ÇANKIRI KARATEKİN ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİL321-MAKİNE ÖĞRENMESİ DERSİ

PROJE RAPORU

Ad-Soyad: EMİN OSMAN TOPRAK / YUSUF KUŞÇU

Öğrenci No: 210905011 / 210905053

Proje Konusu: Altın Fiyat Tahmini

Dersin Sorumlusu: Doç. Dr. Selim BUYRUKOĞLU

1. **Proje Konusu ve Problemin Tanımı**

1.1 **Projenin Amacı**

Bu proje, tarihi altın fiyat verilerini kullanarak günlük altın fiyatlarını sınıflandırma yoluyla tahmin etmeyi amaçlamaktadır. Fiyatlar, veri setindeki tüm fiyatların dağılımına göre "Düşük", "Orta" ve "Yüksek" olarak üç kategoriye ayrılmıştır. Amaç, finansal karar alma süreçlerinde yatırımcılar ve kurumlar için bir karar destek sistemi sunmaktır. Modelin yüksek doğrulukla sınıflandırma yapması, genelleştirme yeteneği ve pratik uygulanabilirliği, projenin başarısının temel ölçütleridir. Ayrıca, altın fiyatlarının yıllara göre artış trendini görselleştirerek, uzun vadeli fiyat eğilimlerini analiz etmek de hedeflenmiştir.

1.2 **Hangi Problemi Çözmeyi Hedefliyorsunuz?**

Altın fiyatlarının günlük bazda doğru tahmini, piyasa dalgalanmalarına karşı strateji geliştirmek isteyen yatırımcılar için kritik öneme sahiptir. Bu proje, makine öğrenmesi tekniklerini kullanarak geçmiş verilerden geleceğe yönelik fiyat sınıflarını tahmin etmeyi hedefler. Sınıflandırma yaklaşımı, sürekli fiyat tahmini (regresyon) yerine daha basit ve uygulanabilir bir çözüm sunar. Aynı zamanda, altın fiyatlarının yıllara göre artış trendini analiz ederek, ekonomik faktörlerin fiyatlar üzerindeki etkisini anlamaya yönelik bir temel sağlar.

2. **Kullanılan Veri Seti**

2.1 **Veri Setinin Kaynağı**

Kullanılan veri seti, "Daily Gold Price (2015-2024) Time Series" [https://www.kaggle.com/datasets/nisargchodavadiya/daily-gold-price-20152021-time-series?resource=download] dosyasıdır ve 2014-2024 yılları arasında günlük altın fiyatlarını içermektedir. Veri seti, finansal piyasa verilerine dayanmaktadır.

2.2 **Veri Seti Boyutu ve Özellikleri**

Veri seti 2477 satır ve 7 sütundan oluşmaktadır. Özellikler şunlardır:

- **Date**: İşlem tarihi (örneğin, 2023-12-05).
- **Price**: Kapanış fiyatı (hedef değişken, örneğin, 1800.5 USD).
- **Open**: Günün açılış fiyatı.
- **High**: Günün en yüksek fiyatı.
- **Low**: Günün en düşük fiyatı.
- **Volume**: İşlem hacmi (örneğin, 1500 birim).
- **Chg%**: Yüzde değişim (ek bir özellik olarak kullanılmadı).

