Automotive Sales Prediction

21.11.2022 – Uğur Savcı

Table of Contents

Data Cleaning & Manipulation

- Standardizasyon olması için bütün sütunlar küçük harf ve boşluklara ‘_’ eklendi.
- Veri setindeki date sütununun formatını datetime oalrak değiştirildi
- Daha sonrasında index olarak atadık.

```
df.columns = df.columns.str.lower().str.replace(" ","_")
```

```
df = df.iloc[:149,:]
```

```
df['date'] = pd.to_datetime(df['date'], format='%Y/%m/%d')
df = df.set_index('date')
df = df.asfreq('MS')
df = df.sort_index()
```

Understanding Data

Date	otv_orani	faiz	eur/tl	kredi_stok	otomotiv_satis
2010-01-01	37.00	13.18	2.11	341,244.13	20,095.00
2010-02-01	37.00	13.27	2.07	351,940.95	31,172.00
2010-03-01	37.00	12.73	2.09	361,307.32	51,769.00
2010-04-01	37.00	12.01	2.01	373,575.34	54,946.00
2010-05-01	37.00	11.74	1.95	387,708.05	59,377.00

- Elimizdeki veri seti 2010 Ocak ayından başlayıp 2022 Mayıs ayına kadar devam etmektedir.

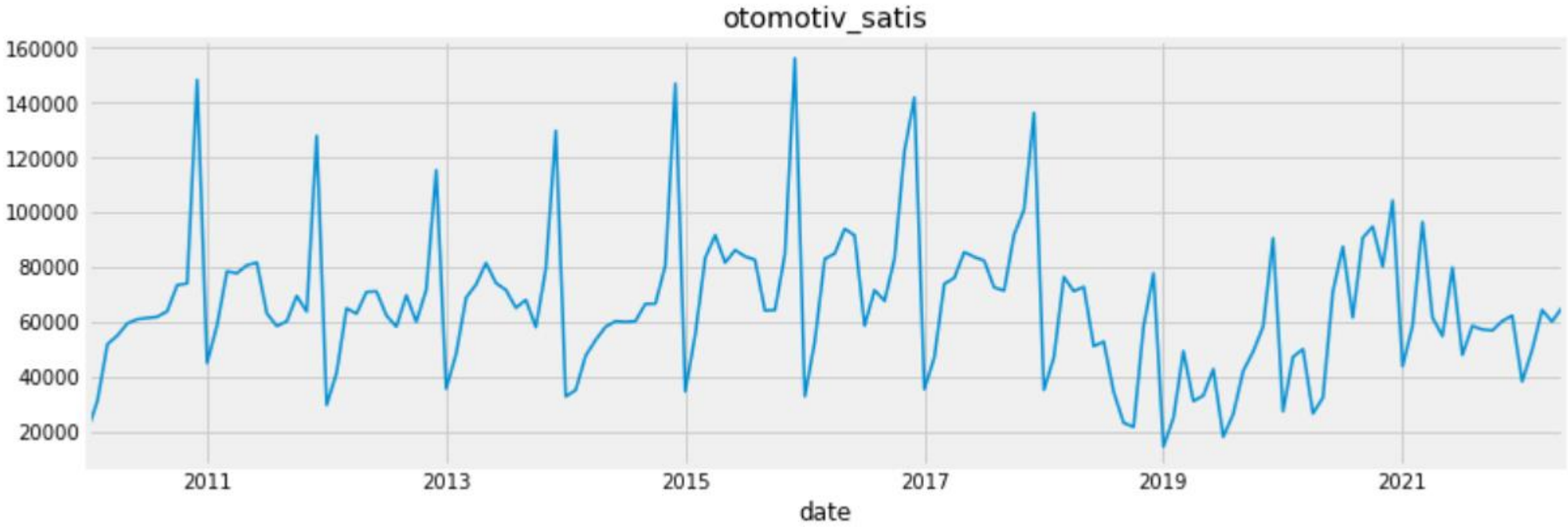
Understanding Data

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 149 entries, 2010-01-01 to 2022-05-01
Freq: MS
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   otv_orani       149 non-null    float64
 1   faiz            149 non-null    float64
 2   eur/tl          149 non-null    float64
 3   kredi_stok      149 non-null    float64
 4   otomotiv_satis  149 non-null    float64
