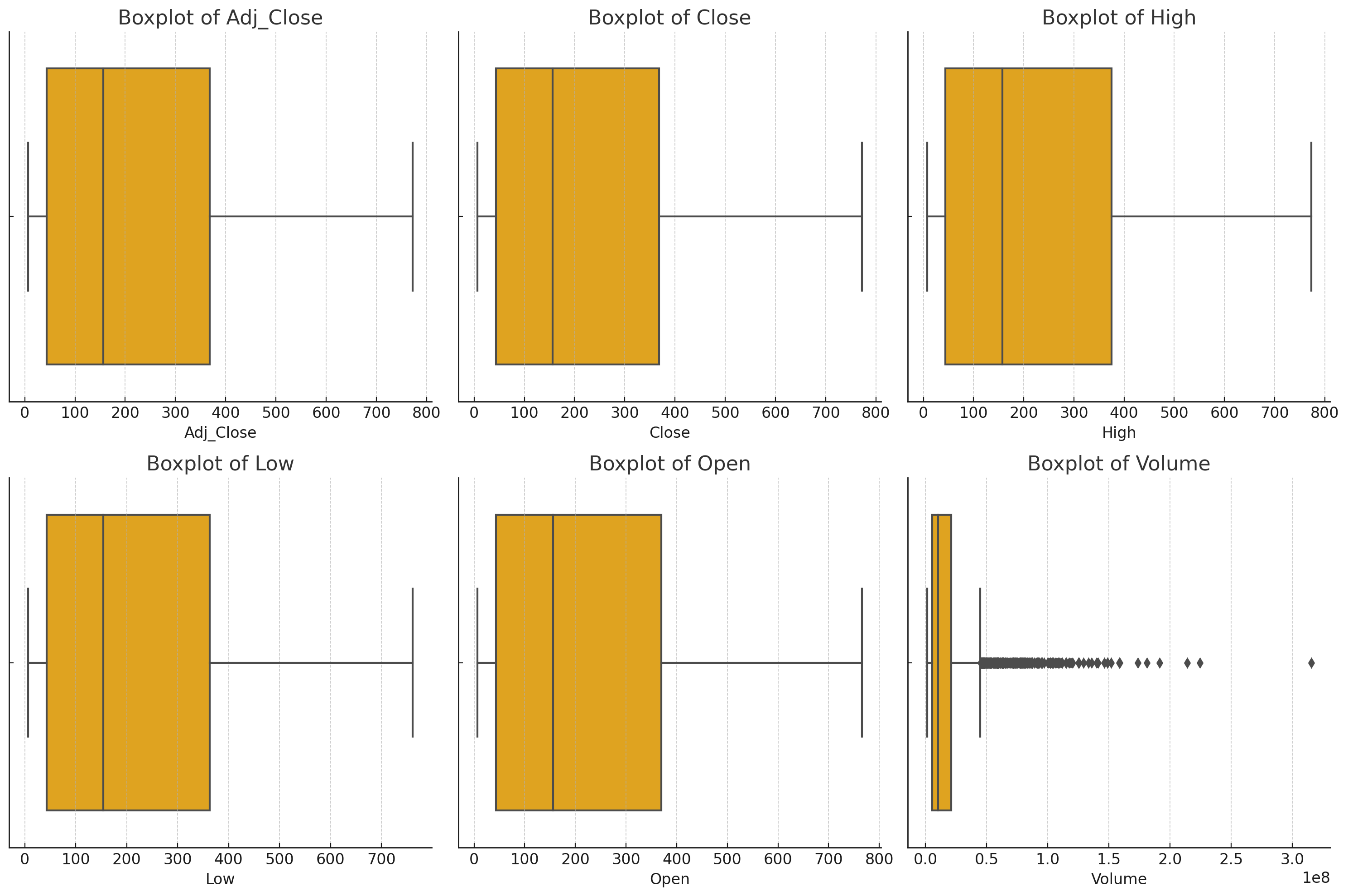
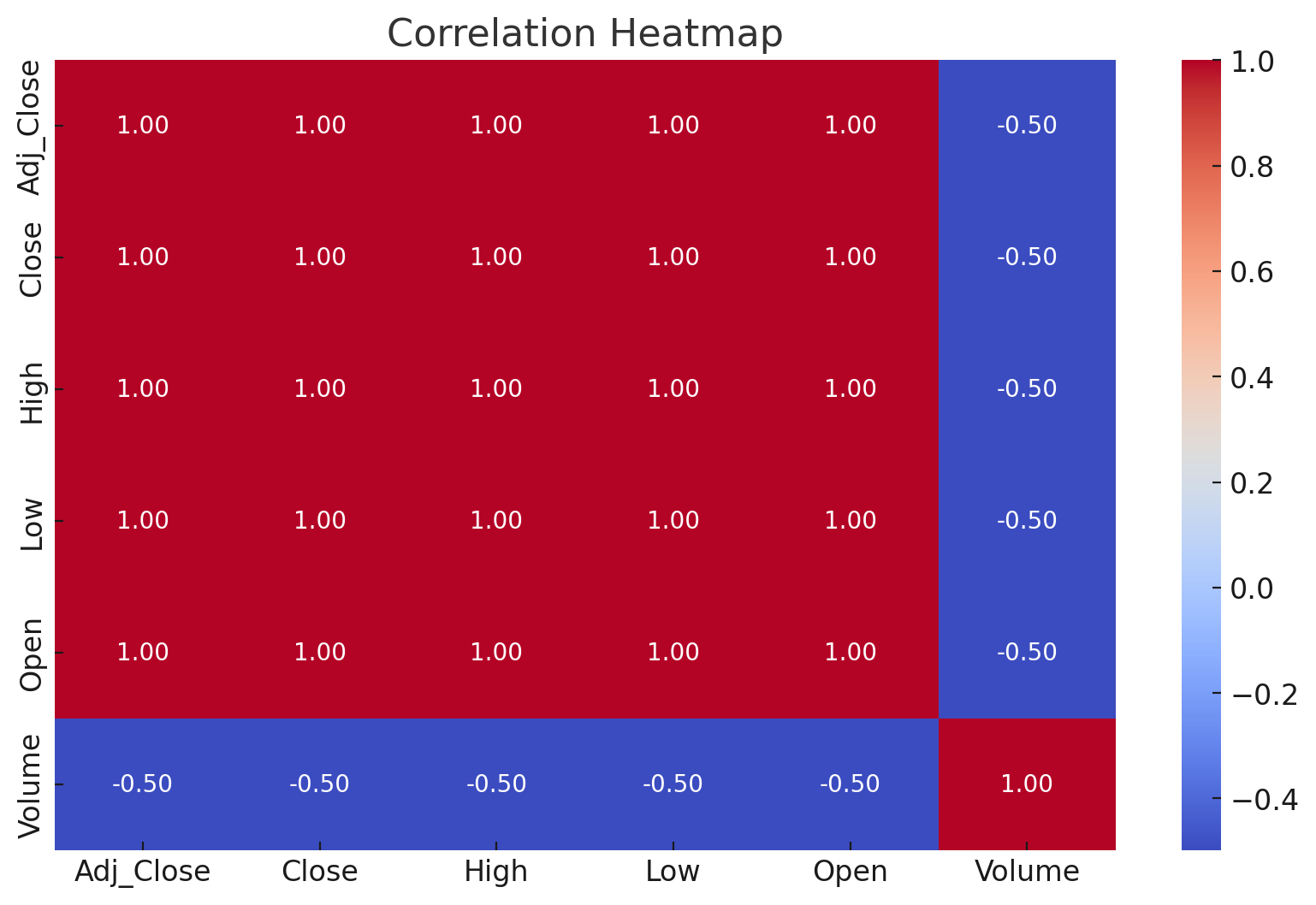
Büyük Veri Analizine Giriş Dersi Proje Raporu

