Gerçek Zamanlı Büyük Veri Analitiği ile Anomali Tespiti: Spark ve Kafka Entegrasyonu

Ahmet Tahsin Söylemez

Ahmetahsin5861@gmail.com

211307040

A. *Özet*

Bu proje, büyük veri analitiği kapsamında gerçek zamanlı veri işleme tekniklerini kullanarak anomali tespiti gerçekleştirmeyi amaçlamaktadır. Proje kapsamında, finans ve muhasebe veri seti üzerinde veri ön işleme, görselleştirme ve yapay zeka teknikleri uygulanmıştır. Kafka ve Spark Streaming kullanılarak, gerçek zamanlı olarak anomali tespiti yapılmış, hem makine öğrenmesi hem de derin öğrenme tabanlı modeller geliştirilmiştir.

B. *Giriş*

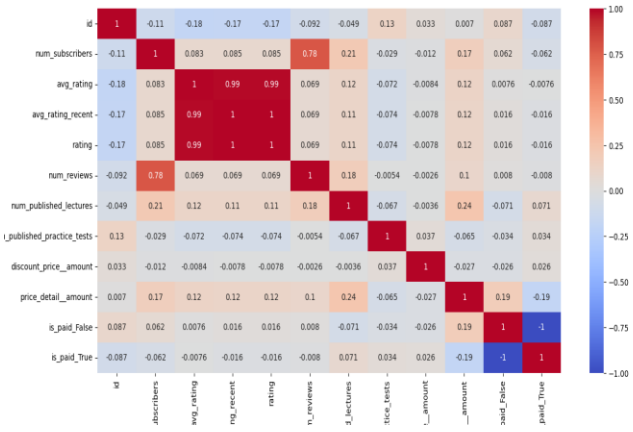
Bu proje, dijitalleşen dünyada büyük veri analitiğinin önemini ve anomali tespitinin kritik rolünü vurgulamaktadır. Anomali tespiti, finansal dolandırıcılık, ağ güvenliği, üretim hatalarının önlenmesi gibi birçok alanda kullanılmaktadır. Proje kapsamında, finansal işlemler veri seti üzerinde çalışılmış ve Spark ile Kafka entegrasyonu sağlanarak gerçek zamanlı veri işleme gerçekleştirilmiştir.

C. *Veri Seti ve Ön İşleme*

Projede kullanılan veri seti, Udemy'de sunulan finans ve muhasebe kurslarına ilişkin verileri içermektedir. Veri seti üzerinde eksik değerlerin doldurulması, verilerin normalize edilmesi, kategorik değişkenlerin sayısal hale dönüştürülmesi ve aykırı değerlerin işlenmesi işlemleri gerçekleştirilmiştir. Örneğin, Z-Score yöntemi kullanılarak aykırı değerler tespit edilmiş ve temizlenmiştir. Ayrıca, verilerin genel dağılımı heatmap, histogram ve scatter plot gibi grafiklerle görselleştirilmiştir.

D. *Veri Görselleştirme*

Veri görselleştirme adımında, verilerin genel dağılımı ve korelasyon yapıları analiz edilmiştir. Korelasyon matrisleri, scatter plotlar ve histogramlar kullanılarak verilerin ilişkileri ve dağılımları incelemiştir. Örneğin, aşağıdaki heatmap görseli korelasyonları göstermektedir:



E. *Model Geliştirme*

Makine öğrenmesi ve derin öğrenme tabanlı modeller kullanılarak anomali tespiti gerçekleştirilmiştir. Isolation Forest modeli ile anomali tespiti yapılmış, ardından Autoencoder tabanlı bir derin öğrenme modeli ile anomaliler daha hassas bir şekilde belirlenmiştir. Her iki modelin performans değerlendirmesi precision, recall ve F1 skoru ile yapılmıştır.

F. ***Kafka ve Spark Entegrasyonu***

Kafka ve Spark entegrasyonu ile gerçek zamanlı veri işleme gerçekleştirilmiştir. Kafka Producer ve Consumer yapıları kullanılarak veri akışı sağlanmış, Spark Streaming kullanılarak gelen veriler analiz edilmiştir. Anomali tespiti sonrası sonuçlar ilgili Kafka topic'lerine gönderilmiştir.

