

FET445 Veri Madenciliği

Araba Fiyat Tahmini

Grup : Motor Çetesi

Tarih:25.12.2025

Araba Fiyat Tahmini

Bu projede, ikinci el araç piyasasında karşılaşılan fiyatların gerçek değerleri yansıtıp yansıtmadığını analiz etmek ve kullanıcıya veri temelli bir fiyat tahmini sunmak amaçlanmaktadır. Craigslist.org sitesinden elde edilen ikinci el araç verileri kullanılarak, araçların sayısal ve kategorik özelliklerine dayalı bir fiyat tahmin modeli geliştirilmiştir. Problem, araç özelliklerinden yola çıkarak satış fiyatını öngörmeyi hedefleyen bir regresyon problemi olarak ele alınmıştır. Model performansı MAE, RMSE ve R^2 metrikleri ile değerlendirilmiş; tahmin hatalarının kontrol altında tutulması ve yüksek açıklayıcılık sağlanması hedeflenmiştir.

Veri Seti Açıklaması

Bu projede, ABD'de yayınlanan ikinci el araç ilanlarını içeren Craigslist Cars & Trucks Dataset kullanılmıştır. Veri seti, Craigslist.org sitesi üzerinden web scraping yöntemiyle toplanmış araç ilanlarından oluşmaktadır. Amaç, araçların sayısal ve kategorik özelliklerini kullanarak satış fiyatlarını tahmin etmektir. Veri seti, farklı araç tipleri, üreticiler ve kullanım durumlarını kapsayarak geniş ve gerçekçi bir piyasa temsili sunmaktadır

Veri Seti Linki

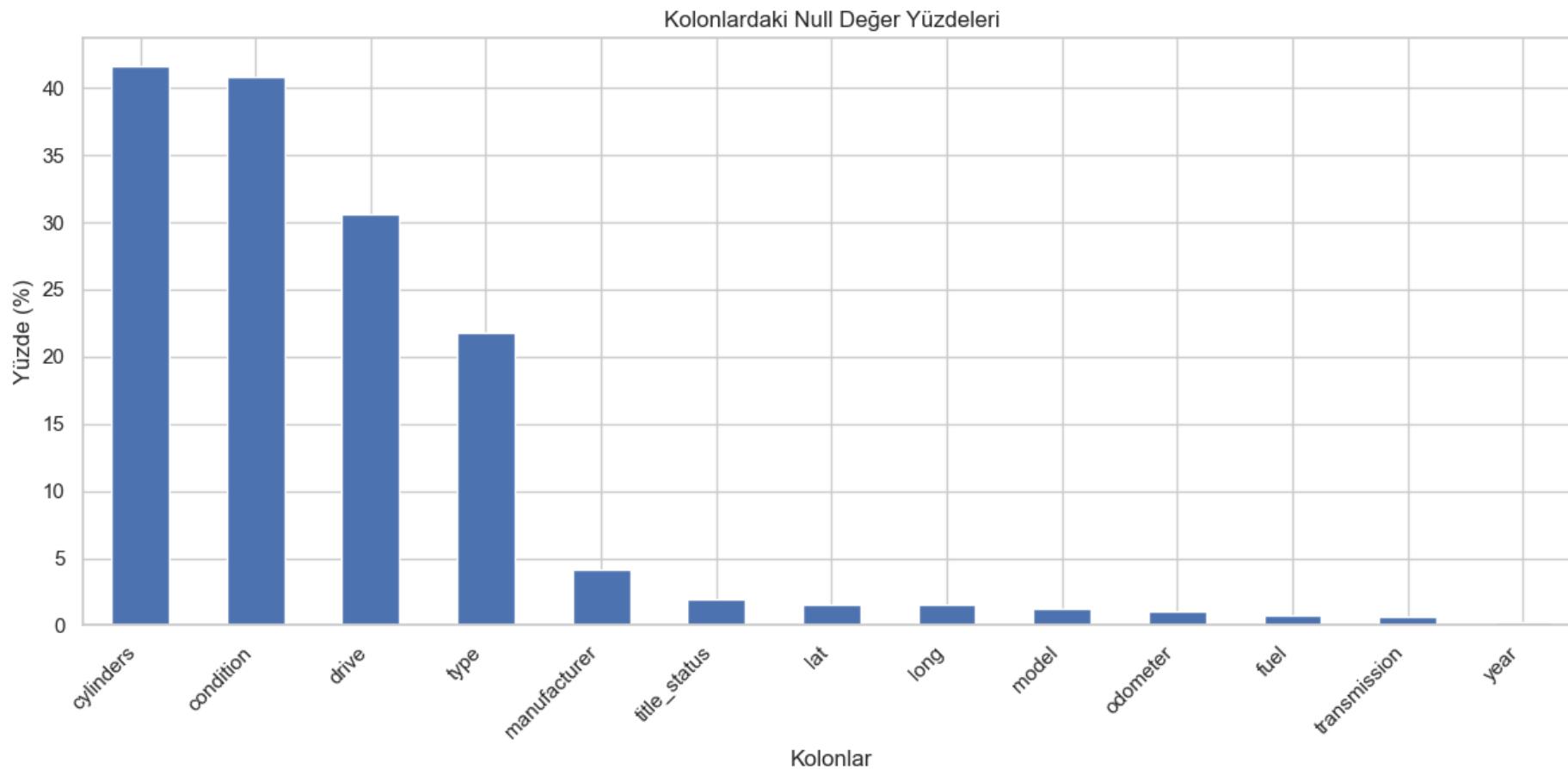
Veri setine Kaggle üzerinden erişilmiştir: <https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data>