dtypes: float64(5)
memory usage: 7.0 KB
```

- Herhangi bir eksik veri gözlemi veri setimizde Yok
- Veri tipleri hepsi için float.

Exploratory Data Analysis

- Otomotiv satışı 2018 ‘e kadar yatay olarak devam ederken 2018 yılından sonra düşüş gerçekleşti.
- Yılın son aylarında otomotiv satışlarında artış görülürken, Ocak ve şubat aylarında otomotiv satışları genellikle en düşük seviyedeler.



Feature Engineering

- Modelimizin daha iyi öğrenmesini sağlamak için verimize bazı değişkenler ekleyebiliriz.
- Bunlar ;
- **Tarih değişkenleri** : Yıl, Ay, Hafta, Çeyrek Yılın Haftası, Seri, Mevsim vb
- **Gecikmeli (Lag) Değişkenler** : Genellikle hedef değişkenin t-1 bir önceki değeri t satırına yazılarak yeni bir değişken oluşturulur. (Period olarak 1 den farklı değerler seçebiliriz.
- **Window değişkenler** : Bu değişkenler hedef değişkenin bir önceki aylardaki aggregate değerleridir. Son 3 aylık ortalama standart sapması vs.

Feature Engineering

Date değişkenleri olarak eklenen yeni veriler.

Adding Date Features

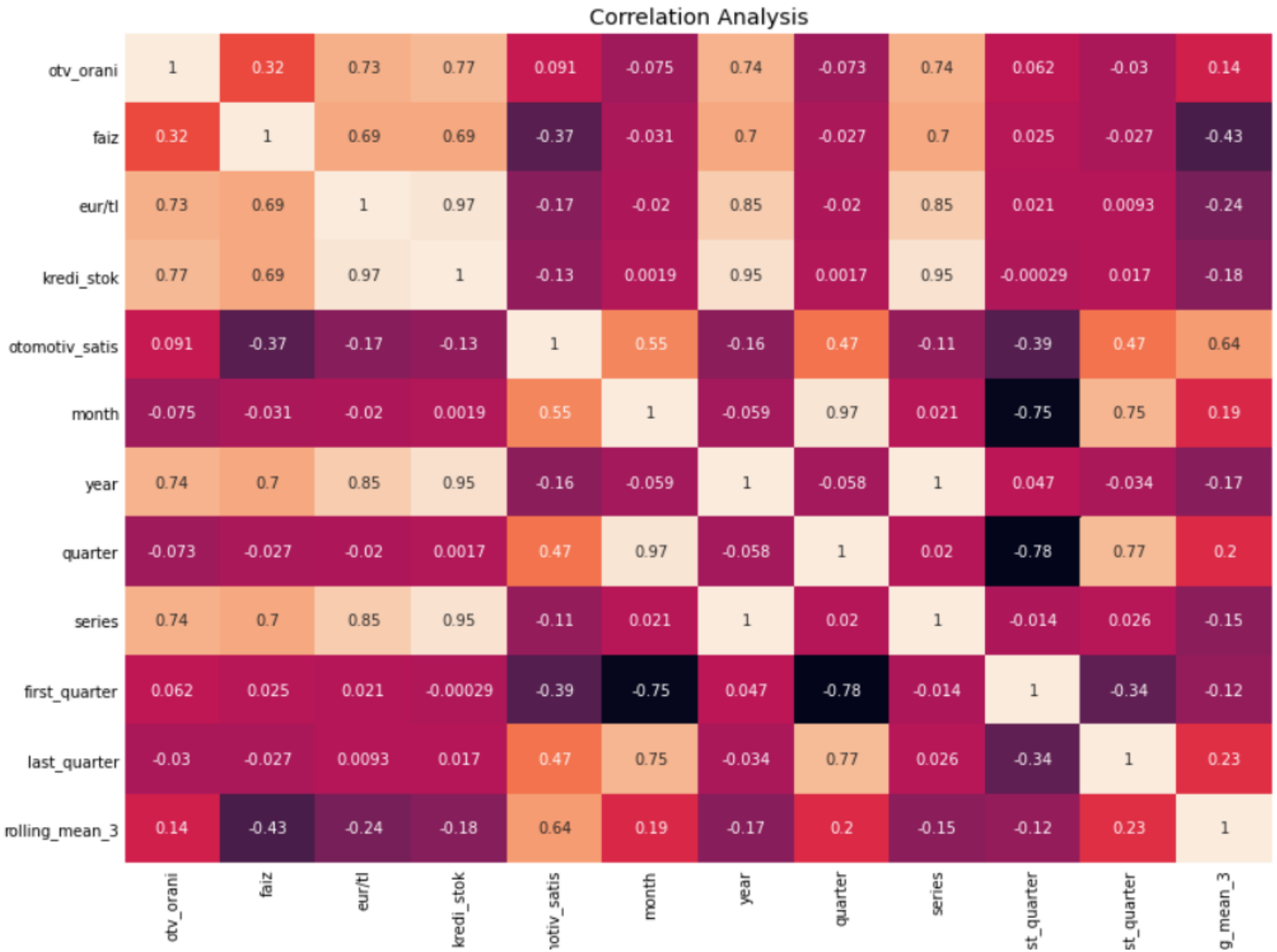
```
]: df["month"] = df.index.month
   df["year"] = df.index.year
   df["quarter"] = df.index.quarter
   df['series'] = np.arange(1,len(df)+1)
   df["first_quarter"] = df.quarter.apply(lambda x : 1 if x== 1 else 0)
   df["last_quarter"] = df.quarter.apply(lambda x: 1 if x== 4 else 0)
```

Son 3 ay ortalaması

Rolling Mean

