Aylık Satış Tahmini (Linear Regression ile)

Ayşegül SARIKAYA – 202113171059

Kaggle Linki: https://www.kaggle.com/code/sarkayas/predict-future-salas-linear-regression

1. Giriş

Proje Amacı:
Bu projede, geçmiş satış verilerine dayanarak gelecekteki satış miktarlarını tahmin etmek için bir Lineer Regresyon modeli geliştirilmiştir. Verilerdeki aykırı değerler ve veri temizliği gibi aşamalar dikkate alınarak modelin performansı değerlendirilmiştir.

Uygulama Alanları:

- Stok yönetimi
- Satış tahmini
- Gelir optimizasyonu

2. Kullanılan Veri Seti

Veri Kaynağı:
Competitive Data Science Predict Future Sales

Özellikler (Features):

- **date_block_num:** Ay numarası (zaman serisi).
- **shop_id:** Mağaza kimliği.
- **item_id:** Ürün kimliği.
- **item_price:** Ürün fiyatı.
- **item_cnt_day:** Günlük satış miktarı.

Hedef Değişken:

- **item_cnt_month:** Aylık toplam satış miktarı.

Gerekli kütüphanelerin kurulumu:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

Veriyi yükleme ve gözlemleme:

```
# Veriyi yükleyelim
sales_data = pd.read_csv('/kaggle/input/competitive-data-science-predict-future-sales/sales_train.csv')
print(sales_data.head())
```

	date	date_block_num	shop_id	item_id	item_price	item_cnt_day
0	02.01.2013	0	59	22154	999.00	1.0
1	03.01.2013	0	25	2552	899.00	1.0
2	05.01.2013	0	25	2552	899.00	-1.0
3	06.01.2013	0	25	2554	1709.05	1.0
4	15.01.2013	0	25	2555	1099.00	1.0

3. Veri Ön İşleme

a. Eksik ve Hatalı Değerlerin İncelenmesi:

Negatif fiyatlar ve satış değerleri ayıklandı.

Tarihler datetime formatına dönüştürüldü.

b. Aykırı Değerlerin Çıkarılması:

Çok büyük veya anlamlı olmayan değerler kaldırıldı.

c. Veri Özet Bilgileri:

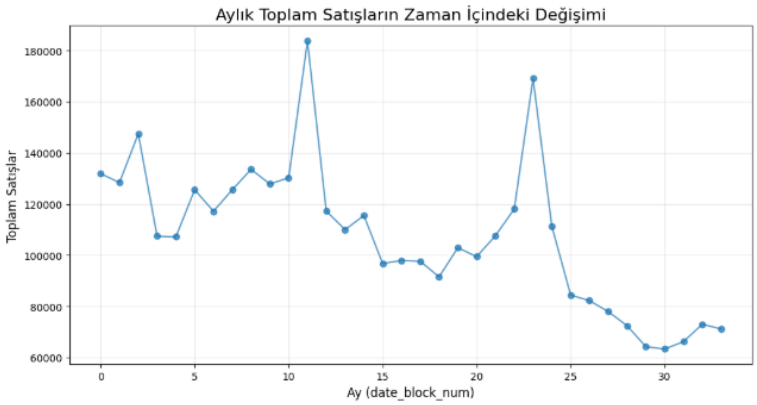
Toplam 2.9 milyon veri noktası temizlendikten sonra işlemeye alındı.

```python
# Eksik değer kontrolü
print(sales_data.isnull().sum())

# Veri türlerini ve temel istatistikleri inceleme
print(sales_data.info())
print(sales_data.describe())

# Aykırı değerleri bulma (örn. Negatif satışlar, çok büyük değerler)
sales_data = sales_data[sales_data['item_price'] > 0]
sales_data = sales_data[sales_data['item_cnt_day'] > 0]

# Tarihleri datetime formatına çevirme
sales_data['date'] = pd.to_datetime(sales_data['date'], format='%d.%m.%Y')
```

```
date              0
date_block_num    0
shop_id           0
item_id           0
item_price        0
item_cnt_day      0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2935849 entries, 0 to 2935848
Data columns (total 6 columns):
 #   Column          Dtype
---  ------          -----
 0   date            object
 1   date_block_num  int64
 2   shop_id         int64
 3   item_id         int64
 4   item_price      float64
 5   item_cnt_day    float64
dtypes: float64(2), int64(3), object(1)
memory usage: 134.4+ MB
None
       date_block_num       shop_id       item_id    item_price  item_cnt_day
count    2.935849e+06  2.935849e+06  2.935849e+06  2.935849e+06  2.935849e+06
mean     1.456991e+01  3.300173e+01  1.019723e+04  8.908532e+02  1.242641e+00
std      9.422988e+00  1.622697e+01  6.324297e+03  1.729800e+03  2.618834e+00
min      0.000000e+00  0.000000e+00  0.000000e+00 -1.000000e+00 -2.200000e+01
25%      7.000000e+00  2.200000e+01  4.476000e+03  2.490000e+02  1.000000e+00
50%      1.400000e+01  3.100000e+01  9.343000e+03  3.990000e+02  1.000000e+00
75%      2.300000e+01  4.700000e+01  1.568400e+04  9.990000e+02  1.000000e+00
max      3.300000e+01  5.900000e+01  2.216900e+04  3.079800e+05  2.169000e+03
```