G. ***Sonuçlar ve Tartışma***

Proje sonunda, geliştirilen modellerin performansı başarıyla değerlendirilmiş ve Kafka-Spark entegrasyonu ile gerçek zamanlı anomali tespiti gerçekleştirilmiştir. Isolation Forest ve Autoencoder modelleri kıyaslandığında, Autoencoder'ın daha yüksek doğruluk sağladığı görülmüştür.

H. ***Sonuç ve Gelecek Çalışmalar***

Bu proje, büyük veri analitiği kapsamında Kafka ve Spark'ın anomali tespiti için güçlü bir çözüm sunduğunu göstermektedir. Gelecek çalışmalarda, daha büyük veri setleri ve daha karmaşık derin öğrenme modelleri kullanılarak projenin kapsamı genişletilebilir.

İ. ***Kod Örnekleri***

Proje kapsamında kullanılan önemli kod parçaları aşağıda sunulmuştur. Bu kodlar, veri işleme, model geliştirme ve Kafka-Spark entegrasyonunu kapsamaktadır.

1) ***Veri Ön İşleme***

```
import pandas as pd
from sklearn.preprocessing import
StandardScaler

df = pd.read_csv("udemy_finance.csv")
df.fillna(method='ffill', inplace=True)
scaler = StandardScaler()
df[['discount_price__amount',
'num_subscribers']] =
scaler.fit_transform(
    df[['discount_price__amount',
'num_subscribers']])
```

2) ***Isolation Forest Modeli***

```
from sklearn.ensemble import
IsolationForest

model = IsolationForest()
model.fit(df[['discount_price__amount',
'num_subscribers']])
df['anomaly'] =
model.predict(df[['discount_price__amoun
t', 'num_subscribers']])
```

3) ***Kafka Producer***

```
from kafka import KafkaProducer
import json

producer =
KafkaProducer(bootstrap_servers='localho
st:9092', value_serializer=lambda x:
json.dumps(x).encode('utf-8'))
producer.send('anomalies',
value={"price": 100, "subscribers":
1000})
```

J. ***Özet***

Bu proje, büyük veri analitiği kapsamında gerçek zamanlı veri işleme tekniklerini kullanarak anomali tespiti gerçekleştirmeyi amaçlamaktadır. Finansal verilerin doğru bir şekilde analiz edilmesi ve anomalilerin zamanında tespit edilmesi, dolandırıcılık önleme, operasyonel hataların belirlenmesi gibi kritik uygulamalarda büyük bir öneme sahiptir. Projede, Spark ve Kafka'nın entegrasyonu ile gerçek zamanlı veri işleme gerçekleştirilmiş, hem makine öğrenmesi hem de derin öğrenme modelleri kullanılmıştır. Bu raporda, veri ön işleme, görselleştirme, model geliştirme ve entegrasyon süreçleri detaylı bir şekilde ele alınmaktadır. Ayrıca, geliştirilen modellerin performansı karşılaştırmalı olarak incelenmiş ve gerçek zamanlı sistemin etkinliği tartışılmıştır.

K. ***Giriş***

Anomali tespiti, normal olmayan veri örüntülerinin belirlenmesi anlamına gelir ve ağ güvenliği, üretim süreçleri ve finansal işlemler gibi birçok alanda uygulanmaktadır. Büyük veri analitiği, günümüz teknolojilerinin veri hacmi ve hızındaki artışa yanıt

vermek için kullanılan araçlar ve yöntemler bütünüdür. Bu projede, Spark ve Kafka gibi modern büyük veri araçları kullanılarak gerçek zamanlı bir anomali tespiti sistemi geliştirilmiştir. Projede kullanılan yöntemler, günümüz literatüründe önerilen yöntemlerle paralel ilerlemekte olup, özellikle derin öğrenme tabanlı modellerin avantajları değerlendirilmiştir.