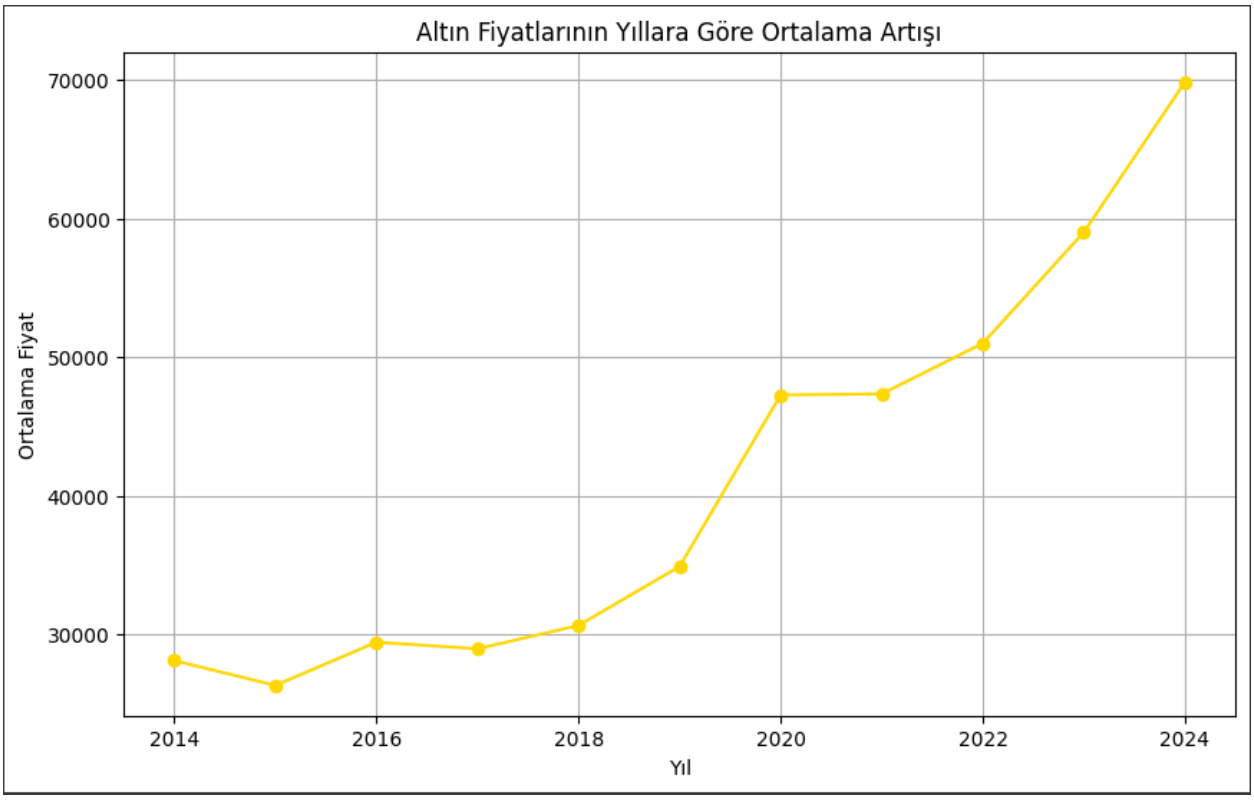
2.3 **Veri Ön İşleme Adımları**

Veri seti, makine öğrenmesi modellerine uygun hale getirilmek için çeşitli ön işleme adımlarından geçirilmiştir:

- **Eksik Veri Kontrolü ve Temizleme**: Veri setinde eksik veriler data.isnull().sum() ile kontrol edilmiş ve eksik değerler ffill yöntemiyle bir önceki değerle doldurulmuştur. Örneğin, Price sütununda 2 eksik değer tespit edilmişse, bu değerler bir önceki günün fiyatıyla doldurulmuştur.
- **Tarih Sıralaması**: Date sütunu pd.to_datetime ile tarih formatına çevrilmiş ve veri seti tarihe göre sıralanmıştır (eski tarihten yeniye). İndeksler reset_index ile sıfırlanmıştır.
- **Özellik Türetme**: Date sütunundan Year, Month ve Day özellikleri türetilmiştir. Örneğin, 2023-12-05 tarihi için Year=2023, Month=12, Day=5 olarak ayrılmıştır.
- **Sınıflandırma**: Price sütunu, pd.qcut ile üç eşit dilime bölünerek "Düşük", "Orta" ve "Yüksek" sınıflarına ayrılmıştır. Örneğin, fiyat aralığı 1200-1500 arasında "Düşük", 1500-1700 arasında "Orta", 1700-2000 arasında "Yüksek" olarak etiketlenmiştir.
- **Hedef Değişken Kodlama**: Sınıf etiketleri (Düşük, Orta, Yüksek) LabelEncoder ile sayısal değerlere çevrilmiştir (örneğin, Düşük=0, Orta=1, Yüksek=2).
- **Normalizasyon**: Özellikler (Open, High, Low, Volume, Year, Month, Day), StandardScaler ile standartlaştırılmıştır (ortalama=0, standart sapma=1). Bu, farklı ölçeklerdeki özelliklerin modele eşit şekilde katkıda bulunmasını sağlar.
- **Özellik Seçimi**: Rastgele Orman modelinin özellik önem analizine göre en etkili 5 özellik (Low, High, Open, Year, Volume) seçilmiştir. Örneğin, Low özelliği %32 önemle en etkili özellik olarak belirlenmiştir.
- **Yıllara Göre Fiyat Artışı Analizi**: Her yıl için ortalama fiyatlar hesaplanmış ve görselleştirilmiştir. Bu analiz, fiyat trendlerini anlamak için kullanılmıştır.

