

Otomativ Satış Tahmin Projesi

Gökalp Eren Akol

Problem Tanımı ve Veri Analizi

Problem:

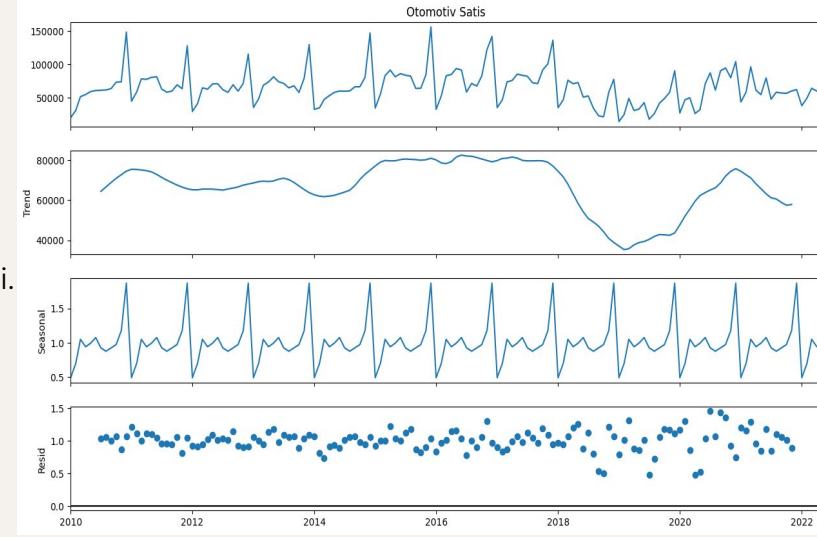
2022.06 – 2023.06 döneminde otomotiv satış miktarlarını, ekonomik değişkenlere duyarlı bir tahmin modeliyle öngörmek.

Veri Analizi:

Kriz dönemlerini temsil eden ek değişkenler oluşturuldu.

Ekonomik göstergeler ile satışlar arasındaki korelasyon ilişkileri incelendi.

Satış serisinin trend ve sezonsal yapısı analiz edildi.

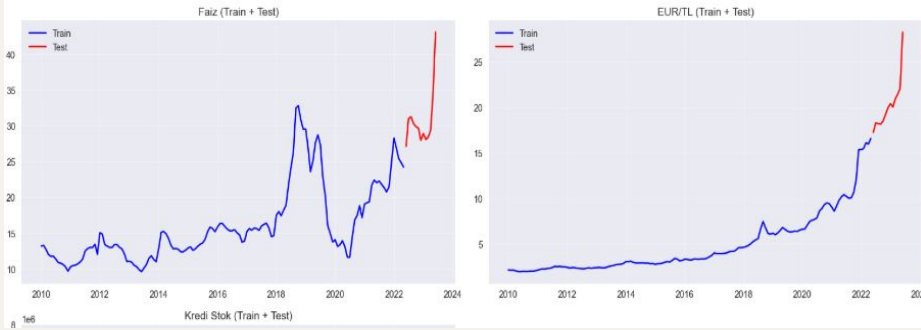


Eksik Veri analizi

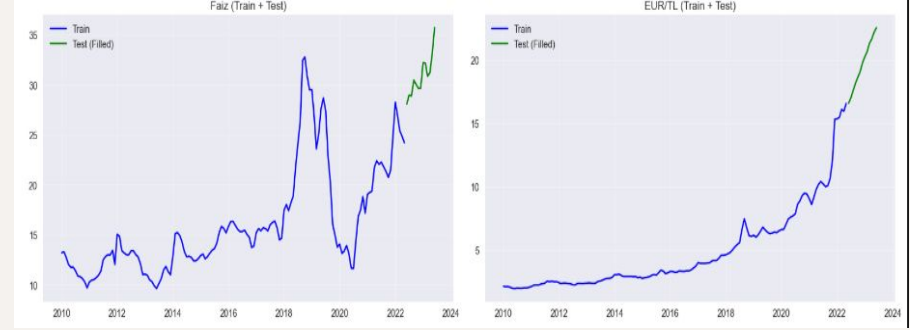
2022-06 tarihinden sonraki eksik bağımsız veriler (ÖTV, EUR/TL vs.) EVDS API ile grafiklerden aylık olarak toplandı. Bu veriler gerçek hayatta ileriye dönük elde edilemeyeceği için sadece analiz amaçlı tutuldu ve **kullanılmadı**.