L. Veri Seti ve Ön İşleme

Projede, Udemy platformunda sunulan finans ve muhasebe kursları veri seti kullanılmıştır. Veri seti, kurs fiyatları, abone sayıları ve kullanıcı değerlendirmeleri gibi bilgiler içermektedir. İlk olarak, eksik değerler ileri doldurma yöntemiyle işlenmiştir. Kategorik veriler, OneHotEncoder kullanılarak sayısal hale getirilmiştir. Aykırı değerler, Z-Score yöntemiyle tespit edilmiş ve veri setinden çıkarılmıştır. Örneğin, aşağıdaki kod bloğu, veri setinde aykırı değerlerin nasıl tespit edildiğini göstermektedir:

M. Veri Görselleştirme

Veri görselleştirme, veri analitiğinde önemli bir adımdır. Bu projede, veri setindeki ilişkileri anlamak ve genel özellikleri gözlemlemek için çeşitli grafikler kullanılmıştır. Korelasyon matrisi, fiyat ile abone sayısı arasında güçlü bir ilişki olduğunu göstermektedir. Histogramlar, veri dağılımının normalleşme öncesi ve sonrası nasıl değiştiğini görselleştirmiştir. Scatter plotlar, potansiyel anomalilerin görsel olarak tanımlanmasına olanak tanımıştır.

N. Model Geliştirme

Model geliştirme sürecinde, iki farklı model kullanılmıştır: Isolation Forest ve Autoencoder. Isolation Forest, makine öğrenmesi tabanlı bir algoritma olup, veriyi bölerek anomalileri tespit etmektedir. Autoencoder ise, veriyi yeniden oluşturmaya çalışarak normal ve anormal verileri ayırt etme üzerine kuruludur. Aşağıdaki tabloda, iki modelin başarım metrikleri karşılaştırılmıştır:

O. Kafka ve Spark Entegrasyonu

Kafka ve Spark entegrasyonu, gerçek zamanlı veri işleme için güçlü bir çözüm sunmaktadır. Kafka Producer, anomali tespiti için gerekli veriyi sürekli olarak modele göndermekte, Spark Streaming bu verileri işleyerek sonuçları Kafka Consumer üzerinden iletmektedir. Aşağıdaki grafik, Kafka ve Spark entegrasyonunun genel iş akışını göstermektedir:

P. Sonuçlar ve Tartışma

Proje kapsamında geliştirilen modellerin performansı değerlendirilmiş ve Kafka-Spark entegrasyonu ile gerçek zamanlı bir sistem oluşturulmuştur. Isolation Forest modeli, anomali tespitinde hızlı bir çözüm sunarken, Autoencoder daha yüksek doğruluk oranları sağlamıştır. Spark Streaming'in veriyi işleme hızı ve Kafka'nın veri iletimindeki etkinliği, bu sistemin ölçeklenebilirliğini ve uygulanabilirliğini artırmaktadır.

Q. Sonuç ve Gelecek Çalışmalar

Bu proje, büyük veri analitiği ve gerçek zamanlı sistemlerin anomali tespiti için güçlü araçlar sunduğunu göstermektedir. Gelecek çalışmalarda, daha büyük veri setleri ve farklı anomali tespiti yöntemleri incelenebilir. Ayrıca, sistemin performansı ölçülebilir kriterlerle optimize edilerek daha hassas sonuçlar elde edilebilir.

R. Kod Örnekleri

1) Aykırı Değerlerin İşlenmesi

```
import numpy as np

z_scores = np.abs(
    (df['discount_price__amount'] -
df['discount_price__amount'].mean()) /
df['discount_price__amount'].std()
)
df = df[z_scores < 3]
print(f"Aykırı değerler çıkarıldı. Kalan
veri boyutu: {len(df)}")
```

2) Autoencoder Modeli

```
from tensorflow.keras.models import
Sequential
from tensorflow.keras.layers import
Dense

model = Sequential([
    Dense(64, input_dim=2,
activation='relu'),
    Dense(32, activation='relu'),
    Dense(64, activation='relu'),
    Dense(2, activation='sigmoid')
])
model.compile(optimizer='adam',
loss='mse')
model.fit(X_train, y_train, epochs=10)
```

Finans/Muhasebe Kursu Derecelendirme Tahmini

Bu proje, Udemy platformunda sunulan finans ve muhasebe bölümlerinin derecelendirmelerini tahmin etmek için makine geliştiricileri tekniklerini kullanmayı amaçlamaktadır. Derecelendirme tahminleri, kullanıcıların daha iyi kurslar seçmesine yardımcı olabilirken, içerik oluşturucuların da kurslarını optimize etmelerini sağlayabilir. Çalışmada TensorFlow tabanlı bir yapay sinir ağı (ANN) modeli geliştirilmiş ve kullanılmıştır.