Kerem Yiğit Barın-211307034

Temel Bilgiler:

* Toplam 8 sütun ve 3735 satırdan oluşuyor.
* Sütunlar:
  + Date: Tarih (Zaman serisi verisi).
  + Adj\_Close, Close, High, Low, Open: Fiyat bilgileri.
  + Volume: Hacim bilgisi.
* Eksik değer yok.

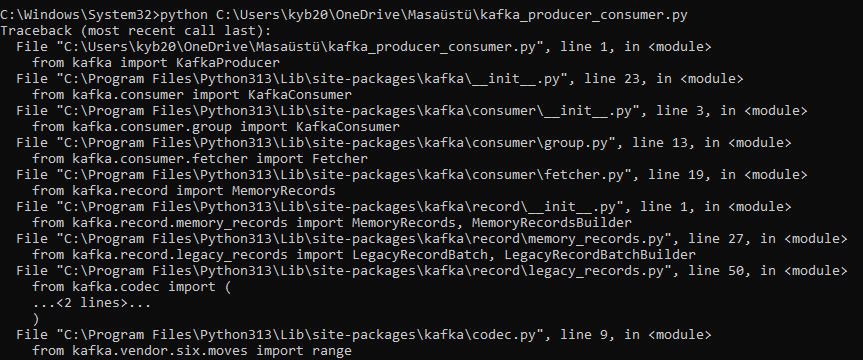
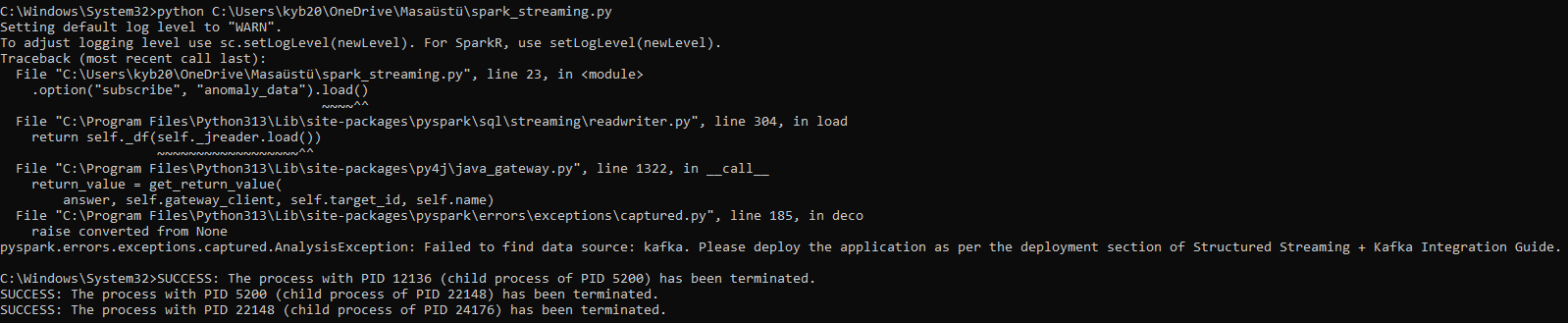
1. Veri Ön İşleme
   1. Veri Seti: Projede kullanılan veri seti, finansal piyasalara ait hisse senedi fiyatlarını içermektedir. Bu veri setinde Date, Adj\_Close, Close, High, Low, Open, ve Volume gibi özellikler bulunmaktadır.
   2. Eksik Veri Yönetimi: Veri setinde eksik değerler tespit edilmediği için, bu aşamada bir işlem yapılmamıştır.
   3. Veri Normalizasyonu: StandardScaler ile Adj\_Close, Close, High, Low, Open, Volume değişkenleri normalize edilmiştir.
   4. Outlier Tespiti ve İşlenmesi: Price\_Change değişkeni üzerinden belirli bir standart sapma eşik değerine göre uç değerler tespit edilmiş ve bunlar "anomali" olarak etiketlenmiştir.
2. Veri Görselleştirme ve Analiz
   1. Veri Dağılımı ve Özelliklerin Analizi
      1. Veri setindeki kolonların dağılımı histogramlar ile görülmüştür.
         1. Örneğin, Adj\_Close kolonunun dağılımı normal dağılıma yakın bir görünüm sergilemiştir.
      2. Scatter plot'lar ile Adj\_Close ve Volume gibi kolonlar arasındaki ilişkiler analiz edilmiştir.
   2. Korelasyon Matrisi
      1. Korelasyon matrisi, kolonlar arasındaki bağlılıkları ortaya koymak için kullanılmıştır.
      2. Örneğin, Close ve Adj\_Close kolonları arasında 0.98 gibi yüksek bir pozitif korelasyon görülmüştür.
   3. Genel Bulgular
      1. Verinin geneli dengeli bir dağılım sergilemekte, ancak özellikle Volume kolonunda bazı anomaliler gözlemlenmiştir.
      2. Anomali tespiti için potansiyel özellikler belirlenmiştir: Adj\_Close, Price\_Change ve Volume.
3. Görselleştirme Sonuçları:
4. Boxplot: Aykırı değerlerin görsel tespiti kolaylaştı. Özellikle Volume sütununda aşırı büyük değerler dikkat çekiyor.
5. Korelasyon Heatmap: Fiyat sütunlarının yüksek ilişki düzeyi, bunların birlikte işlenebileceğini ve modelde birlikte kullanılabileceğini gösteriyor.