Daha sonra proje gereksinimlerinden dolayı Prophet modeli ile bağımsız verilerin tahmini yapıldı. Final model oluşturulurken tahmin edilen değerler kullanıldı.

TCMB EVDS Verileri



Prophet Tahminleri



Modelleme Yaklaşımları

Projede matematiksel anlamda birbirinden farklı tahmin modelleri kullanıldı. Bir sonraki sayfada ana modelin detayları bulunmaktadır. Diğer modellerde tüm detaylara girilmeyecek olup, yapılan temel çalışmalar şu şekilde özetlenebilir.

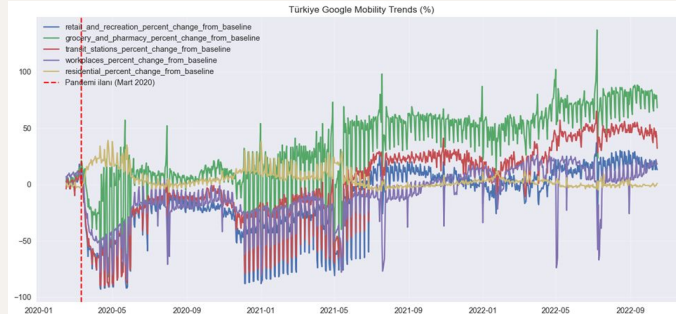
ARIMA/SARIMA: Exojen değişkenler ve PCA entegrasyonu ile zaman serisi tahminleri.

XGBoost: Farklı feature engineering varyasyonları (lag, rolling, vb.) ile çok sayıda deneme.

Prophet: Güçlü mevsimsellik ve trend değişimlerini yakalamada kullanıldı.

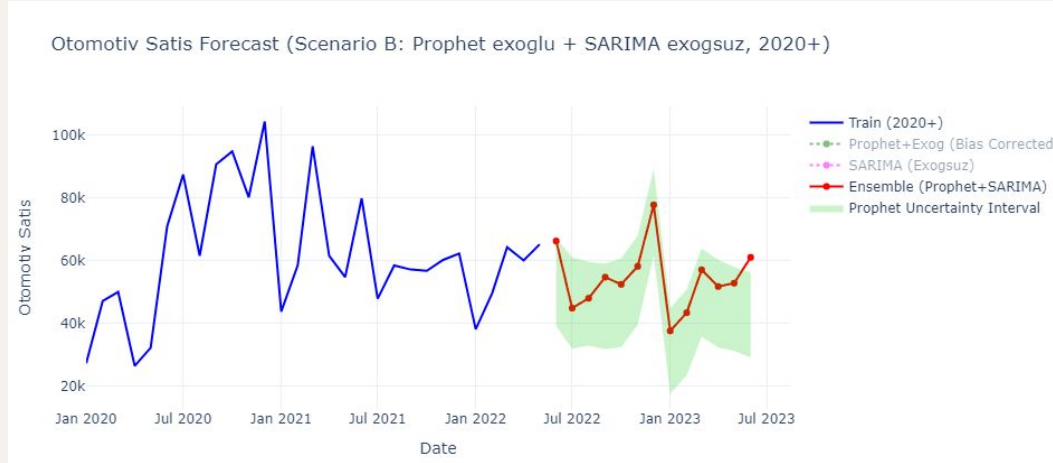
Fold ve Doğrulama Stratejileri: Expanding Window, Rolling Window ve Purged K-Fold CV uygulandı.

Proje boyunca en önemli zorluk **2018–2020 döneminde yaşanan ekonomik kriz** ve **pandeminin otomotiv satışlarında yarattığı ani dalgalanmalar** oldu. Bu dalgalanmaları modele yansıtabilmek için **Google** tarafından toplanan pandemi ve kriz verileri ek değişken olarak sisteme entegre edildi. Ancak bu yaklaşım, modelin geçmiş veriye aşırı uyum sağlamasına neden olarak **overfitting** problemine yol açtı.



