Gerçek Zamanlı Büyük Veri Analitiği ile Anomali Tespiti

Esma Gelmez, Hacer Sueda Efe

Bilişim Sistemleri Mühendisliği
Teknoloji Fakültesi - Kocaeli Üniversitesi
İzmit, Kocaeli /Türkiye

esmagelmez57@gmail.com , suedaefe@gmail.com

github : https://github.com/suedaefe/GoldPricePrediction

Giriş

Günümüzde finansal piyasalardaki dalgalanmalar ve çeşitli risk faktörleri, yatırımcıların karar verme süreçlerini doğrudan etkileyen temel unsurlar arasında yer almaktadır. Yatırımcılar, hem kazançlarını maksimize etmek hem de kayıplarını minimize etmek için daha bilinçli ve veriye dayalı kararlar almanın yollarını aramaktadır. Bu süreçte özellikle altın, ekonomik belirsizlik dönemlerinde güvenilir bir yatırım aracı olarak öne çıkmaktadır. Altın fiyatlarının doğru bir şekilde öngörülebilmesi, yatırımcılar için büyük bir avantaj sağlamanın yanı sıra piyasadaki risklerin daha iyi yönetilmesine de olanak tanımaktadır.

Bu bağlamda, son yıllarda makine öğrenmesi ve veri analizi teknikleri, finansal verilerin analiz edilmesinde kritik bir rol oynamaktadır. Gelişmiş algoritmalar ve modeller, özellikle zaman serisi verilerinde karmaşık örüntülerin öğrenilmesi ve tahmin edilmesi konusunda önemli katkılar sunmaktadır.

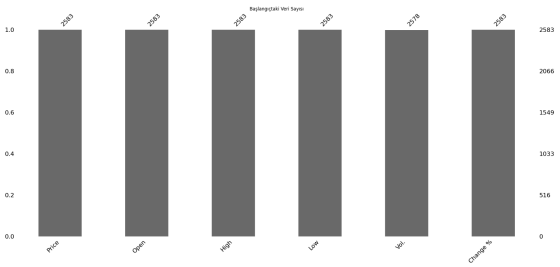
Bu çalışmada, finans piyasalarındaki bu ihtiyaca yanıt verebilmek amacıyla, son 10 yıllık altın fiyatı verileri kullanılarak bir LSTM (Long Short-Term Memory) modeli geliştirilmiştir. LSTM algoritması, geçmiş verilere dayalı olarak zaman serisi örüntülerini öğrenme konusundaki başarısıyla tanınmakta ve finansal tahmin uygulamalarında sıkça tercih edilmektedir.

Geliştirilen çalışmada, yalnızca altın fiyatları tahmin edilmekle kalmayıp, aynı zamanda gerçek zamanlı anomali tespiti de yapılmaktadır. Gerçek zamanlı olarak benzer veriler üretilip Kafka platformuna gönderilmekte ve bu veriler üzerinden potansiyel anormallikler bildirilmektedir. Bu sistem, finansal piyasalarda anormalliklerin erken tespiti ile yatırımcıların risk yönetimine katkı sağlamayı hedeflemektedir.

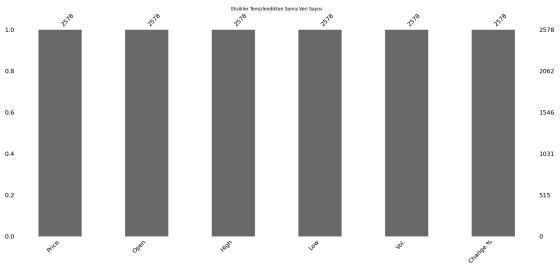
1. Veri Seti Seçimi ve Ön İşleme

Çalışmada 2012-2023 arası 10 yıllık günlük altın fiyatlarını içeren bir veri seti kullanılmıştır. Bu veri seti, zaman serisi analizi ve anomali tespiti için uygun bir yapı sunmaktadır. Veri setindeki her bir kayıt tarih ve fiyat bilgilerini içermektedir. Anomali tespiti ve tahmin modellerinin uygulanabilmesi için bu veri seti üzerinde ön işleme adımları gerçekleştirilmiştir.