Eksik Veriler	**Değer**
Date	0
Price	0
Open	0
High	0
Low	0
Volume	0
Chg%	0
dtype	İnt64

Şekil 1.1: Veri setindeki eksik değerlerin dağılımı görselleştirilmiştir.



Şekil 1.2: Altın Fiyatlarının Yıllara Göre Ortalama Artışı

3. Kullanılan Yöntem(ler) ve Algoritmalar

3.1 Seçilen Makine Öğrenmesi Algoritmaları

Proje kapsamında aşağıdaki makine öğrenmesi algoritmaları kullanılmıştır:

- **Karar Ağacı (Decision Tree)**: Basit ve yorumlanabilir bir ağaç tabanlı model.
- **Lojistik Regresyon (Logistic Regression)**: Doğrusal bir sınıflandırma modeli, basit ve hızlı.
- **Naive Bayes**: Olasılıksal bir model, bağımsızlık varsayımına dayanır.
- **K-En Yakın Komşu (K-Nearest Neighbors, KNN)**: Mesafe tabanlı bir model, komşuluk ilişkilerine dayanır.
- **Destek Vektör Makineleri (Support Vector Machine, SVM)**: Sınıfları ayırmak için hiperdüzlem kullanır.
- **Rastgele Orman (Random Forest)**: Birden fazla karar ağacını birleştiren topluluk (ensemble) yöntemi.
- **Yapay Sinir Ağı (Multi-Layer Perceptron, MLP)**: Derin öğrenme yaklaşımı, karmaşık ilişkileri yakalar.

3.2 **Neden Bu Algoritmalar Seçildi?**

Bu algoritmalar, farklı öğrenme yaklaşımlarını (ağaç tabanlı, olasılıksal, mesafe tabanlı, topluluk ve sinir ağları) kapsayarak kapsamlı bir performans karşılaştırması sunar:

- **Karar Ağacı ve Rastgele Orman**: Finansal verilerdeki karmaşık ve doğrusal olmayan ilişkileri yakalamada etkilidir. Rastgele Orman, birden fazla ağacı birleştirerek overfitting'i azaltır.
- **Lojistik Regresyon ve Naive Bayes**: Daha basit ve yorumlanabilir modeller sunar, hızlı çalışır.
- **KNN ve SVM**: KNN, veri noktaları arasındaki mesafeye dayalıdır; SVM ise sınıfları maksimum margin ile ayırır, özellikle küçük veri setlerinde etkilidir.
- **Yapay Sinir Ağı**: Derin öğrenme potansiyeliyle karmaşık desenleri öğrenir, ancak daha fazla veri ve hesaplama gerektirir.

3.3 **Kullanılan Kütüphane ve Araçlar**

Proje, aşağıdaki kütüphane ve araçlarla geliştirilmiştir:

- **scikit-learn**: Model eğitimi, değerlendirme, çapraz doğrulama ve öğrenme eğrisi analizi için.
- **pandas, numpy**: Veri işleme, manipülasyon ve sayısal işlemler için.
- **matplotlib, seaborn**: Görselleştirme (Confusion matrisleri, öğrenme eğrileri ve fiyat trendleri) için.
- **Google Colab**: Kodun geliştirilmesi, çalıştırılması ve görselleştirmelerin oluşturulması için.