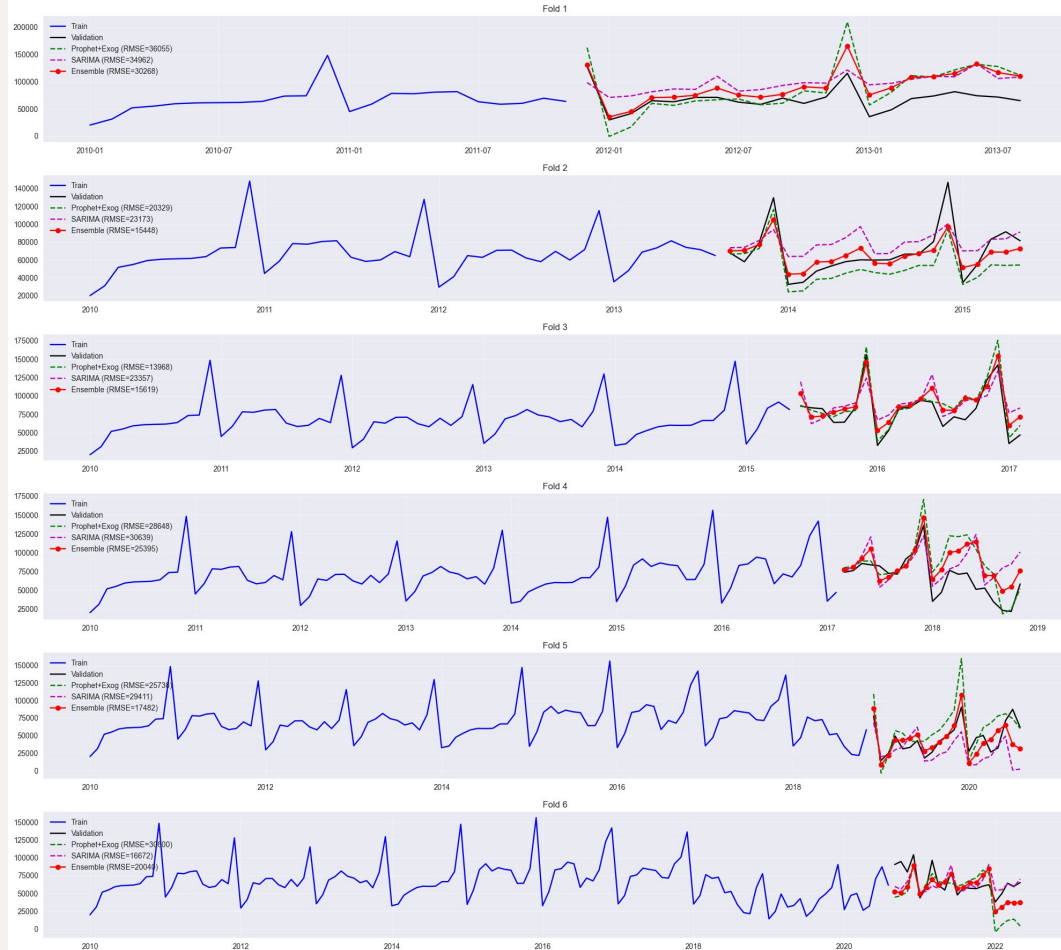
Final Model

Final modelimizde **hibrit** bir yaklaşım bulunmaktadır. Facebook'un piyasaya sürdüğü **Prophet** modelini faiz, EUR gibi bağımsız değişkenlerle ve kriz dönemlerini temsil eden değişkenlerle besledik. Her bir validasyon foldunda bias correction uyguladık. **SARIMA** modelini de hibrit bir şekilde bağımsız değişkenleri beslemeden kullandık. Buradaki ana fikir iki ayrı modelin ayrı özelliklerini birleştirerek hem trendi hem de makro etkilerin ortalamasını almaktı. Notebookta bunların ensemble alpha değeri (hangi modele ağırlık verileceği) 0.5 olarak belirlenmiş olup trend ya da makro etkilere bağlı olarak değiştirilip modelin davranışını değiştirebiliriz. Aşırı uyum gösteren modeller her bir foldda ortalama olarak 12.000 RMSE değeri bulundururken bu modelimiz ortalama olarak 20.700 RMSE değerindedir. Daha genel ve önceki verimize tam olarak uyum sağlamaması için parametre ayarlaması yapılmıştır. Expanding window split üzerinde sonuçları bir sonraki sayfada görebilirsiniz.