Veri seti, Python kullanılarak Pandas kütüphanesi ile yüklenmiş ve incelenmiştir. İlk adımda, eksik veriler kontrol edilmiş ve tespit edilen eksik değer bulunan satırlar temizlenmiştir. Tarih sütunu, datetime formatına dönüştürülerek zaman sırasına göre sıralanmış ve veri setinin tarih sütunu indeks olarak belirlenmiştir.



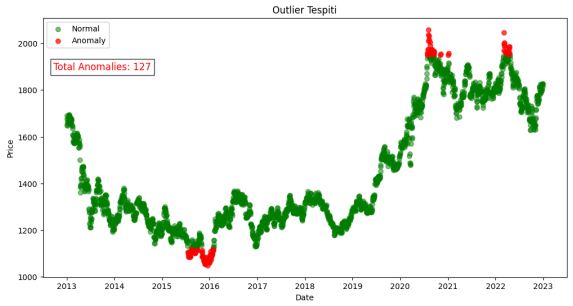
Şekil 1-Veri temizlemeden önceki veri sayısı



Şekil 2-Veri temizlendikten sonraki veri sayısı

Altın fiyatları sütunundaki veriler metinsel formatta olduğu için virgüller kaldırılmış ve sayısal bir formata dönüştürülmüştür. Fiyat verilerinin dağılımı histogram grafikleri ile görselleştirilmiş ve zaman içerisindeki değişimleri bir dağılım grafiği ile analiz edilmiştir. Veriler, MinMaxScaler kullanılarak 0 ile 1 arasında ölçeklendirilmiş, böylece model eğitimi için uygun hale getirilmiştir.

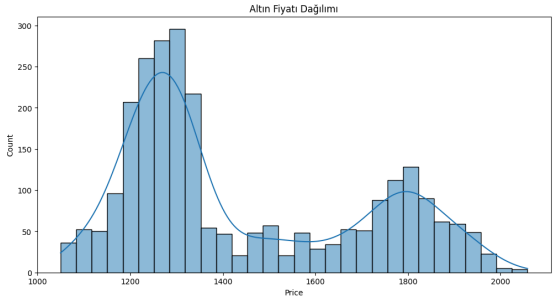
Anomali tespiti için Isolation Forest yöntemi uygulanmış, verideki aykırı değerler (-1) ve normal değerler (1) etiketlenmiştir. Aykırı değerler görselleştirilerek toplam anomali sayısı hesaplanmıştır.



Şekil 3-Outlier tespiti grafiği

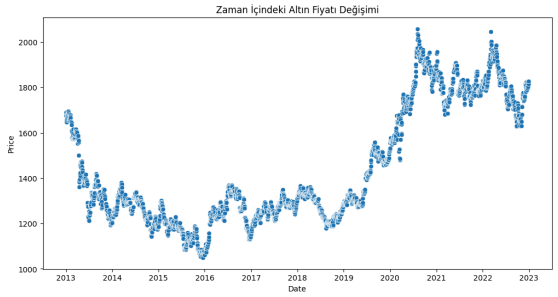
2. Veri Görselleştirme

Veri setinin görselleştirilmesi, altın fiyatlarının zaman içindeki değişimini ve genel dağılımını analiz etme konusunda önemli bilgiler sunmaktadır. Şekil 4'te, altın fiyatlarının dağılımını gösteren bir histogram yer almaktadır. Bu grafik, altın fiyatlarının genellikle 1200 ile 1400 arasında yoğunlaştığını ve daha yüksek fiyatlarda sayının azaldığını göstermektedir.



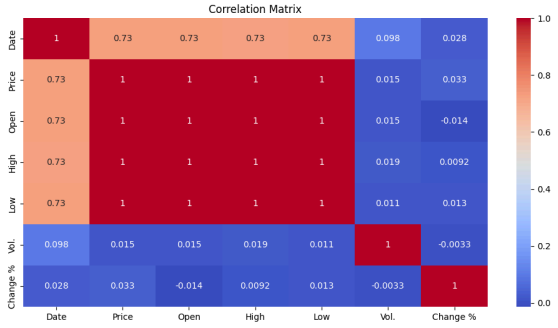
Şekil 4-Veri setindeki değişkenin dağılımını gösteren histogram.