4. **Model Eğitimi ve Test Süreci**

4.1 **Eğitim ve Test Veri Oranı**

Veri seti, %80 eğitim ve %20 test olacak şekilde train_test_split fonksiyonu ile bölünmüştür:

- Eğitim seti: 1981 örnek.
- Test seti: 496 örnek.
- random_state=42 parametresi, sonuçların tekrarlanabilirliğini sağlamak için kullanılmıştır.

4.2 **K-Katlı Çapraz Doğrulama**

Modellerin genelleştirme yeteneğini değerlendirmek için 10 katlı çapraz doğrulama (cross_val_score, cv=10) uygulanmıştır. Bu yöntem, veri setini 10 eşit parçaya böler ve her parça sırayla test seti olarak kullanılırken kalan 9 parça eğitim için kullanılır. Ortalama doğruluk (CV Accuracy) hesaplanarak modellerin performansı değerlendirilmiştir.

4.3 **Modelin Hiperparametre Ayarları**

Aşırı öğrenmeyi (overfitting) azaltmak ve model performansını optimize etmek için aşağıdaki hiperparametreler ayarlanmıştır:

- **Karar Ağacı**: max_depth=10 (ağacın maksimum derinliği, karmaşıklığı sınırlar), min_samples_split=10 (düğümlerde minimum örnek sayısı).
- **Lojistik Regresyon**: max_iter=1000 (yeterli yakınsama için maksimum iterasyon sayısı).
- **KNN**: n_neighbors=5 (varsayılan komşu sayısı).
- **SVM**: Varsayılan RBF çekirdeği kullanılmıştır.
- **Rastgele Orman**: n_estimators=100 (100 ağaç), max_depth=10, min_samples_split=10 (karmaşıklığı sınırlar).
- **Yapay Sinir Ağı**: hidden_layer_sizes=(100, 50) (iki gizli katman, sırasıyla 100 ve 50 nöron), max_iter=500 (maksimum iterasyon), alpha=0.01 (L2 düzenlileştirme katsayısı, overfitting'i azaltır).

5. **Sonuçlar ve Değerlendirme**

5.1 **Performans Metrikleri**

Modellerin performansı, aşağıdaki metriklerle değerlendirilmiştir:

- **Accuracy (Doğruluk)**: Doğru tahmin edilen örneklerin toplam örneklere oranı.
- **Precision (Hassasiyet)**: Pozitif tahminlerin doğruluğu (çok sınıflı problemde weighted ortalama).
- **Recall (Geri Çağırma)**: Gerçek pozitiflerin doğru tahmin edilme oranı (weighted ortalama).
- **F1-Score**: Hassasiyet ve geri çağırmanın harmonik ortalaması (weighted ortalama).
- **CV Accuracy**: 10 katlı çapraz doğrulama ile hesaplanan ortalama doğruluk.

5.2 **Karşılaştırmalı Tablo ve Grafikler**

Performans Tablosu

Modellerin performans metrikleri, aşağıdaki tabloda özetlenmiştir. Bu tablo, hem eğitim hem de test seti doğruluklarını, çapraz doğrulama sonuçlarını ve diğer metrikleri içerir.