4. Görselleştirme

Aylık Toplam Satışlar:
Aylık satışların zaman içindeki değişimi incelendi ve satışların belirgin bir sezonluk eğilim gösterdiği gözlemlendi.



5. Özellik Mühendisliği

- **Aylık Toplam Satışlar:**
  Mağaza ve ürün bazında aylık toplam satışlar hesaplandı.
- **Yeni Değişkenler:**
  item_cnt_month isimli hedef değişken oluşturuldu.

```
# Aylık toplam satışları hesaplayalım
monthly_data = sales_data.groupby(['date_block_num', 'shop_id', 'item_i
d']).agg({
    'item_cnt_day': 'sum'
}).reset_index()

# Kolonları düzenleme
monthly_data.rename(columns={'item_cnt_day': 'item_cnt_month'}, inplace=Tru
e)
print(monthly_data.head())
```

	date_block_num	shop_id	item_id	item_cnt_month
0	0	0	32	6.0
1	0	0	33	3.0
2	0	0	35	1.0
3	0	0	43	1.0
4	0	0	51	2.0

6. Model Eğitimi ve Testi

1. **Model Seçimi:**
   Lineer Regresyon modeli kullanıldı.
2. **Veri Bölünmesi:**
   - Eğitim verisi: %80
   - Test verisi: %20
3. **Modelin Eğitilmesi:**
   Eğitim verisi üzerinde model eğitildi ve test verisi ile değerlendirildi.

```
# Modeli eğitme
model = LinearRegression()
model.fit(X_train, y_train)

# Test seti tahmini
y_pred = model.predict(X_test)

# Performans metriği
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
```

Mean Squared Error: 79.69708835181014

7. Model Performansı

Performans Metrikleri:

```
Mean Absolute Error (MAE): 1.76
Mean Squared Error (MSE): 79.70
Root Mean Squared Error (RMSE): 8.93
R² Score: 0.00
```