Şekil 5'te ise, altın fiyatlarının zaman içindeki değişimini gösteren bir scatter plot bulunmaktadır. Bu grafik, 2013 ile 2023 yılları arasındaki dönemde altın fiyatlarının dalgalandığını ve özellikle 2020 sonrasında bir artış trendinin belirginleştiğini göstermektedir. Bu da, piyasa koşullarının zamanla nasıl evrildiğini gösteren önemli bir grafiktir.



Şekil 5-Veri setindeki değişkenin zaman içindeki scatter plot grafiği

Şekil 6'da ise, altın fiyatları ile diğer ekonomik faktörler arasındaki korelasyonları gösteren bir heatmap yer almaktadır. Bu harita, altın fiyatları ile tarih, açılış fiyatı, yüksek ve düşük fiyatlar arasında güçlü pozitif ilişkiler olduğunu; ancak işlem hacmi ve değişim oranı gibi faktörlerle daha zayıf ilişkiler bulunduğunu göstermektedir.



Şekil 6-Veri setindeki değişkenler arasındaki korelasyon matrisi

3. Yapay Zeka Modeli Geliştirme

Model, Long Short-Term Memory (LSTM) mimarisi kullanılarak tasarlanmıştır. LSTM, özellikle sıralı verilerin analizinde güçlü performans gösterir. Model geliştirme süreci, mimari tasarım, eğitim, hiperparametre seçimi ve değerlendirme aşamalarını kapsamaktadır.

3.1. Model Mimarisi

Geliştirilen modelde, zaman serisi verilerinin sıralı yapısını işleyebilmek için birden fazla LSTM katmanı kullanılmıştır. Modelin mimarisi şu şekilde yapılandırılmıştır:

Giriş Katmanı: Veri, (örnek sayısı, zaman adımları, özellik sayısı) formatında işlenmiştir. Bu format, zaman serisi verisinin sıralı yapısının korunmasını sağlamaktadır.

LSTM Katmanları:

- İlk LSTM katmanı 30 hücreden oluşacak şekilde tasarlanmıştır (units=30).

- İlk LSTM katmanında return_sequences=True parametresi kullanılarak, tüm zaman adımlarının çıktılarının bir sonraki katmana aktarılması sağlanmıştır.

- Aşırı öğrenmeyi önlemek için ağırlık düzenlemesi olarak kernel_regularizer=l2(0.01) eklenmiştir.

- İkinci LSTM katmanı return_sequences=False ile yapılandırılmıştır ve zaman serisi verisinin daha karmaşık özelliklerini öğrenmektedir.

Normalizasyon ve Dropout Katmanları:

- LSTM katmanlarının ardından, eğitim sürecinin kararlılığını artırmak için BatchNormalization uygulanmıştır.

- Overfitting'i önlemek için her LSTM katmanının ardından %40 oranında bağlantılar devre dışı bırakılmıştır (Dropout=0.4).

Yoğun Katman (Dense Layer):

- Son katmanda, modelin sürekli bir değer üretmesini sağlamak amacıyla yalnızca bir nöronlu bir yoğun katman bulunmaktadır.

- Bu katmanda doğrusal aktivasyon fonksiyonu (activation='linear') kullanılmıştır.

3.2. Modelin Eğitim Süreci

Modelin eğitimi sırasında, kayıp fonksiyonu, optimizasyon algoritması ve hiperparametre ayarları dikkatlice seçilmiştir:

Kayıp Fonksiyonu: Modelin doğruluğunu artırmak için Mean Squared Error (MSE) kayıp fonksiyonu tercih edilmiştir. Bu fonksiyon, tahmin edilen ve gerçek değerler arasındaki hata karelerinin ortalamasını minimize etmeyi amaçlamaktadır.

Optimizasyon Algoritması: Eğitim sürecinde Adam optimizasyon algoritması kullanılmıştır. Adam, adaptif öğrenme oranı ve momentumun birleşiminden faydalanarak hem hızlı hem de stabil bir öğrenme süreci sağlar. Başlangıç öğrenme oranı (learning_rate) olarak 0.001 değeri atanmıştır.

3.3. Hiperparametreler

- Epoch Sayısı: Modelin eğitimi maksimum 50 epoch ile sınırlandırılmıştır. Ancak aşırı öğrenmeyi önlemek için EarlyStopping mekanizması eklenmiştir. patience=5 değeriyle model, doğrulama kaybı 5 ardışık epoch boyunca iyileşme göstermediğinde eğitim durdurulmuştur.