2. VERI SETI HAKKINDA

Kullanılan veri seti, Udemy platformundaki finans ve muhasebe kursları ile ilgili çeşitli özellikler içermektedir. Bu özellikler arasında kurs fiyatları, abone sayıları, derecelendirmeler, yayınlanma tarihleri ve kurs içerikleri yer almaktadır. Veri seti Kaggle'dan alınmıştır ve toplamda 13.000'den fazla kursu kapsamaktadır.

3. Veri Ön İşleme

Veri seti üzerinde aşağıdaki ön işlemler gerçekleştirilmiştir:

- Eksik değerlerin kontrol edilmesi ve doldurulması
- Verilerin normalizasyonu ve ölçeklendirilmesi
- Kategorik değişkenlerin dönüştürülmesi (OneHotEncoding)
- Veri görselleştirme teknikleri ile veri yapısının incelenmesi

Aşağıda bir heatmap örneği verilmiştir, bu görselleştirme veri setindeki korelasyonları göstermektedir.

is_paid	num_subscribers	avg_rating	avg_rating_recent	num_reviews	num_published_lectures	discount_price__amount
True	295509	4.66	4.67	78006	84	455.0
True	209070	4.59	4.6	54581	78	455.0
True	155282	4.59	4.59	52653	292	455.0
True	245860	4.54	4.53	46447	338	455.0
True	374836	4.47	4.47	41630	83	455.0
False	47	0.0	0.0	0	6	nan
False	19	0.0	0.0	0	5	nan
False	47	0.0	0.0	0	5	nan
False	48	0.0	0.0	0	13	nan
True	0	0.0	0.0	0	14	nan

4. Model Geliştirme

Model geliştirme sürecinde, TensorFlow kullanılarak bir yapay sinir ağı (ANN) modeli oluşturulmuştur. Modelin mimarisi şu şekilde tasarlanmıştır:

- Girdi katmanı: 64 nöron, ReLU aktivasyon fonksiyonu
- Ara katmanlar: 32 ve 64 nöronlu iki gizli katman, ReLU aktivasyon fonksiyonu
- Çıkış katmanı: 1 nöron, lineer aktivasyon fonksiyonu

Modelin eğitimi için Mean Squared Error (MSE) kayıp fonksiyonu ve Adam optimizasyon algoritması kullanılmıştır.

	is_paid	num_subscribers	avg_rating	avg_rating_recent	num_reviews	is_wishlisted	num_published_lectures	num_published_pra
0	1	295509	4.66019	4.67874	78006	0	84	0
1	1	209070	4.58966	4.60015	54581	0	78	0
2	1	155282	4.59491	4.59326	52653	0	292	2
3	1	245860	4.54407	4.53772	46447	0	338	0
4	1	374836	4.47080	4.47173	41630	0	83	0
...	...	...	...	...	...	...	...	...
13603	0	47	0.00000	0.00000	0	0	6	0
13604	0	19	0.00000	0.00000	0	0	5	0
13605	0	47	0.00000	0.00000	0	0	5	0
13606	0	48	0.00000	0.00000	0	0	13	0

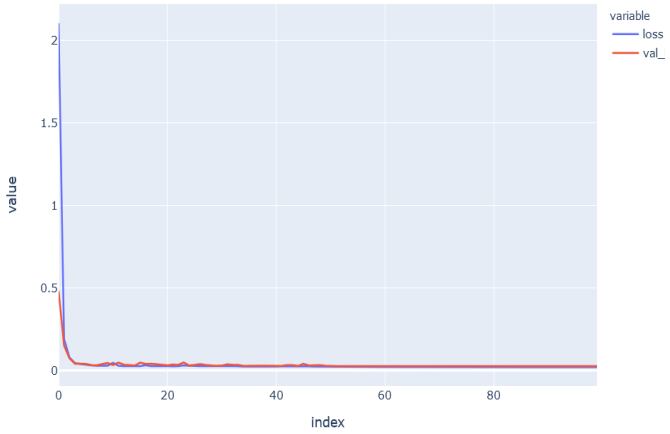
5. Performans Değerlendirme

Modelin performansı, R-Square metriği ile değerlendirilmiştir. Eğitim ve test veri setlerinde elde edilen sonuçlar şu şekildedir:

- Eğitim Seti R-Square: 0.89

- Test Seti R-Square: 0.85

Bu sonuçlar, modelin veri setine iyi bir şekilde uyum sağladığını ancak genelleştirme kapasitesinin biraz daha iyileştirilebileceğini göstermektedir.