Her bir fold başına modellerin tahminleri.

Ensemble modelimiz, iki modelin hibrit olarak çalışmasıdır.



API Design Test Case - 1

GET `http://localhost:8000/get_inputs?date=2022-07-01`

Params Authorization Headers (6) Body Scripts Tests Settings

Query Params

Key	Value
date	2022-07-01

Body Cookies Headers (4) Test Results

{ } JSON Preview Visualize

```
1 {
2   "date": "2022-07-01",
3   "EUR_TL": 16.58113499667224,
4   "Faiz": 28.072871151121323,
5   "Kredi_Stok": 5060853.234431182,
6   "OTV_Orani": 65.0,
7   "alpha": 0.5
8 }
```

get_inputs ile kullanıcı, talep edilen tahmin tarihi için kullanılacak olan tahmin edilmiş bağımsız değişkenleri (EUR/TL, ÖTV vs.) alabilir.

POST `http://localhost:8000/forecast`

Params Authorization Headers (8) Body Scripts Tests Settings

none form-data x-www-form-urlencoded raw binary

```
1 {
2   "date": "2022-07-01",
3   "EUR_TL": 16.58113499667224,
4   "Faiz": 28.072871151121323,
5   "Kredi_Stok": 5060853.234431182,
6   "OTV_Orani": 65.0,
7   "alpha": 0.5
8 }
```

Body Cookies Headers (4) Test Results

{ } JSON Preview Visualize

```
1 {
2   "date": "2022-07-01",
3   "prophet_forecast": 46370.25563266318,
4   "sarima_forecast": 43316.95249260511,
5   "ensemble_forecast": 44843.60406263414,
6   "alpha": 0.5
7 }
```

forecast ile kullanıcı, tarih ve bağımsız değişkenleri JSON formatında girdikten sonra sistem üç farklı modelin tahminlerini üretir. Bu modellerin sonuçları ayrı ayrı gösterilir, ancak asıl tahmin olarak Ensemble modeli tarafından hesaplanan sonuç kullanılır.

POST `http://localhost:8000/predict`

Params Authorization Headers (8) Body Scripts Tests Settings

none form-data x-www-form-urlencoded raw binary

```
1 {
2   "date": "2022-07-01",
3   "EUR_TL": 16.58113499667224,
4   "Faiz": 28.072871151121323,
5   "Kredi_Stok": 5060853.234431182,
6   "OTV_Orani": 65.0,
7   "alpha": 0.5
8 }
```

