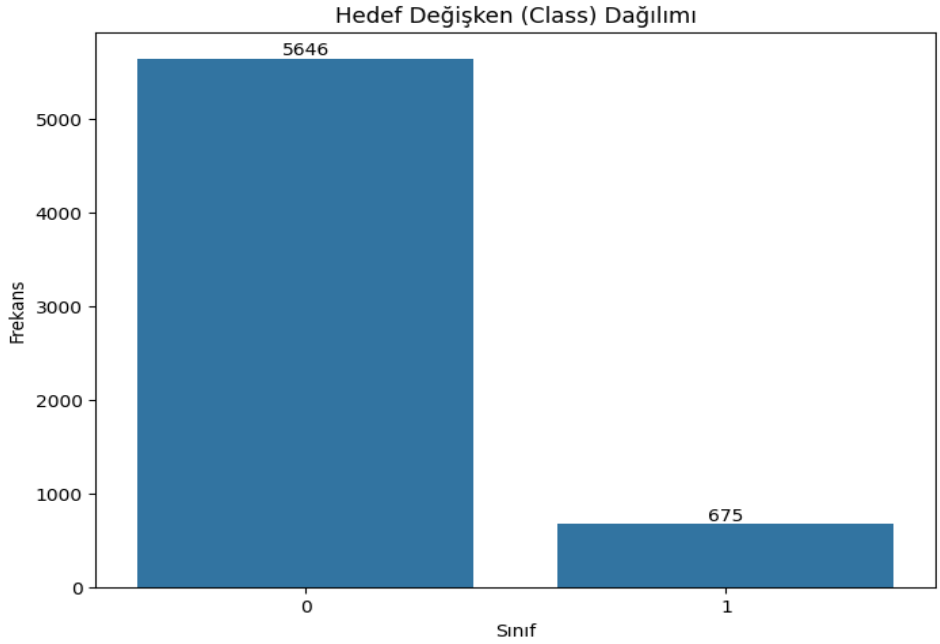
Sayısal Değişkenlerin Boxplot Analizi



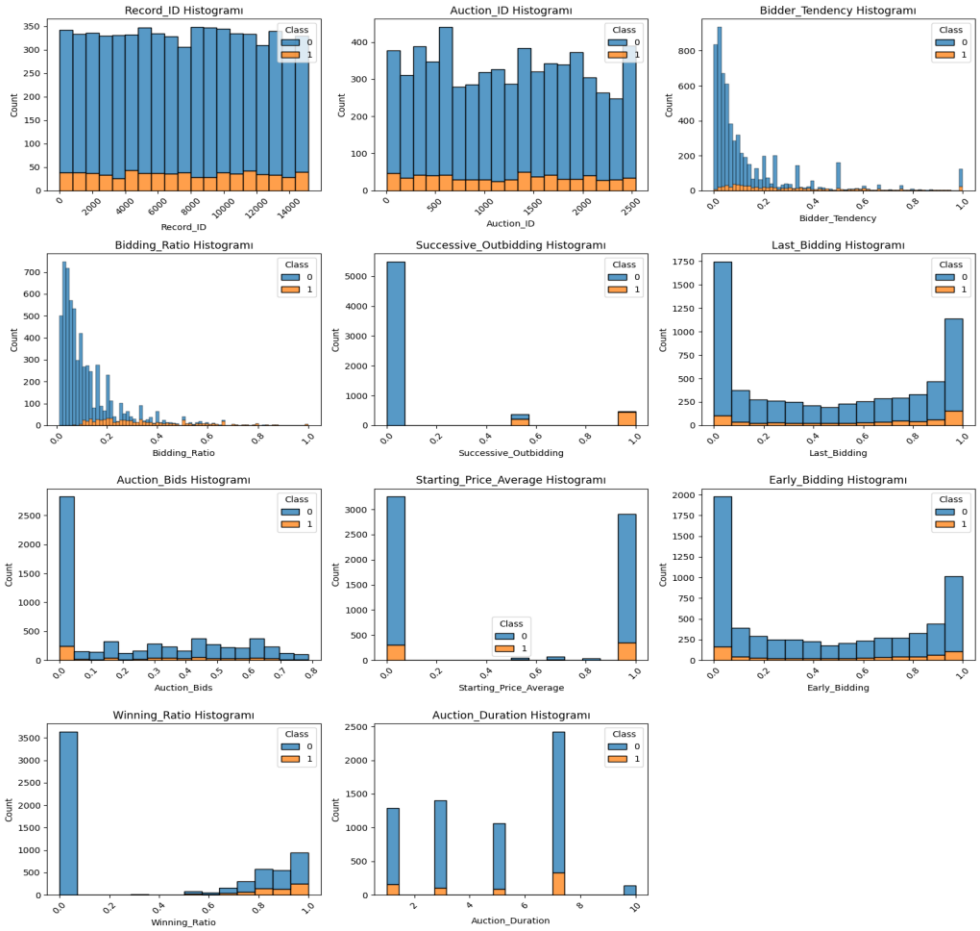
Sınıf Bazlı Boxplot Analizi

Boxplot grafikleri, sayısal değişkenlerin dağılımını ve aykırı değerleri görselleştirmek için kullanılmıştır. Her değişken için medyan, çeyrek değerler ve aykırı gözlemler analiz edilmiştir.

3.3 Sınıf Bazlı Analiz



Sınıf Dağılımı Çubuk Grafiği



Sınıf Bazlı Histogramlar

Sınıf dağılımı analizi, veri setinde ciddi bir dengesizlik olduğunu göstermiştir. Sahte kullanıcılar (Class=1) azınlık sınıfı oluşturmaktadır. Bu durum, SMOTE uygulamasının gerekliliğini ortaya koymuştur.

4. Sınıf Dengesizliği ve SMOTE

4.1 SMOTE Uygulaması

Sınıf dengesizliğini gidermek için SMOTE (Synthetic Minority Oversampling Technique) algoritması uygulanmıştır:

- Yalnızca eğitim seti üzerinde SMOTE uygulanmıştır
- Azınlık sınıf örnekleri sentetik olarak çoğaltılmıştır
- Test seti orijinal dağılımını korumuştur
- random_state=42 ile tekrarlanabilirlik sağlanmıştır

5. Modelleme ve Performans Analizi

5.1 Kullanılan Modeller

5.1.1 Random Forest Classifier (RF)

Random Forest, karar ağaçları topluluğuna dayalı güçlü bir ansambl modeldir. Bu model, overfitting kontrolü için max_depth parametresi kullanılarak eğitilmiştir. Özellik önem dereceleri hesaplanmış ve modelin karar verme sürecini anlamak için kullanılmıştır.

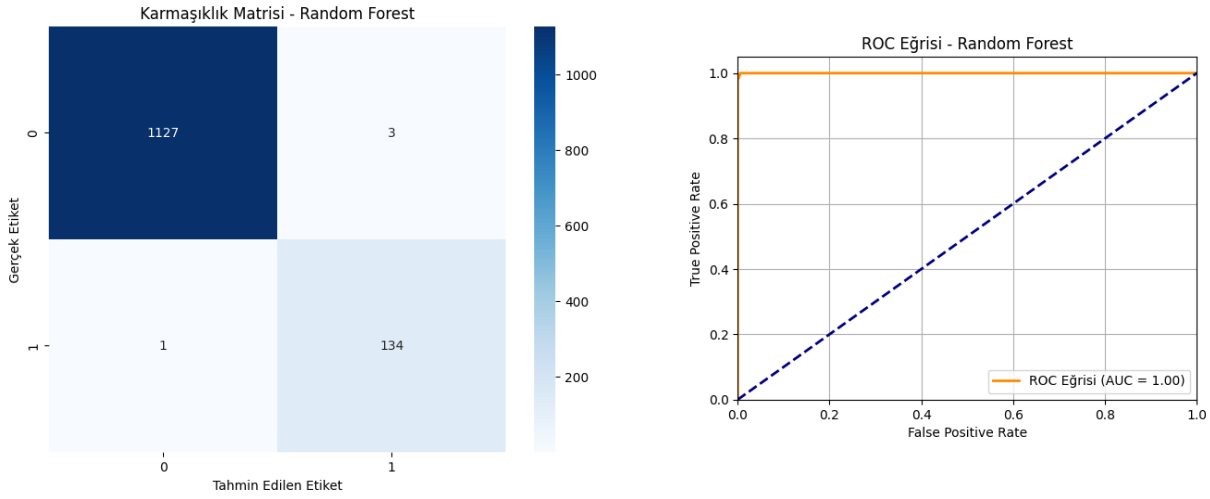
5.1.2 Support Vector Machine (SVM)

SVM, doğrusal ve doğrusal olmayan ayırımlar için etkili bir sınıflandırma algoritmasıdır. RBF kernel kullanılmış ve probability=True parametresi ile olasılık tahminleri etkinleştirilmiştir.