```
df['rolling_mean_3'] = df['otomotiv_satis'].rolling(3).mean()
```

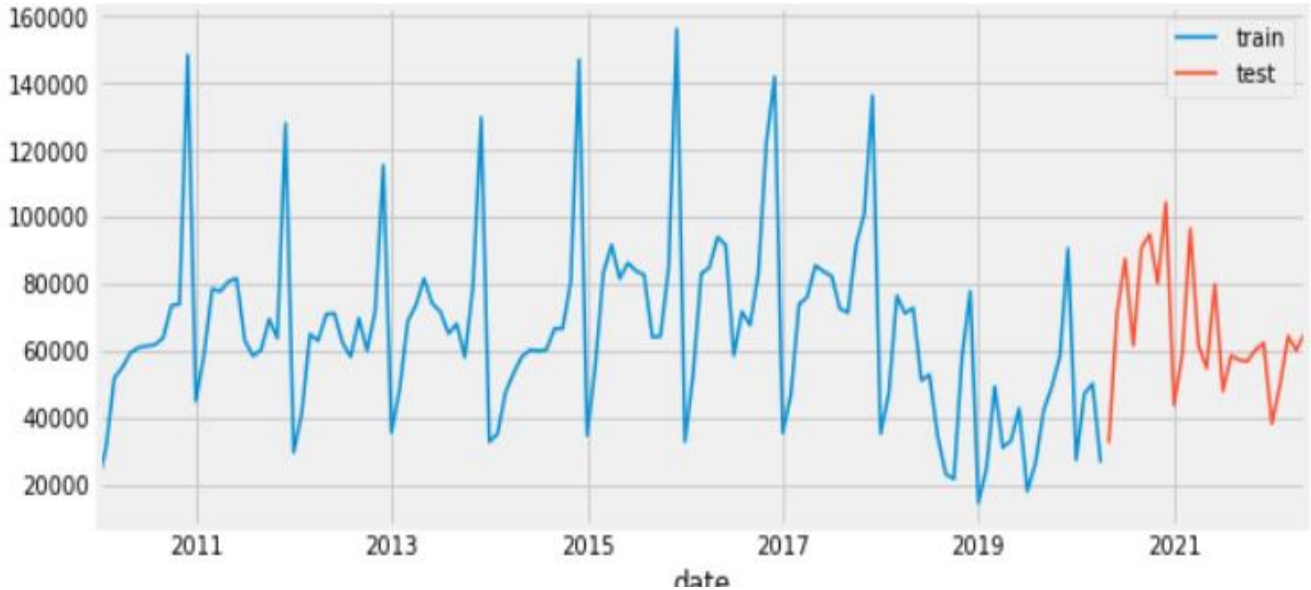
Exploratory Data Analysis



- Kredi stok ve eur/tl arasında güçlü korelasyon vardır.
- Otomotiv satışı ilk çeyrek ile negatif korelasyona sahip iken last_quarter ile güçlü pozitif korelasyona sahiptir.

Model Building & Prediction

Train-Test Set



- Son 25 Ayı test olarak kullanırken diğer veriler ile modelimizi test edeceğiz.
- Değerlendirme metriği olarak MAPE'yi seçtik.

Model Building & Prediction

Lag değişkenlerle birlikte **Baseline Model**

- Baseline model sadece lag değişkenleri ve hedef değişkeninden oluşuyor.
- Modelimizi kurmak için skforecast kütüphanesini kullandık.
- Model recursive olarak bir sonraki periodu hesaplayacaktır.
- Modeli kurarken parametre olarak lag= 15 seçtik. 15 farklı gecikmeli değişken oluşturduk.
- Multicollinearity problemini çözmek adına Regresyon olarak Ridge seçtik.
- Bu şekilde verdiğimiz parametreye göre . Modelin overfit olmasını engellemiş olacağız.