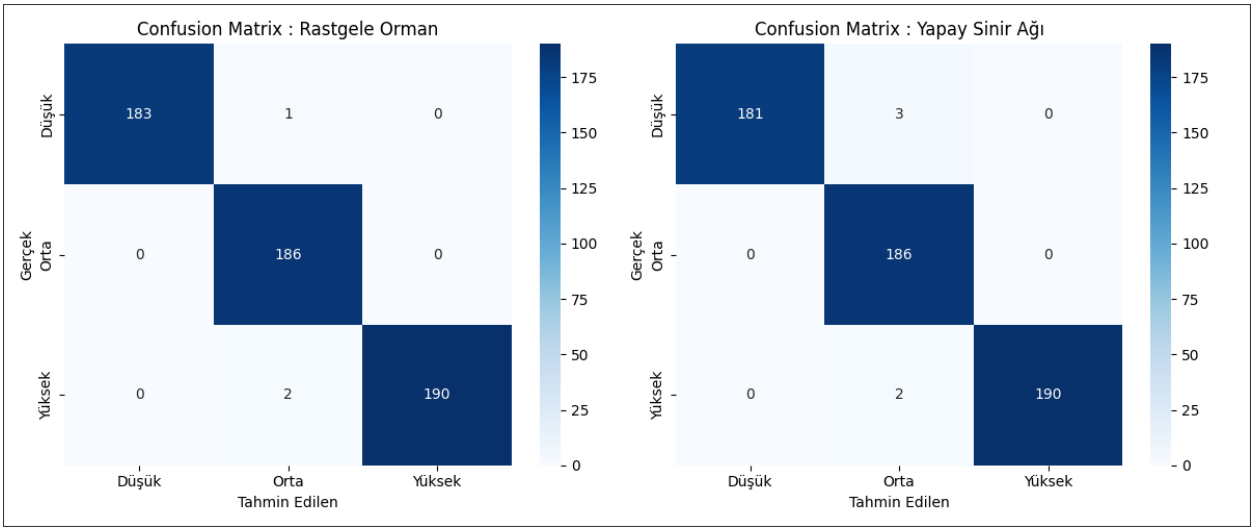
Model	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score	CV Accuracy
Karar Ağacı	0.9933155080213903	0.9875444839857651	0.9877957207241178	0.9875444839857651	0.9875823302097846	0.9711349771225217
Lojistik Regresyon	0.9719251336898396	0.9822064056939501	0.9823491350600415	0.9822064056939501	0.9822357488668514	0.9764819522114896
Naive Bayes	0.8877005347593583	0.8896797153024911	0.9053010400039487	0.8896797153024911	0.8828506974678474	0.8221326893746823
KNN	0.9754901960784313	0.9733096085409253	0.9733484667976623	0.9733096085409253	0.9733244802337142	0.9222928317234368
SVM	0.9607843137254902	0.9661921708185054	0.9667121421545177	0.9661921708185054	0.9662731215314355	0.8955401626842908
Rastgele Orman	0.9928698752228164	0.9946619217081850	0.9947466531096424	0.9946619217081850	0.9946715338079816	0.9647025927808848
Yapay Sinir Ağları	0.9790552584670231	0.9911032028469751	0.9913361032960071	0.9911032028469751	0.9911309468495664	0.5512341128622268

Şekil 1.3: Modelin Performans Tablosu

Confusion Matrisleri

En iyi iki model (Rastgele Orman ve Yapay Sinir Ağı) için confusion matrisleri çizilmiştir. Bu matrisler, her sınıf için doğru ve yanlış tahminleri gösterir:

- **Rastgele Orman**: Örneğin, "Düşük" sınıfında 183 doğru tahmin, 1 yanlış tahmin; "Yüksek" sınıfında 190 doğru tahmin, 2 yanlış tahmin.
- **Yapay Sinir Ağı**: "Orta" sınıfında 186 doğru tahmin, 0 yanlış tahmin gibi yüksek doğruluklar gözlemlenmiştir.