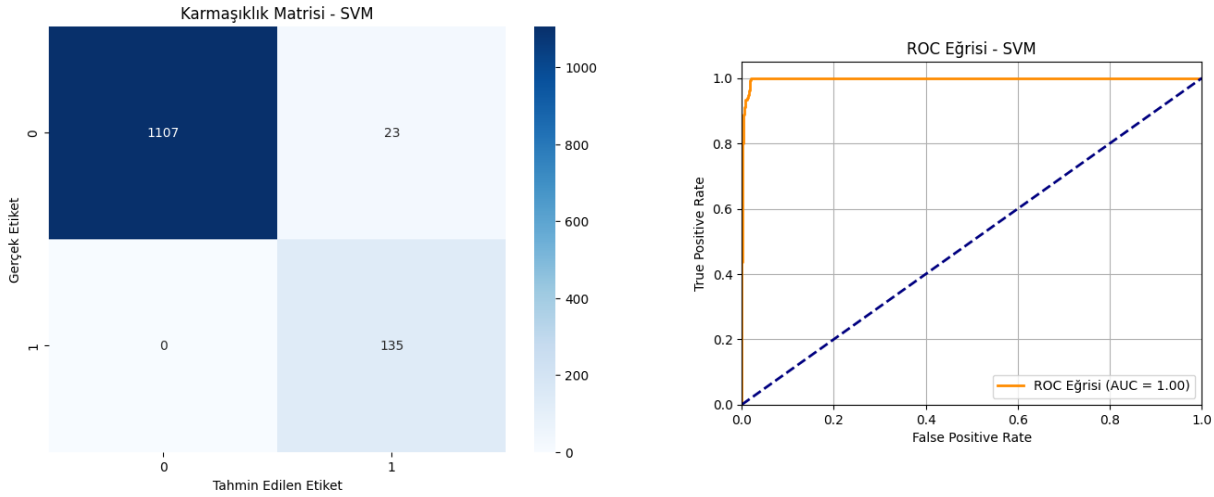
5.1.3 K-Nearest Neighbors (KNN)

KNN, komşuluk temelli basit ama etkili bir yöntemdir. Varsayılan k=5 parametresi ile başlangıç modeli oluşturulmuş ve Euclidean distance metriği kullanılmıştır.