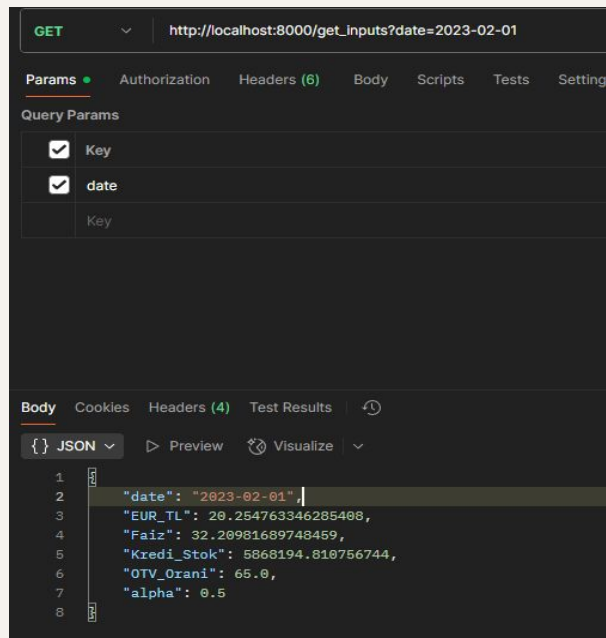
Body Cookies Headers (4) Test Results

{ } JSON Preview Visualize

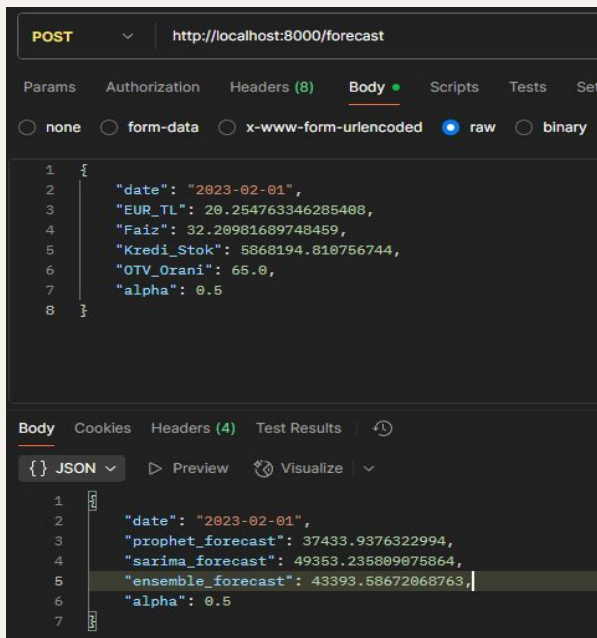
```
1 {
2   "date": "2022-07-01",
3   "prediction": 44843.60406263414
4 }
```

predict ile kullanıcı, sadece 1 tahmin çıktısı almak için ensemble modeli kullanabilir. Bağımsız değişkenler ve model karışım oranı parametre olarak değiştirilebilir.

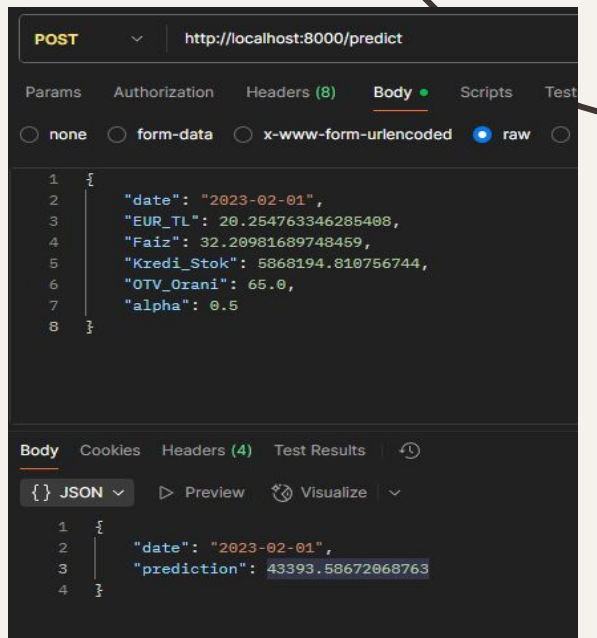
API Design Test Case - 2



get_inputs ile kullanıcı, talep edilen tahmin tarihi için kullanılacak olan tahmin edilmiş bağımsız değişkenleri (EUR/TL, ÖTV vs.) alabilir.



forecast ile kullanıcı, tarih ve bağımsız değişkenleri JSON formatında girdikten sonra sistem üç farklı modelin tahminlerini üretir. Bu modellerin sonuçları ayrı ayrı gösterilir, ancak asıl tahmin olarak Ensemble modeli tarafından hesaplanan sonuç kullanılır.



predict ile kullanıcı, sadece 1 tahmin çıktısı almak için ensemble modeli kullanabilir. Bağımsız değişkenler ve model karışım oranı parametre olarak değiştirilebilir.

Teşekkürler

Kısıtlı süreden dolayı her detayıyla projeyi açıklayamamış olsam da merak ettikleriniz için bana ulaşmanızdan mutluluk duyarım. **README.md içerisinde api örnek kullanım senaryosu mevcuttur.**

Proje yapısı:

main_notebook.ipynb - Veri ön işleme, model geliştirme, çapraz doğrulama, metriklerin hesaplanması ve tahmin görselleştirmeleri.

Forecast_api.py - FastAPI tabanlı servis

app.py - streamlit tabanlı tahmin görüntüleme arayüzü

models - modellerin bulunduğu klasör

presentation - sunum dosyası

Dockerfile - Docker dosyası