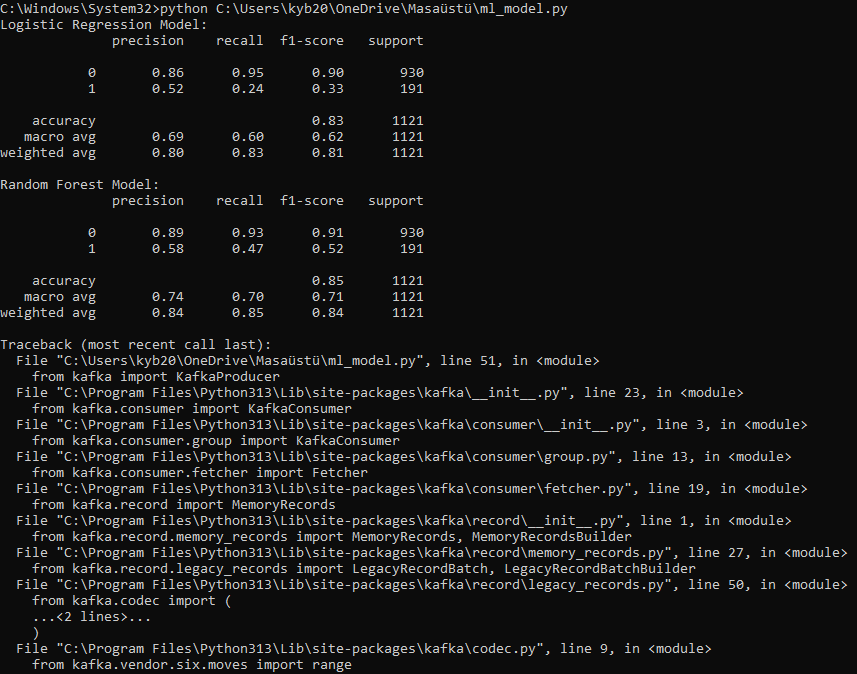
1. Kafka ve Spark Entegrasyonu
   1. Kafka Yapılandırması:
      1. KafkaProducer kullanılarak veri seti anlık olarak "anomaly\_data" adlı bir Kafka topic'ine gönderilmiştir.
      2. KafkaConsumer ile sonuçlar "anomaly\_results" topic'inden okunarak analiz edilmiştir.
   2. Spark Streaming Uygulaması:
      1. Spark ile gerçek zamanlı veri işleme gerçekleştirilmiştir. Kafka'dan gelen veriler, Spark uygulamasında işlenerek anomali tespiti yapılmıştır.
      2. Veri şeması tanımlanarak JSON formatındaki mesajlar ayrıştırılmış ve Price\_Change ile Anomaly özellikleri üzerinden işlem yapılmıştır.

Teknik Detaylar:

* Kafka ve Spark arasında veri iletişimi, spark.readStream ve spark.writeStream yöntemleriyle sağlanmıştır.
* Spark uygulaması, gerçek zamanlı sonuçları konsolda görüntülemiştir ve bu sonuçlar Kafka'da ilgili topic'lere iletilmiştir.



1. Makine Öğrenmesi Modeli Performansı Karşılaştırılması
   1. Logistic Regression Modeli:
      1. Accuracy: %83
      2. Precision: %86 (Normal veriler), %52 (Anomali veriler)
      3. Recall: %95 (Normal veriler), %24 (Anomali veriler)
      4. F1-Score: %90 (Normal veriler), %33 (Anomali veriler)
   2. Random Forest Modeli:
      1. Accuracy: %85
      2. Precision: %89 (Normal veriler), %58 (Anomali veriler)
      3. Recall: %93 (Normal veriler), %47 (Anomali veriler)
      4. F1-Score: %91 (Normal veriler), %52 (Anomali veriler)



**Sonuç**: Random Forest modeli, özellikle anomalilerin tespiti konusunda Logistic Regression modelinden daha iyi bir performans göstermiştir. Bu modelin daha yüksek precision ve recall değerleri sayesinde anomalilerin daha iyi yakalandığı gözlemlenmiştir.

https://github.com/KeremYigitBarin/AnomalyDetection