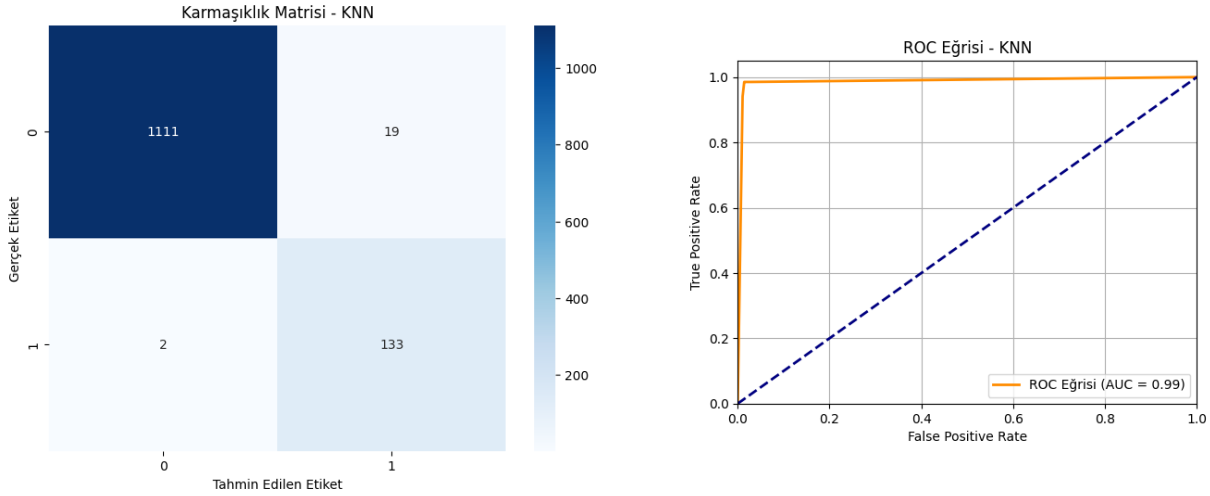
5.2 Model Performans Değerlendirmesi



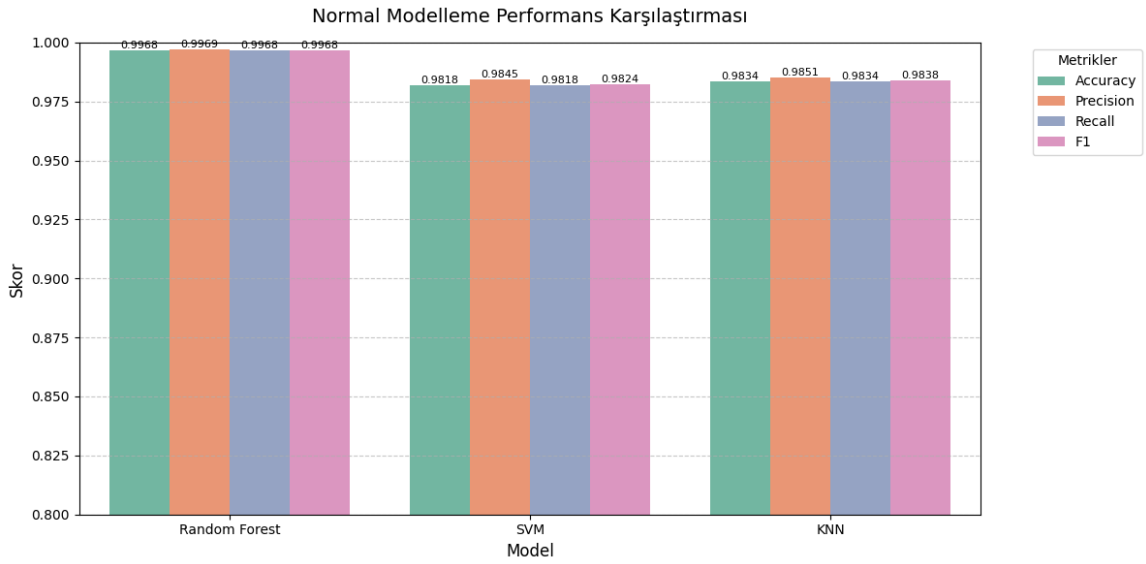
Random Forest – Karmaşıklık Matrisi | Roc Eğrisi



SVM – Karmaşıklık Matrisi | Roc Eğrisi



KNN – Karmaşıklık Matrisi | Roc Eğrisi



Modellerin Performans Metriklerinin Karşılaştırılması

Değerlendirme Metrikleri:

- Doğruluk (Accuracy): Genel sınıflandırma başarısı
- Kesinlik (Precision): Pozitif tahminlerin doğruluk oranı
- Duyarlılık (Recall): Gerçek pozitiflerin tespit oranı
- F1-Skoru: Precision ve Recall'un harmonik ortalaması
- ROC Eğrisi ve AUC Değeri: Model ayrıştırma gücü

6. Hiperparametre Optimizasyonu

6.1 Optimizasyon Süreci

GridSearchCV kullanılarak her model için en iyi parametre kombinasyonları belirlenmiştir:

Random Forest:

- n_estimators: [100, 200, 300] (ağaç sayısı)
- max_depth: [10, 20, 30, None] (maksimum derinlik)
- min_samples_split: [2, 5, 10] (bölünme için minimum örnek sayısı)
- min_samples_leaf: [1, 2, 4] (yaprak için minimum örnek sayısı)