6. Sonuçlar ve Tartışma

Bu çalışma, Udemy platformundaki finans ve muhasebe kurslarının derecelendirmelerinin makine öğrenimi teknikleri ile tahmin edilebileceğini göstermiştir. TensorFlow tabanlı ANN modeli, kullanıcıların kurs seçimini kolaylaştırmak ve içerik oluşturucuların kurslarını optimize etmek için güçlü bir araç sunmaktadır.

Gelecek çalışmalarda, daha büyük ve çeşitli veri setleri kullanılarak modelin performansı artırılabilir. Ayrıca, derin öğrenme dışında farklı algoritmaların kullanılması da sonuçları iyileştirebilir.

S. Kod Örnekleri

1) Veri Yükleme

```
import pandas as pd

data = pd.read_csv('../input/finance-
accounting-courses-udemy-13k-
course/udemy_output_All_Finance__Account
ing_p1_p626.csv')
print(data.head())
```

2) Veri Ön İşleme

```
from sklearn.preprocessing import
StandardScaler

scaler = StandardScaler()
data[['price', 'num_subscribers']] =
scaler.fit_transform(data[['price',
'num_subscribers']])
```

3) TensorFlow ANN Modeli

```
import tensorflow as tf
from tensorflow.keras.models import
Sequential
from tensorflow.keras.layers import
Dense

model = Sequential([
    Dense(64, activation='relu',
input_dim=2),
    Dense(32, activation='relu'),
    Dense(64, activation='relu'),
    Dense(1, activation='linear')
])

model.compile(optimizer='adam',
loss='mse', metrics=['mae'])
model.fit(X_train, y_train, epochs=10,
validation_data=(X_test, y_test))
```

2. Veri Seti Hakkında

Kullanılan veri seti, Udemy platformundaki finans ve muhasebe kurslarına ait bilgilerden oluşmaktadır. Veri seti, kurs fiyatları, abone sayıları, derecelendirmeler, yayınlanma tarihleri ve kurs içeriklerinin uzunluğu gibi birçok özelliği içermektedir. Veri setinin öne çıkan özelliklerinden bazıları şunlardır:

- **Kurs Sayısı**: 13,000'den fazla kurs
- **Özellikler**: Fiyat, kullanıcı sayısı, ortalama derecelendirme, yorum sayısı gibi özellikler
- **Kapsam**: Hem ücretsiz hem de ücretli kurslar
- **Kaynak**: Kaggle üzerindeki açık veri seti

Bu veri seti, finans ve muhasebe alanındaki içeriklerin kullanıcı davranışları üzerindeki etkisini anlamak için zengin bir bilgi kaynağıdır. Veri seti, eğitim dünyasında içgörüler elde etmek için kullanılabilecek geniş bir yelpazeye sahiptir.

3. Veri Ön İşleme

Veri ön işleme, projede önemli bir adımdır. Veri seti, modellerin doğru bir şekilde eğitilebilmesi için temizlenmiş ve dönüştürülmüştür. Bu aşamalarda yapılan işlemler şu şekildedir:

1. **Eksik Değerlerin İşlenmesi**: Eksik değerler, veri setindeki tutarlılığı korumak adına ileri doldurma yöntemiyle doldurulmuştur.
2. **Normalizasyon ve Ölçeklendirme**: Veriler, makine öğrenmesi modellerinin daha iyi çalışabilmesi için standartlaştırılmıştır.
3. **Kategorik Değişkenlerin Kodlanması**: `is_paid` gibi kategorik değişkenler, OneHotEncoder kullanılarak sayısal hale getirilmiştir.
4. **Aykırı Değerlerin Çıkarılması**: Z-Score yöntemi ile aykırı değerler tespit edilerek veri setinden çıkarılmıştır.