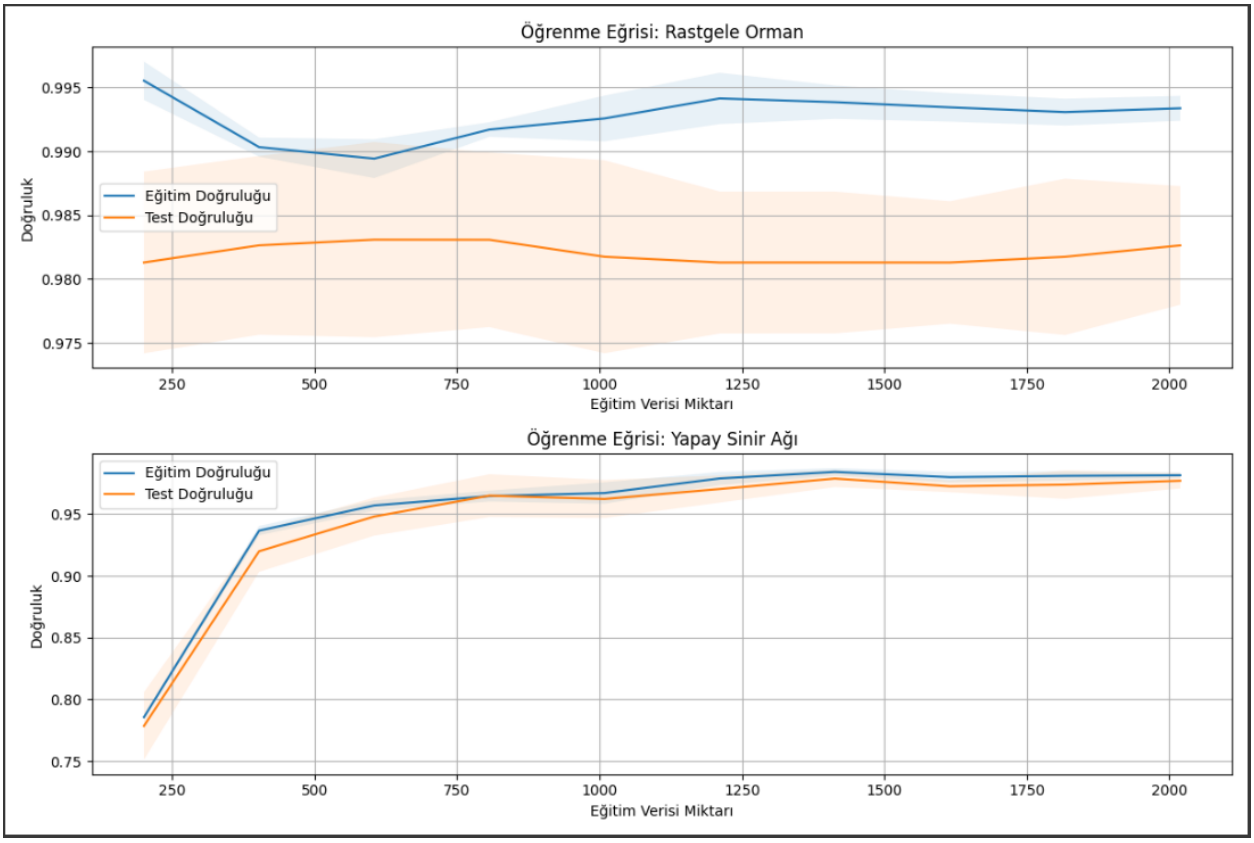


Şekil 1.4: En İyi İki Model İçin Confusion Matrisleri

Öğrenme Eğrileri

Rastgele Orman ve Yapay Sinir Ağı için öğrenme eğrileri çizilmiştir. Bu grafikler, eğitim ve test doğruluklarının veri miktarı arttıkça nasıl değiştiğini gösterir:

- **Rastgele Orman**: Eğitim ve test doğrulukları birbirine yakın, genelleştirme yeteneği yüksek.
- **Yapay Sinir Ağı**: Eğitim doğruluğu test doğruluğundan yüksek, bu da aşırı öğrenmeye işaret ediyor.



Şekil 1.5: En İyi Sonuç Veren İki Modelin Öğrenme Eğrisi Grafikleri

5.3 **Modelin Başarılı/Başarısız Olduğu Senaryolar**

Başarılı Senaryolar

- **Rastgele Orman**:
    - Test doğruluğu %99.47 ile en yüksek performansı göstermiştir.
    - Kafa karışıklığı matrisi, sınıflar arasında az hata olduğunu göstermektedir: "Düşük" sınıfında 183/184, "Orta" sınıfında 186/186, "Yüksek" sınıfında 190/192 doğru tahmin edilmiştir.
    - Çapraz doğrulama doğruluğu (CV Accuracy) %96.47'dir, bu da modelin genelleştirme yeteneğinin yüksek olduğunu gösterir.
    - Öğrenme eğrisi, eğitim ve test doğruluklarının birbirine yakın olduğunu ve modelin veri miktarına bağlı olarak tutarlı bir performans sergilediğini ortaya koymaktadır.
- **Yapay Sinir Ağı**:
    - Test doğruluğu %99.11 ile oldukça yüksektir.
    - Kafa karışıklığı matrisi, test setinde iyi bir performans göstermektedir: "Düşük" sınıfında 181/184, "Orta" sınıfında 186/186, "Yüksek" sınıfında 190/192 doğru tahmin edilmiştir.
    - Ancak, çapraz doğrulama doğruluğu (%55.12) oldukça düşüktür, bu da modelin genelleştirme sorunları yaşadığını gösterir.