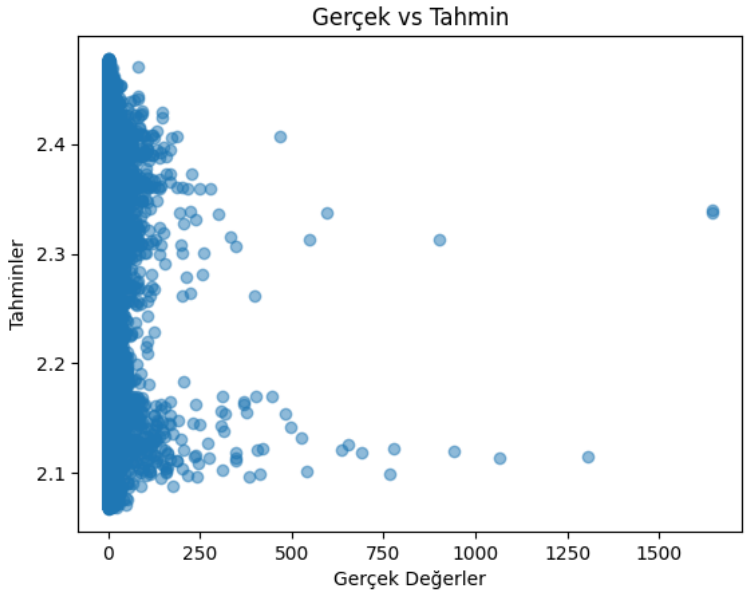
Analiz:
Model, basit bir doğrusal ilişkiyi yakalamakta zorlanmıştır. Verilerin karmaşıklığı ve sezonsallık etkileri doğrusal modellerin performansını sınırlamış olabilir.

Hata Dağılımı:

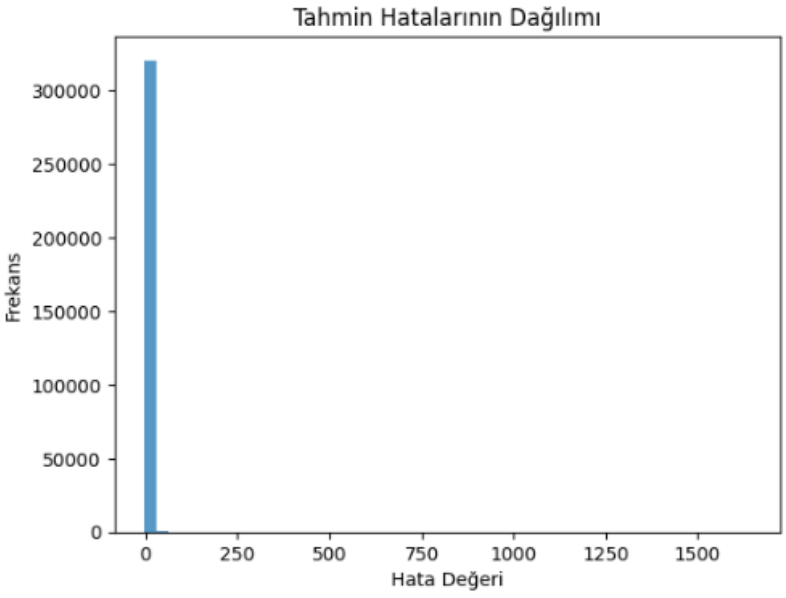
- Hataların büyük bir kısmı düşük frekansta, ancak bazı uç hatalar gözlemlenmiştir.

- Görselleştirme: Gerçek ve tahmin edilen değerler ile hata histogramı analiz edildi.

```
plt.scatter(y_test, y_pred, alpha=0.5)
plt.xlabel("Gerçek Değerler")
plt.ylabel("Tahminler")
plt.title("Gerçek vs Tahmin")
plt.show()
```




```
errors = y_test - y_pred
plt.hist(errors, bins=50, alpha=0.75)
plt.xlabel("Hata Değeri")
plt.ylabel("Frekans")
plt.title("Tahmin Hatalarının Dağılımı")
plt.show()
```



8. Çıkarımlar ve Gelecekteki Çalışmalar

1. **Çıkarımlar:**
    - Aylık toplam satış tahmininde, verilerin doğrusal olmayan yapısı model performansını etkiledi.
    - Daha karmaşık modeller (ör. XGBoost, LSTM) ile performans artırılabilir.
2. **Gelecekteki Çalışmalar:**
    - Zaman serisi analizi için ARIMA, Prophet gibi modellerin denenmesi.
    - Sezonsallık ve trendlerin daha iyi yakalanması için veri mühendisliği iyileştirmeleri.

Ekler

- **Kod Dosyası:**
  Tüm veri işleme, model eğitimi ve görselleştirme adımları içeren Python kodları.
- **Kullanılan Kütüphaneler:**
    - Pandas, Numpy
    - Scikit-learn
    - Matplotlib