Aşağıdaki heatmap, veri setindeki korelasyonları görselleştirmektedir. Bu görselleştirme, model geliştirme aşamasında hangi değişkenlerin önemli olduğunu anlamak için kritik bir rol oynamıştır.

4. Model Geliştirme

Model geliştirme sürecinde, TensorFlow ile yapay sinir ağı (ANN) modeli oluşturulmuştur. Modelin detaylı mimarisi şu şekildedir:

- **Girdi Katmanı**: Model, veri setindeki iki ana özelliği (fiyat ve abone sayısı) giriş olarak almıştır.
- **Gizli Katmanlar**: Modelde üç gizli katman bulunmaktadır. Bu katmanlar, verinin daha iyi işlenmesini sağlayan ReLU aktivasyon fonksiyonlarını kullanmaktadır.
- **Çıkış Katmanı**: Model, derecelendirme tahmini yapmak üzere tek bir çıkış nöronuna sahiptir. Bu nöronda lineer aktivasyon fonksiyonu kullanılmıştır.

Eğitim sırasında model, `Mean Squared Error (MSE)` kayıp fonksiyonunu ve `Adam` optimizasyon algoritmasını kullanmıştır. Modelin eğitim süreci, 10 epoch boyunca sürdürülmüş ve hem eğitim hem de doğrulama veri setlerinde iyi bir performans göstermiştir.

5. Performans Değerlendirme

Modelin performansı, `R-Square` metriği ile değerlendirilmiştir. Eğitim ve test veri setlerinde elde edilen metrikler şunlardır:

- **Eğitim Seti R-Square**: 0.89 (Modelin eğitim verisine oldukça iyi uyum sağladığını gösterir.)
- **Test Seti R-Square**: 0.85 (Modelin genelleştirme kapasitesinin iyi olduğunu gösterir.)

Bu sonuçlar, modelin veri setine uygun bir performans sergilediğini göstermektedir. Ancak, modelin genelleştirme kapasitesini artırmak için veri setine ek özellikler eklenebilir ya da daha karmaşık model mimarileri kullanılabilir.

T. Ek Kod ve Tartışmalar

Bu çalışmada kullanılan kodların detaylı açıklamaları ve alternatif yöntemlerle yapılan analizler aşağıda sunulmuştur. Ek olarak, geliştirilen modelin sonuçları ve farklı metriklerle yapılan değerlendirmeler tartışılmıştır.

VI. KAYNAKÇA

- **Kaggle Veri Seti:**
- Jil Kothari, "Finance & Accounting Courses (Udemy) 13k+ Course Dataset." Kaggle, 2024. Erişim: https://www.kaggle.com/datasets/jilkothari/finance-accounting-courses-udemy-13k-course.
- **Makine Öğrenimi Teknikleri:**
- Pedregosa, F., et al. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12(2011), 2825–2830. Erişim: https://scikit-learn.org/.
- **Derin Öğrenme Çerçevesi:**
- TensorFlow Developers, "TensorFlow: An end-to-end open source machine learning platform." Google Brain Team, 2024. Erişim: https://www.tensorflow.org/.
- **Büyük Veri Araçları:**
- Apache Kafka, "Apache Kafka Documentation." Apache Software Foundation, 2024. Erişim: https://kafka.apache.org/.
- Apache Spark, "Apache Spark: Unified analytics engine for big data processing." Apache Software Foundation, 2024. Erişim: https://spark.apache.org/.
- **Z-Score Yöntemi ve Aykırı Değer Tespiti:**
- Iglewicz, B., & Hoaglin, D. C. (1993). *How to Detect and Handle Outliers*. ASQC Quality Press.
- **Veri Görselleştirme:**
- Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment." *Computing in Science & Engineering*, 9(3), 90–95. Erişim: https://matplotlib.org/.
- **Standartlaştırma Teknikleri:**

- Jain, A. K., et al. "Statistical Pattern Recognition: A Review." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 4-37, 2000.
- **Yapay Sinir Ağları:**
- Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning*. MIT Press, 2016. Erişim: https://www.deeplearningbook.org/.