- Batch Boyutu: Eğitim sırasında kullanılan batch boyutu 32 olarak belirlenmiştir.

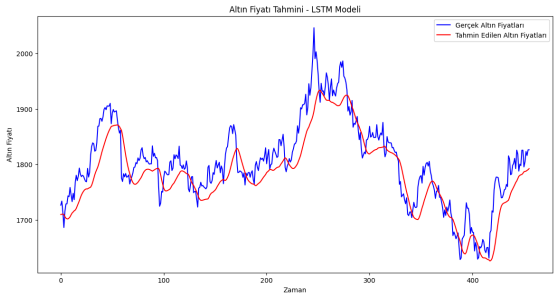
3.4. Performans Değerlendirme

Model eğitiminin ardından performansı test veri kümesi üzerinde değerlendirilmiştir.

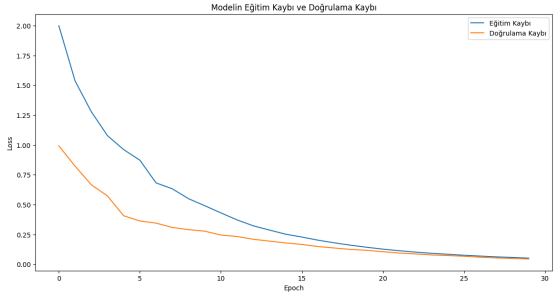
- Kayıp ve Doğrulama Grafikleri:
Eğitim sırasında, kayıp ve doğrulama kayıpları her epoch için izlenmiş ve grafikle görselleştirilmiştir. Bu, modelin aşırı öğrenme yapıp yapmadığını anlamak için kullanılmıştır.

- Ek Metrikler:
Modelin doğruluğunu daha iyi değerlendirebilmek için MSE'ye ek olarak Accuracy, Precision, Recall ve F1 skoru gibi metrikler hesaplanmıştır.

3.5. Sonuçlar



Şekil 7- Tahmin edilen ile gerçek fiyatları gösteren grafik



Şekil 8- Train ve Test için Loss Epoch Grafiği

Mean Squared Error: 1507.7848049890201

Accuracy: 0.9824561403508771

Precision: 1.0

Recall: 0.9824561403508771

F1 Score: 0.9911504424778761

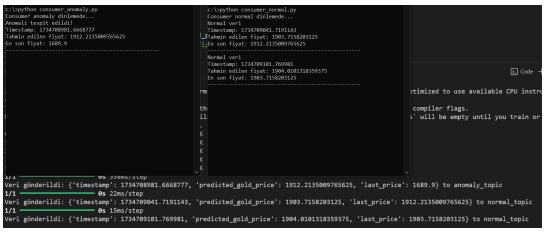
Geliştirilen LSTM modeli, zaman serisi verilerindeki trendleri ve dalgalanmaları öğrenerek yüksek doğrulukta tahminler yapmıştır. Bu mimari, özellikle uzun süreli bağımlılıkların önemli olduğu veriler için etkili bir şekilde çalışmış ve test sonuçları modelin başarılı bir şekilde genelleştirme yapabildiğini göstermiştir.

4. Kafka Yapılandırması

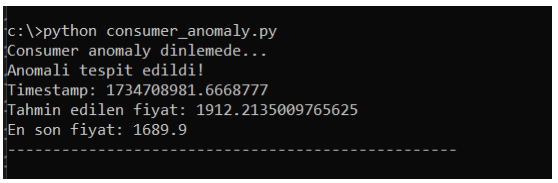
Zookeper ve Kafka kurulumu yapılmış ardından projede kullanılmak üzere anomali ve normal veriler için topicler oluşturulmuştur.

Producer, Kafka kullanarak altın fiyatı tahmin verilerini gönderen bir bileşendir. İlk olarak, KafkaProducer sınıfı ile producer yapılandırılmıştır. Verilerin JSON formatında gönderilmesi için value_serializer kullanılmıştır. Modelin tahmin yapabilmesi için önceden eğitilmiş olan Keras model dosyası yüklenmiş ve altın fiyatlarına dair geçmiş veriler bir CSV dosyasından okunmuştur. Bu verilerdeki sayısal olmayan değerler temizlenmiş ve eksik veriler düzeltilmiştir. Model, son altın fiyatı üzerinde tahmin yaparak, tahmin edilen fiyat ile gerçek fiyat arasındaki farkı kontrol etmek için bir anomali tespiti algoritması kullanır. Eğer fiyat farkı belirli bir eşiği aşarsa, veri anomali olarak kabul edilip "anomaly_topic" adında bir topic'e gönderilir. Aksi takdirde, normal veri "normal_topic" adlı topic'e iletilir. Producer, günlük olarak yeni veri göndermektedir, fakat çalışmada test edilirken dakikada bir göndermesi sağlanmıştır. Bu işlem, timestamp ile birlikte gerçekleşir ve veriler her iki topic'e yönlendirilir.