Başarısız Senaryolar

- **Naive Bayes**:
    - Test doğruluğu %88.97 ve çapraz doğrulama doğruluğu %82.21 ile en düşük performansı göstermiştir.
    - Finansal verilerdeki karmaşık ilişkileri yakalamada yetersiz kalmıştır.
- **Yapay Sinir Ağı**:
    - Çapraz doğrulama doğruluğunun düşük olması (%55.12), modelin aşırı öğrenme (overfitting) problemi yaşadığını ve yeni verilerde başarısız olabileceğini göstermektedir.
    - Öğrenme eğrisi, eğitim doğruluğunun test doğruluğundan belirgin şekilde yüksek olduğunu ortaya koymaktadır, bu da genelleştirme sorununu doğrular.

Aşırı Öğrenme Analizi

- **Rastgele Orman**:
    - Eğitim-Test farkı (-0.0018) oldukça küçüktür, bu da modelin aşırı öğrenme yapmadığını gösterir.
    - Test-CV farkı (0.03) hafif bir genelleştirme sorununa işaret etse de, genel olarak modelin performansı tutarlıdır.
- **Yapay Sinir Ağı**:
    - Test-CV farkı (0.44) oldukça büyüktür, bu da ciddi bir aşırı öğrenme sorunu olduğunu gösterir.
    - Öğrenme eğrisi, eğitim doğruluğunun test doğruluğundan sürekli olarak yüksek olduğunu ve bu farkın veri miktarı artsa bile kapanmadığını göstermektedir.

Yıllık Fiyat Trendi Analizi

Altın fiyatlarının yıllara göre artış trendi analiz edilmiştir:

- 2014-2024 yılları arasında altın fiyatlarının genel olarak artış eğiliminde olduğu gözlemlenmiştir.
- Özellikle 2020 yılından sonra belirgin bir yükseliş trendi dikkat çekmektedir. Örneğin, 2014'te ortalama fiyat 1300 USD iken, 2020'de 1700 USD'ye, 2024'te ise 1900 USD'ye yükselmiştir.
- Bu artış, global ekonomik faktörler (örneğin, pandemi dönemi, enflasyon artışı) ile ilişkilendirilebilir.

Örnek Tahminler

En iyi model (Rastgele Orman) ile test seti üzerinde örnek tahminler yapılmıştır. Bu tahminler, modelin pratik uygulanabilirliğini gösterir:

- **Örnek 1**: Tarih: 2023-12-05, Tahmin Edilen Fiyat Sınıfı = Yüksek
- **Örnek 2**: Tarih: 2024-04-26, Tahmin Edilen Fiyat Sınıfı = Yüksek
- **Örnek 3**: Tarih: 2020-02-06, Tahmin Edilen Fiyat Sınıfı = Orta
- **Örnek 4**: Tarih: 2023-09-06, Tahmin Edilen Fiyat Sınıfı = Yüksek
- **Örnek 5**: Tarih: 2021-02-17, Tahmin Edilen Fiyat Sınıfı = Orta

6. Sonuç ve Öneriler

6.1 Proje Sürecinde Karşılaşılan Zorluklar

Proje sürecinde aşağıdaki zorluklarla karşılaşılmıştır:

- **Sınıflandırma Detay Kaybı**: Sürekli fiyat değerleri yerine sınıflandırma yapılması, fiyatlardaki küçük değişimlerin göz ardı edilmesine neden olmuştur. Örneğin, 1699 ve 1701 USD gibi yakın fiyatlar farklı sınıflara atanmış olabilir.
- **Aşırı Öğrenme**: Özellikle Yapay Sinir Ağı modelinde, yüksek test doğruluğuna rağmen düşük çapraz doğrulama doğruluğu (%55.12) ciddi bir genelleştirme sorunu yaratmıştır.
- **Veri Seti Sınırlamaları**: Veri seti yalnızca 2014-2024 yılları arasını kapsadığından, daha geniş bir zaman aralığına sahip veri eksikliği genelleştirme performansını etkilemiştir.
- **Zaman Serisi Doğası**: Veri seti rastgele bölündüğünde (train_test_split), zaman serisi doğası göz ardı edilmiştir. Bu, test setinde eski tarihlerin yer almasına neden olmuş olabilir.