```
forecaster = ForecasterAutoreg(
                 regressor = Ridge(.1),
                 lags      = 15
                 |
             )

forecaster.fit(
        y = data_train['otomotiv_satis'],

            )
```

Model Building & Prediction

Prediction

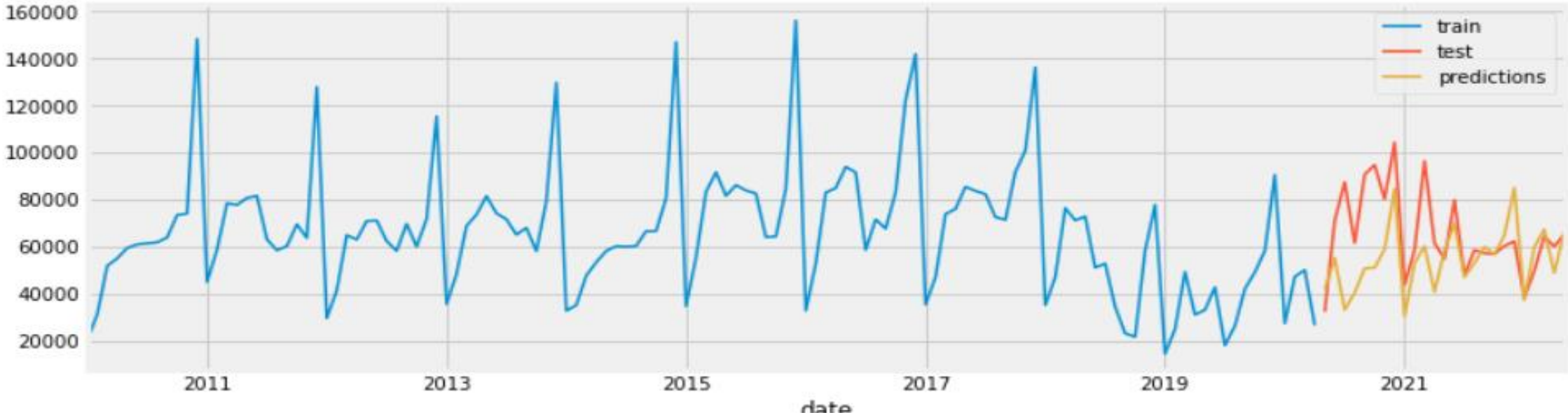
```
steps = 25
predictions = forecaster.predict(steps)
```

Test Error

```
error_mape = mean_absolute_percentage_error(
                y_true = data_test['otomotiv_satis'],
                y_pred = predictions
             )

print(f"Test error (mape):% {round((error_mape),3)*100}")
```

Output : **%20.9**



Model Building & Prediction

Prediction

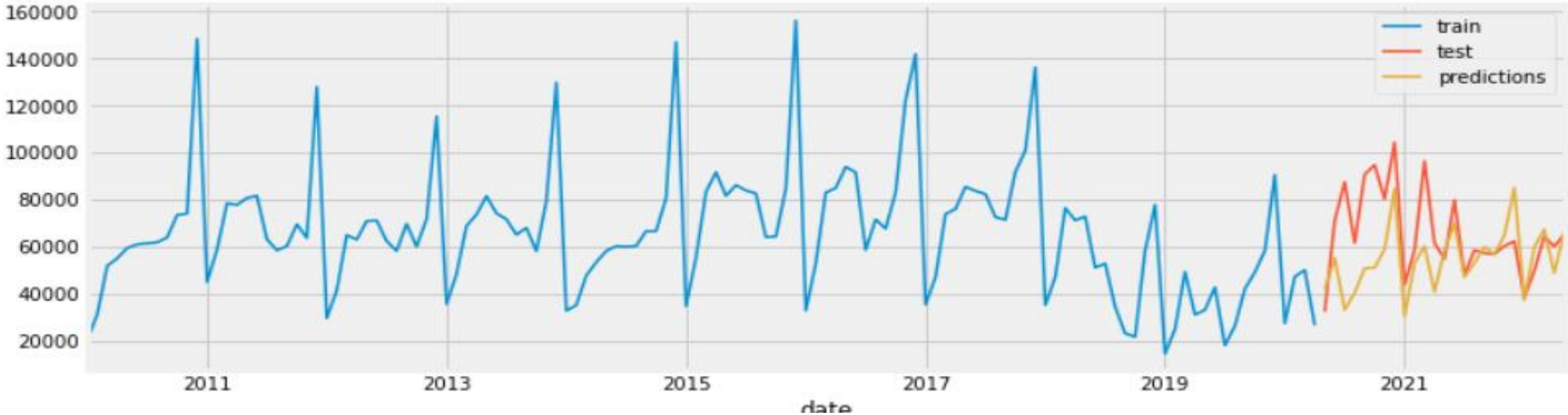
```
steps = 25
predictions = forecaster.predict(steps)
```

Test Error