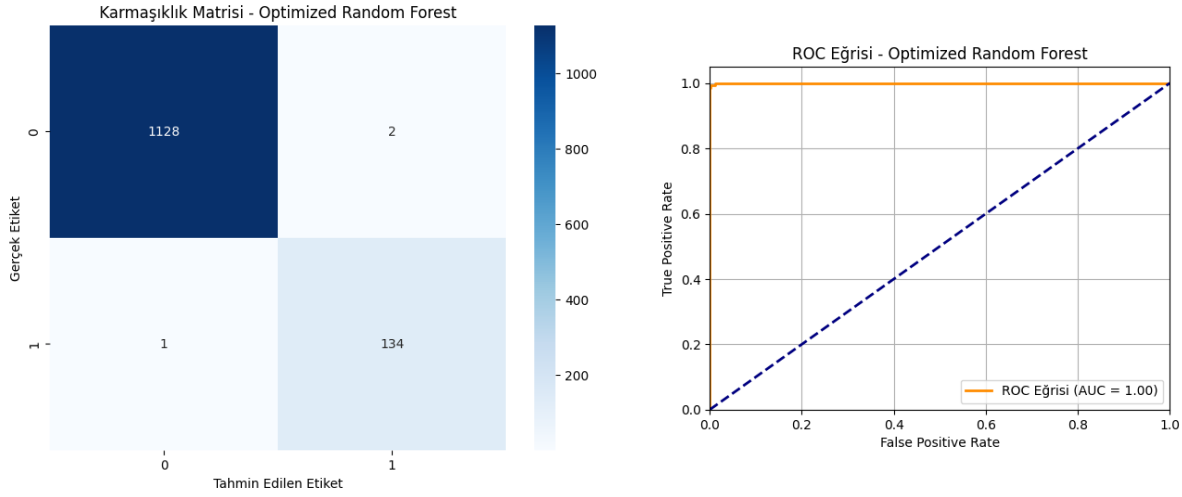
SVM:

- C: [0.1, 1, 10, 100] (düzenleştirme parametresi)
- gamma: ['scale', 'auto', 0.1, 0.01] (kernel katsayısı)
- kernel: ['rbf', 'linear'] (kernel fonksiyonu)

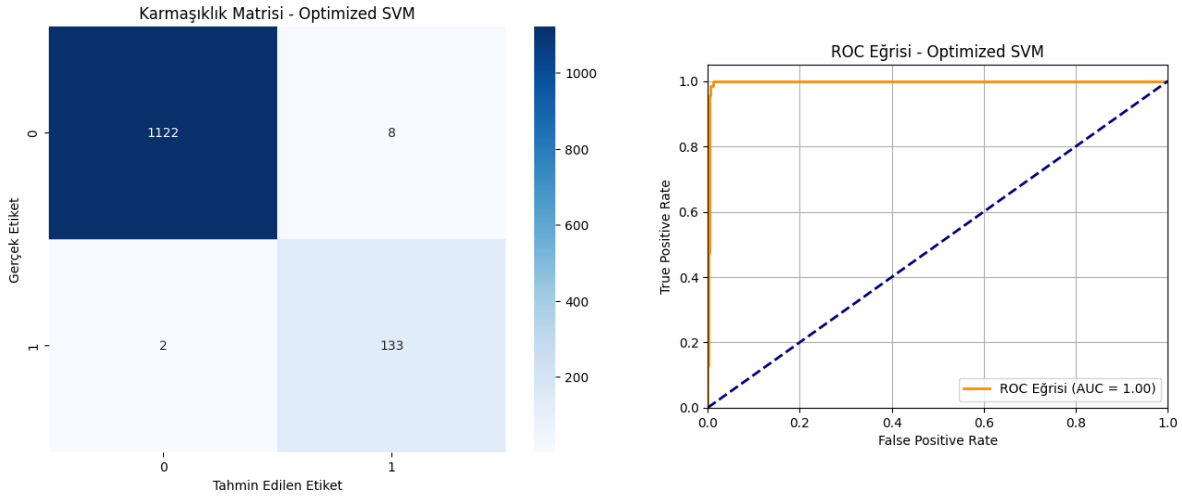
KNN:

- n_neighbors: [3, 5, 7, 9, 11] (komşu sayısı)
- weights: ['uniform', 'distance'] (ağırlıklandırma yöntemi)
- metric: ['euclidean', 'manhattan'] (mesafe metriği)