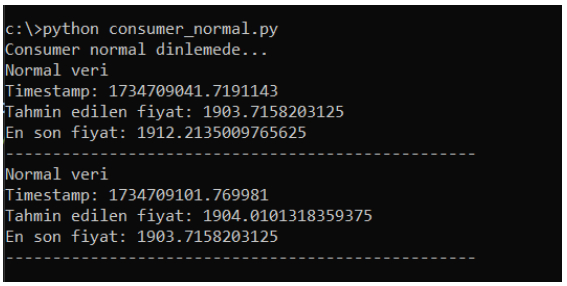
İki adet consumer oluşturulmuştur bir tanesi anomaly topic'e gelen verileri listelerken diğeri normal_topic'r gelen verileri listemektedir, bir tür filtreleme gibi çalışmaktadır. Bu consumerlar, JSON formatındaki veriyi deserialize ederek alır ve gelen mesajları işler. Eğer mesaj anomali verisi içeriyorsa, anomaly consumer bir uyarı mesajı yazdırır, aksi takdirde normal veri olarak işlenir normal consumer'da görüntülenir. Verinin içeriği, tahmin edilen altın fiyatı, son fiyat gibi bilgilerle birlikte ekrana yazdırılır. Bu işlem sürekli olarak çalışır ve gelen verileri anlık olarak takip eder.



Şekil 9- Kafka ile gerçek zamanlı veri gönderip alma



Şekil 10- Kafkadan gerçek zamanlı alınan verilerle anormal veri tespiti



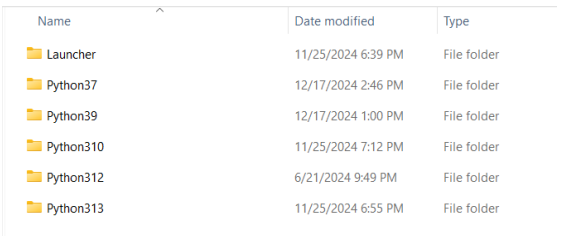
Şekil 11- Kafkadan gerçek zamanlı alınan verilerle normal veri tespiti

5. Spark Uygulaması

Spark'ın Python API'si olan PySpark, dağıtık veri işleme ve paralel hesaplama yetenekleri sunarak büyük veri setlerinin verimli bir şekilde işlenmesini sağlamaktadır.

Ancak, proje sırasında karşılaşılan sürüm uyumsuzlukları nedeniyle PySpark entegrasyonu gerçekleştirilememiştir. Python sürümleri ve TensorFlow gibi diğer kütüphanelerle yaşanan uyumluluk sorunları, Spark tabanlı veri işleme işlemlerini devre dışı bırakmıştır. Bu nedenle Spark entegrasyonu projede uygulanamamıştır.

Şekil 12'de gösterilen bütün pyhton sürümleri denenmiştir ancak bu denemeler sonucunda sürüm uyumsuzluğu devam ettiği için spark entegrasyonu yapılamamıştır.



Şekil 12- Pyspark ile denediğimiz python sürümleri

Kaynakça

- https://www.kaggle.com/datasets/farzadnekouei/gold-price-10-years-20132023
- https://www.veribilimiokulu.com/intellij-idea-ile-apache-spark-projesini-uzak-yarn-cluster-uzerinde-calistirmak-2/
- https://kafka.apache.org/downloads
- https://www.veribilimiokulu.com/windows-10-uzerine-kafka-kurmak/
- https://www.youtube.com/watch?v=t4m7lMfO9i4
- https://www.veribilimiokulu.com/windows-10-spark-2-kurulumu/
- https://zookeeper.apache.org/releases.html#download
- https://spark.apache.org/downloads.html