Yüksek Düzey Programlama Proje Ödevi

Predict Future Sales

Giriş

Bu çalışma, zaman serisi veri seti üzerinde satış tahmini yapmak amacıyla çeşitli makine öğrenmesi ve derin öğrenme modellerini uygulamayı hedeflemektedir. Çalışmanın temel amacı, geçmiş satış verilerini analiz ederek gelecekteki satış miktarlarını tahmin edecek bir model geliştirmektir. Proje kapsamında veri ön işleme, regresyon teknikleri ve CNN-LSTM tabanlı bir modelleme süreci gerçekleştirilmiştir.

Gelişme

1. Veri Hazırlama ve Keşifsel Veri Analizi

Projenin ilk adımı olarak, sales_train.csv, test.csv, items.csv, item_categories.csv ve shops.csv dosyaları yüklenmiştir. Verilerdeki yapının incelenmesi için aşağıdaki adımlar uygulanmıştır:

head() ile veri setinin ilk birkaç satırı gözlemlenmiş ve veri türleri doğrulanmıştır.

Kategorik ve sayısal değişkenlerin analizi yapılmıştır.

Aşağıdaki kod ile veri setleri yüklenmiş ve incelenmiştir:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


# Veri setlerini yükleme

train = pd.read_csv('sales_train.csv')

test = pd.read_csv('test.csv')

items = pd.read_csv('items.csv')

items_cat = pd.read_csv('item_categories.csv')

shops = pd.read_csv('shops.csv')
```

```
# İlk 5 satırı görüntüleme
print(train.head())
print(test.head())
```

Analiz:

Yukarıdaki kod ile veri setlerinin yapısı incelenmiş ve aşağıdaki adımlar uygulanmıştır:

Tarih formatları dönüştürülmüştür.

Eksik değerler kontrol edilmiş, temizlenmiş veya doldurulmuştur.

2. Veri Ön İşleme

Modelin etkili bir şekilde öğrenebilmesi için veri ölçeklendirme işlemi yapılmıştır:

```
from sklearn.preprocessing import StandardScaler

# Özellik ölçekleme
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

3. Modelleme

CNN-LSTM Modeli Oluşturma

Aşağıdaki kod ile CNN-LSTM modeli oluşturulmuş ve eğitilmiştir:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, LSTM, Dense

# CNN-LSTM modeli
model = Sequential([
    Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(X_train.shape[1], X_train.shape[2])),
```

```
    MaxPooling1D(pool_size=2),
    LSTM(50, activation='relu'),
    Dense(1)
])

model.compile(optimizer='adam', loss='mse')
model.fit(X_train_scaled, y_train, epochs=10, batch_size=32,
validation_data=(X_test_scaled, y_test))
```

Model Tahminleri ve Performans Değerlendirme

Modelin tahmin sonuçları ters ölçeklenerek gerçek değerlere dönüştürülmüş ve hata metrikleri hesaplanmıştır:

```
# MAE hesaplama fonksiyonu
def mae_train_test(model, model_name, X_train, y_train, X_test, y_test, scaler):
    train_preds = scaler.inverse_transform(model.predict(X_train))
    test_preds = scaler.inverse_transform(model.predict(X_test))

    train_mae = np.mean(np.abs(train_preds.reshape(-1) - y_train))
    test_mae = np.mean(np.abs(test_preds.reshape(-1) - y_test))

    print(f"Model: {model_name}")
    print(f"Train MAE: {train_mae}")
    print(f"Test MAE: {test_mae}")

# MAE değerlendirme
mae_train_test(model, "CNN-LSTM", X_train_scaled, y_train, X_test_scaled, y_test,
scaler)
```

Sonuçlar:

Eğitim ve test veri setleri üzerinde tahminler yapılmış ve MAE değerleri raporlanmıştır.

Sonuç

Bu çalışma, satış tahmini yapmak için CNN-LSTM modeli gibi modern derin öğrenme yaklaşımlarını uygulamıştır. Çalışmanın başlıca bulguları şunlardır:

Veri ön işleme ve ölçekleme adımları model başarısında önemli rol oynamıştır.

CNN-LSTM modeli, zaman serisi verilerinde trend ve kalıpları öğrenmede etkili olmuştur.

Model, eğitim ve test veri setlerinde düşük MAE değerleri elde ederek güçlü bir tahmin performansı sergilemiştir.