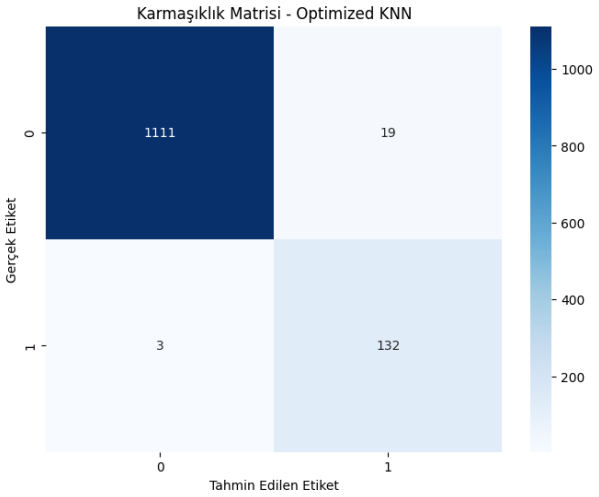
6.2 Optimize Model Performansı

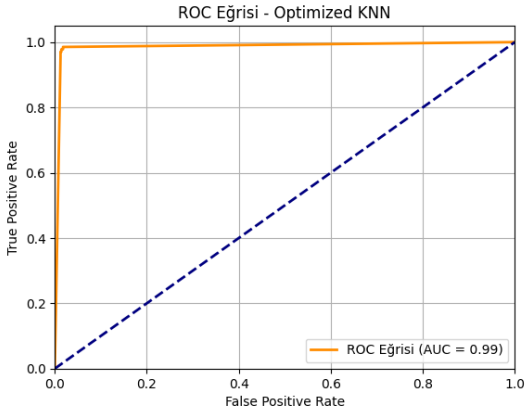


Optimize Random Forest – Karmaşıklık Matrisi | Roc Eğrisi

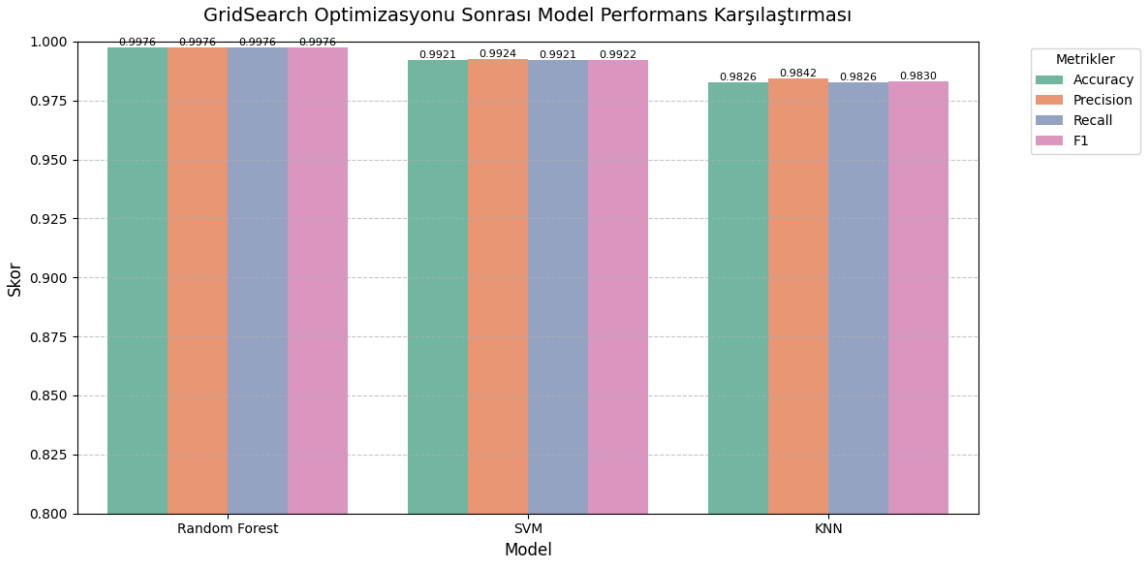


Optimize SVM – Karmaşıklık Matrisi | Roc Eğrisi





Optimize KNN – Karmaşıklık Matrisi | Roc Eğrisi

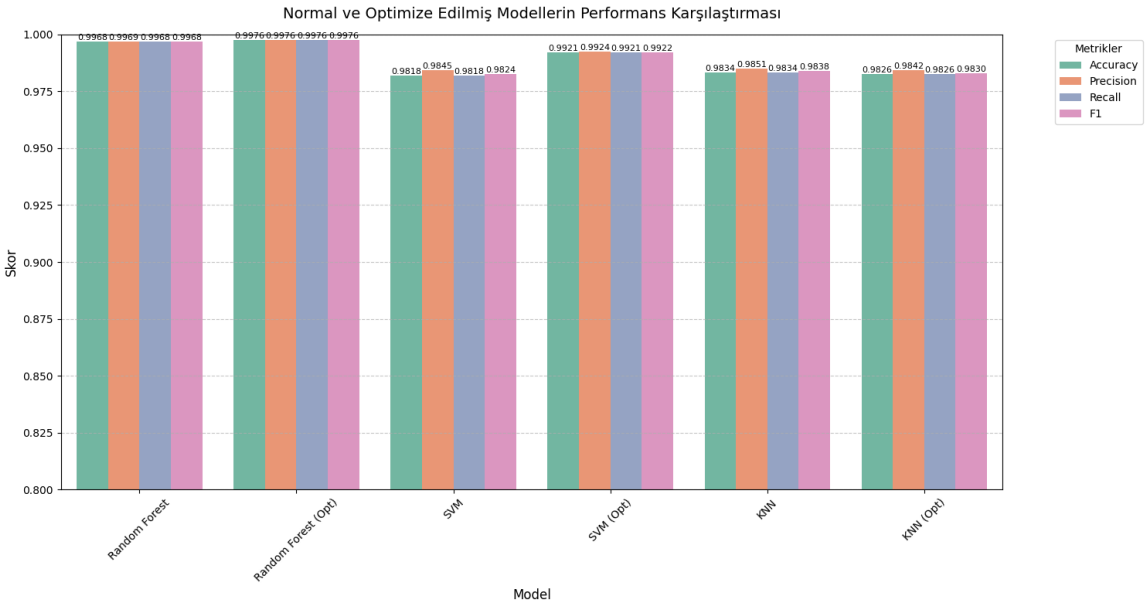


Modellerin Optimize Performans Metriklerinin Karşılaştırılması

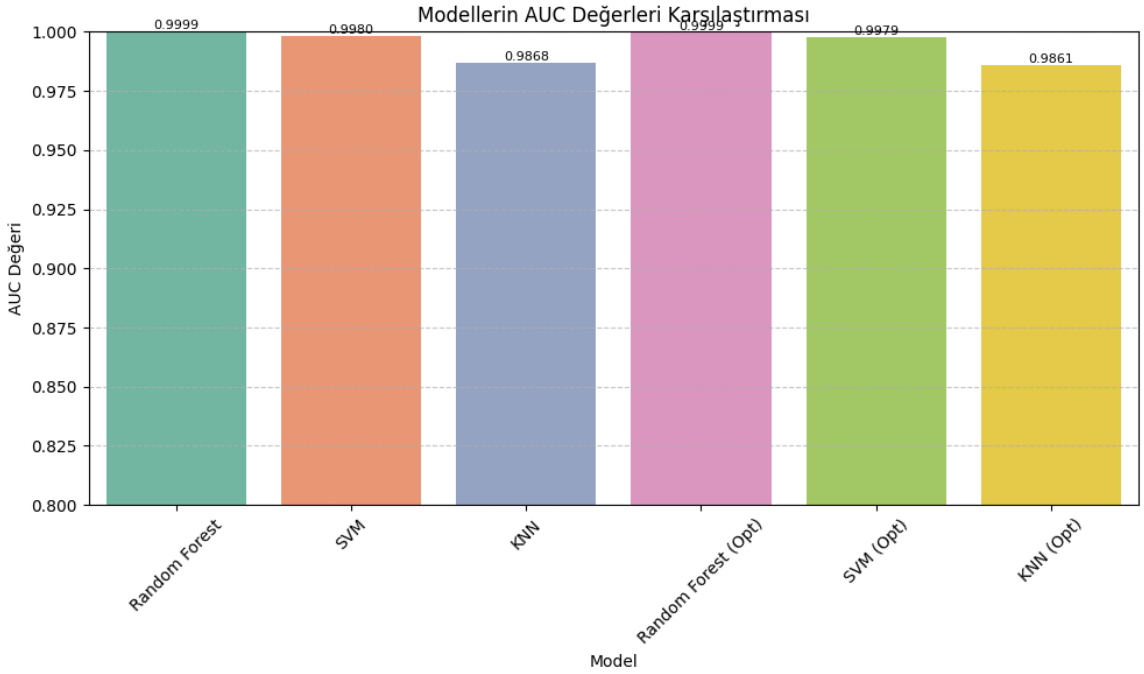
7. Sonuçlar ve Karşılaştırma

7.1 Model Performans Tablosu

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.9968	0.9965	0.9968	0.9968
Random Forest (Opt)	0.9976	0.9976	0.9976	0.9976
SVM	0.9818	0.9845	0.9818	0.9824
SVM (Opt)	0.9921	0.9924	0.9921	0.9922
KNN	0.9834	0.9851	0.9834	0.9838
KNN (Opt)	0.9826	0.9842	0.9826	0.9830



Tüm Modellerin Performans Metriklerinin Karşılaştırılması



Tüm Modellerin AUC Değerlerinin Karşılaştırılması

7.2 İyileştirme Oranları

- Random Forest: +0.08% iyileştirme
- SVM: +1.00% iyileştirme
- KNN: -0.08% değişim

7.3 En İyi Performans

- En iyi model: Optimize edilmiş Random Forest