```
error_mape = mean_absolute_percentage_error(
                y_true = data_test['otomotiv_satis'],
                y_pred = predictions
             )

print(f"Test error (mape):% {round((error_mape),3)*100}")
```

Output : **%20.9**



Model Building & Prediction

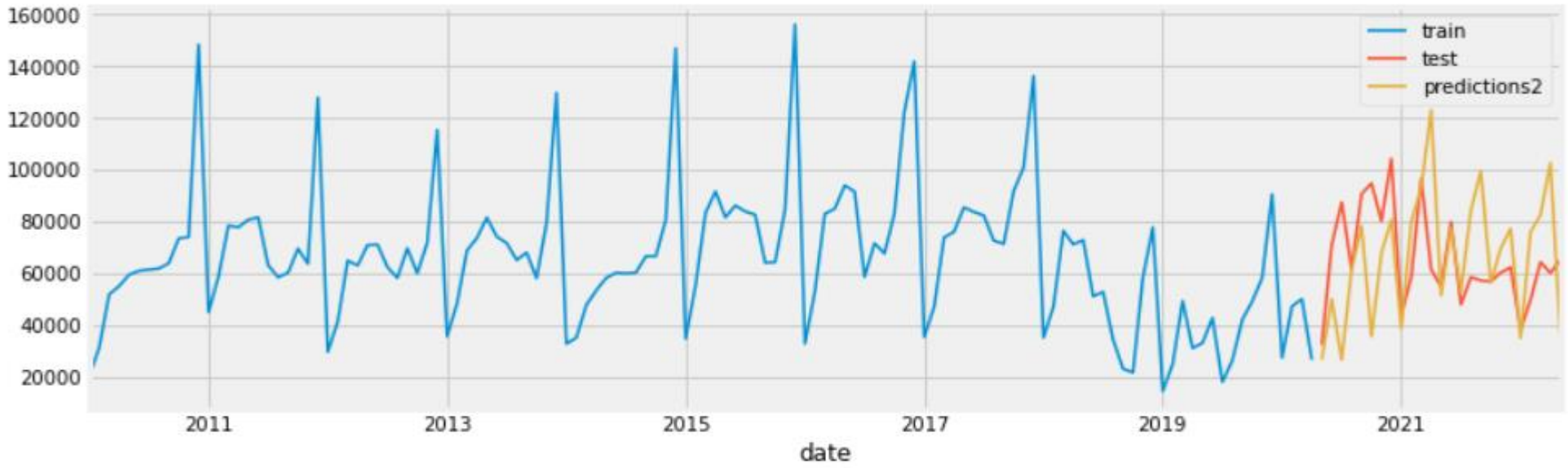
Model with Features

```
steps = 25
predictions2 = forecaster2.predict(steps=steps,exog = data_train[features])
```