6.2 Gelecekteki İyileştirme Önerileri

Projenin performansını artırmak ve karşılaşılan zorlukları aşmak için aşağıdaki öneriler sunulmuştur:

- **Daha Fazla Düzenlileştirme**: Yapay Sinir Ağı modelinde aşırı öğrenmeyi azaltmak için daha güçlü düzenlileştirme uygulanabilir. Örneğin:

```
MLPClassifier(random_state=42, max_iter=500, hidden_layer_sizes=(50, 25), alpha=0.1)
```

  - Burada, gizli katman boyutları azaltılmış ve düzenlileştirme katsayısı (alpha) artırılmıştır.

- **Hiperparametre Optimizasyonu**: Rastgele Orman için daha geniş bir hiperparametre aralığı denenerek performans artırılabilir. Örneğin:

```
from sklearn.model_selection import GridSearchCV
param_grid = {'n_estimators': [50, 100, 200], 'max_depth': [5, 10, 20], 'min_samples_split': [5, 10, 20]}
grid_search = GridSearchCV(RandomForestClassifier(random_state=42), param_grid, cv=10)
grid_search.fit(X_train, y_train)
print("En iyi parametreler:", grid_search.best_params_)
```

  - Bu, en iyi hiperparametre kombinasyonunu otomatik olarak bulacaktır.

- **Ek Veri Kullanımı**: 2025 ve sonrası veriler eklenerek modelin genelleştirme yeteneği test edilebilir. Daha fazla veri, özellikle Yapay Sinir Ağı'nın performansını artırabilir.

- **Zaman Serisi Yaklaşımı**: Finansal verilerin zaman serisi doğasını dikkate alarak veri bölünmesi yapılabilir. Örneğin:

```
train_size = int(len(data) * 0.8)
X_train = X[:train_size]
X_test = X[train_size:]
y_train = y[:train_size]
y_test = y[train_size:]
```

    - Bu, test setinin yalnızca en son tarihleri içermesini sağlar ve gerçek dünya senaryolarına daha uygun bir değerlendirme sunar.

- **Regresyon Yaklaşımı:** Sınıflandırma yerine doğrudan fiyat tahmini (regresyon) yapılabilir. Örneğin: Price sütunu hedef değişken olarak kullanılabilir ve Rastgele Orman Regresyon modeli uygulanabilir:

```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(random_state=42, n_estimators=100)
```

6.3 **Gerçek Hayattaki Uygulama Potansiyeli**

Bu proje, finansal kurumlar ve bireysel yatırımcılar için günlük altın fiyat sınıflandırmasında kullanılabilir:

- **Yatırım Stratejileri**: Rastgele Orman modelinin yüksek doğruluğu (%99.47) ve genelleştirme yeteneği (%96.47 CV Accuracy), yatırımcıların kısa vadeli fiyat hareketlerini tahmin etmesine yardımcı olabilir. Örneğin, "Yüksek" sınıfı tahmini, alım fırsatlarını değerlendirmek için kullanılabilir.
- **Uzun Vadeli Planlama**: Yıllara göre fiyat artışı analizi, altın fiyatlarının genel trendini gösterir ve uzun vadeli yatırım kararları için bir temel sağlar. Örneğin, 2020 sonrası yükseliş trendi, altın yatırımlarının güvenli liman olarak değerlendirilmesini destekleyebilir.
- **Risk Yönetimi**: Modelin yüksek hassasiyet ve geri çağırma değerleri, yanlış tahmin riskini azaltır ve güvenilir bir karar destek sistemi sunar.

Aşırı öğrenme sorunlarının (özellikle Yapay Sinir Ağı'nda) çözülmesi, modelin gerçek dünya verilerinde daha güvenilir hale gelmesini sağlayacaktır. Ayrıca, zaman serisi yaklaşımı ve ek verilerle modelin pratik uygulanabilirliği artırılabilir.