- En yüksek iyileştirme: SVM
- En stabil performans: Random Forest

8. Öneriler ve Gelecek Çalışmalar

8.1 Veri Seti İyileştirmeleri

- Veri setinin daha güncel örneklerle genişletilmesi
- Sahte kullanıcı örneklerinin arttırılması
- Yeni özelliklerin eklenmesi

8.2 Model Geliştirmeleri

- Derin öğrenme tabanlı modellerin denenmesi
- İstatistiksel testlerle desteklenen hibrid sistemlerin araştırılması
- Ensemble yöntemlerin genişletilmesi

9. Karşılaştırmalı Performans Analizi

Bu bölümde, tarafımdan geliştirilen sınıflandırma modellerinin sonuçları, literatürde aynı veri seti üzerinde çalışan **Lucas Pontes (RPubs, 2022)** tarafından gerçekleştirilen çalışma ile karşılaştırılmaktadır.

Veri Ön İşleme ve Keşifsel Veri Analizi (EDA):

- Veri setindeki eksik değerler kontrol edilmiş ve gerekli temizlik işlemleri uygulanmıştır.
- Özelliklerin dağılımları incelenmiş ve görselleştirilmiştir.
- Özellik mühendisliği adımlarıyla bazı yeni değişkenler türetilmiştir.

Boyut İndirgeme Teknikleri:

- **Principal Component Analysis (PCA):** Veri setindeki boyutluluğu azaltmak ve önemli bileşenleri belirlemek için kullanılmıştır.
- **Linear Discriminant Analysis (LDA):** Sınıflar arasındaki ayrımı maksimize etmek amacıyla uygulanmıştır.

Modelleme Uygulamaları:

- **Sınıflandırma Algoritmaları:**
    - **Karar Ağacı (Decision Tree):** En iyi performansı gösteren model olarak belirlenmiştir.
    - **Kullanılan Özellikler:** Successive_Outbidding, Winning_Ratio, Starting_Price_Average, Auction_Duration.

Model	Doğruluk (Accuracy)	F1 Skoru	Kesinlik (Precision)	Duyarlılık (Recall)
Lucas Pontes - Random Forest	0.9955	0.9967	0.9982	0.9967
Semih Temiz – Random Forest	0.9968	0.9968	0.9969	0.9968
Semih Temiz – Random Forest (Opt)	0.9976	0.9976	0.9976	0.9976

10. GitHub Bağlantısı

Proje kodları, veri ön işleme, eğitim, modelleme ve görselleştirme adımları dahil olmak üzere tüm içeriğe aşağıdaki bağlantıdan erişebilirsiniz:

“GitHub Repository”

11. Proje Sunum Videosu

Projenin problem tanımı, kullanılan yöntemler ve sonuçları anlatan tanıtım videosu:

"Veri Madenciliği ile Sahte Teklif (Shill Bidding) Tespiti"