```
error_mape = mean_absolute_percentage_error(
                y_true = data_test['otomotiv_satis'],
                y_pred = predictions2
            )

print(f"Test error (mape):% {round((error_mape),2)*100}")
```

Output : **%30.1**



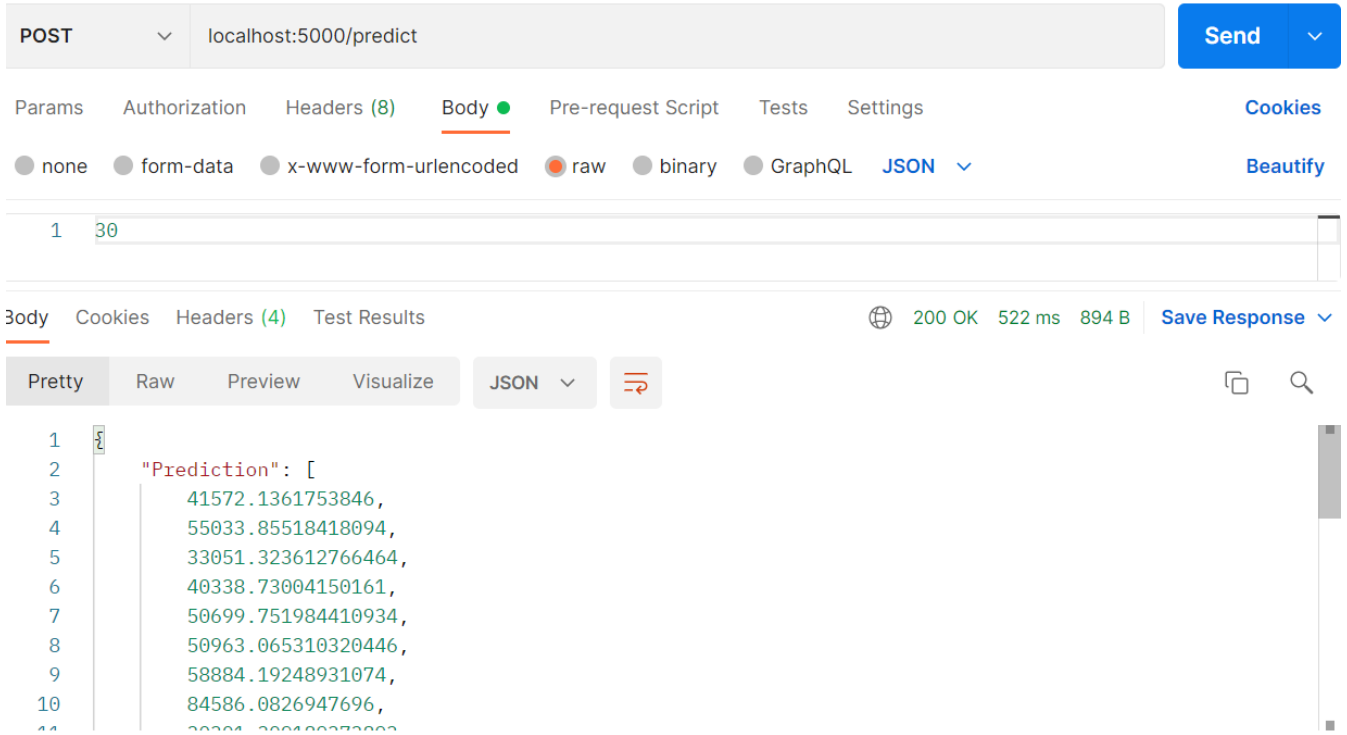
Prediction

- Sadece lag olan modelimizin performansı daha iyi çıktı.
- Modele göre Haziran – 2022 – 2023 dönemleri için tahminleri aldık.

	pred
2022-06-01	75334.973713
2022-07-01	52062.950842
2022-08-01	58188.335866
2022-09-01	63357.121213
2022-10-01	58267.150301
2022-11-01	67341.510781
2022-12-01	83157.501533
2023-01-01	40809.398093
2023-02-01	62600.433519
2023-03-01	69504.188992
2023-04-01	51513.808668
2023-05-01	68812.428978
2023-06-01	75696.384310

Postman API Testing

- Flask ile rest api kuruldu ve api test edildi.
- Kaç ay sonrasını tahmin etmek istiyor isek modele sadece sayı(period) olarak vermemiz yeterli oluyor.



Model Deployment

- Modeli deploy etmek için Dockerfile ve requirements.txt oluşturdum.

- Ancak aws ec2 instancetabir çok kez denememe rağmen private key hatası aldığım için maalesef deploy edemedim.

- Fakat deployment ile ilgili daha önce yaptığım çalışmanın linkini buraya bırakıyorum.
- https://churnpredictionapp1.herokuapp.com/

Anlayışınız için şimdiden teşekkür ederim.