Veri Seti Boyutu

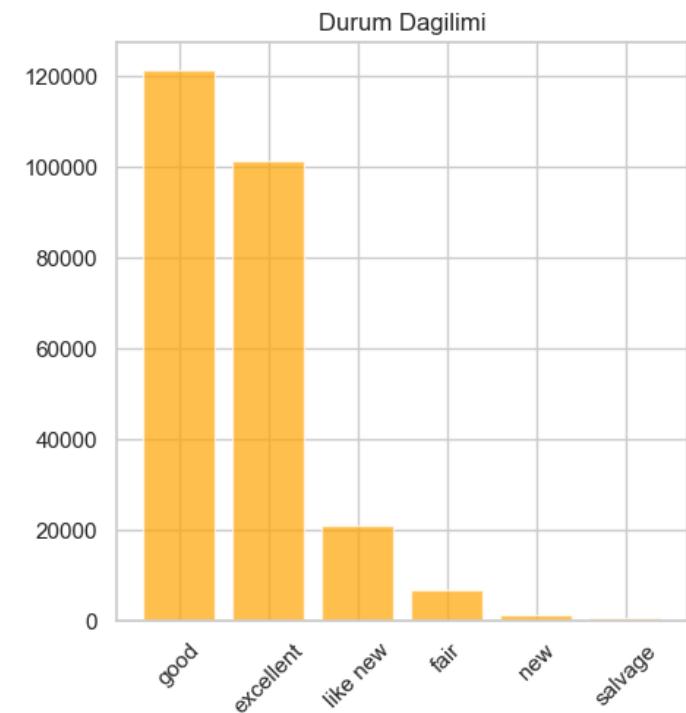
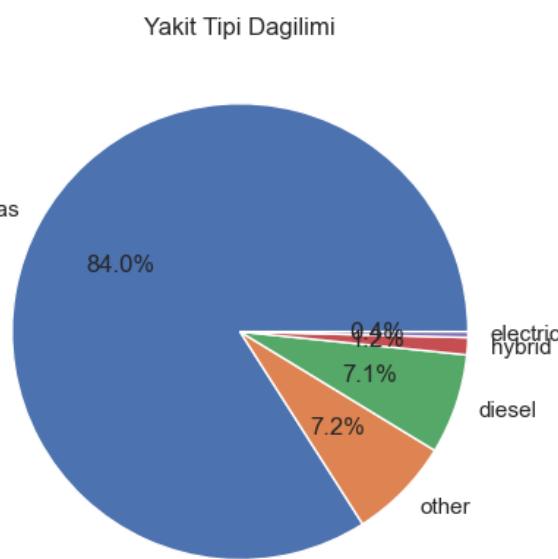
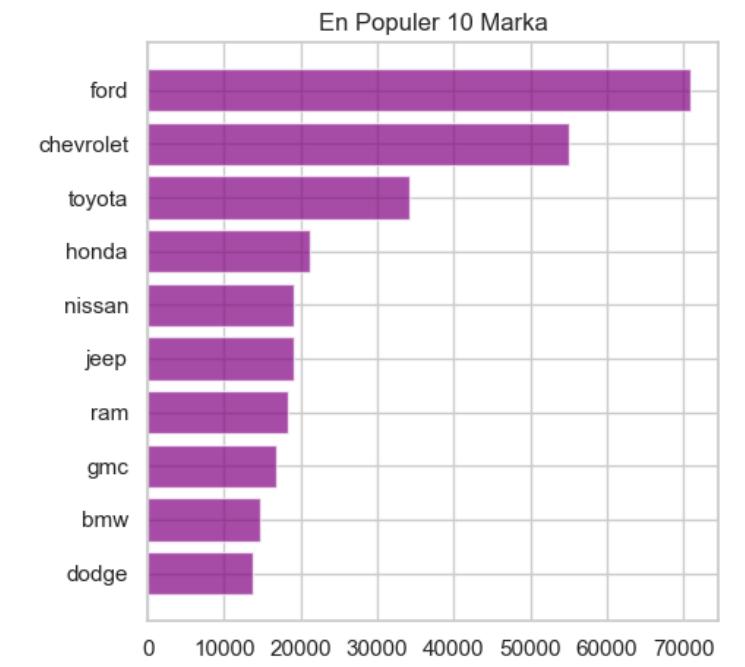
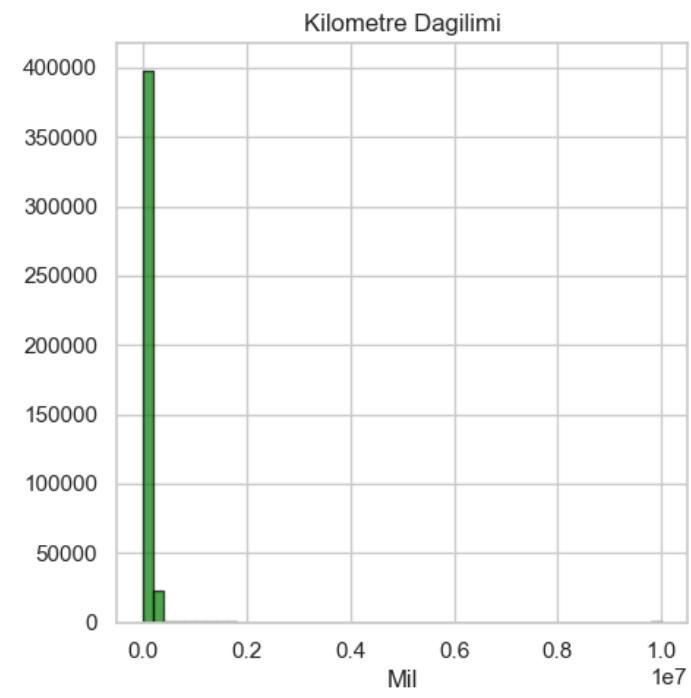
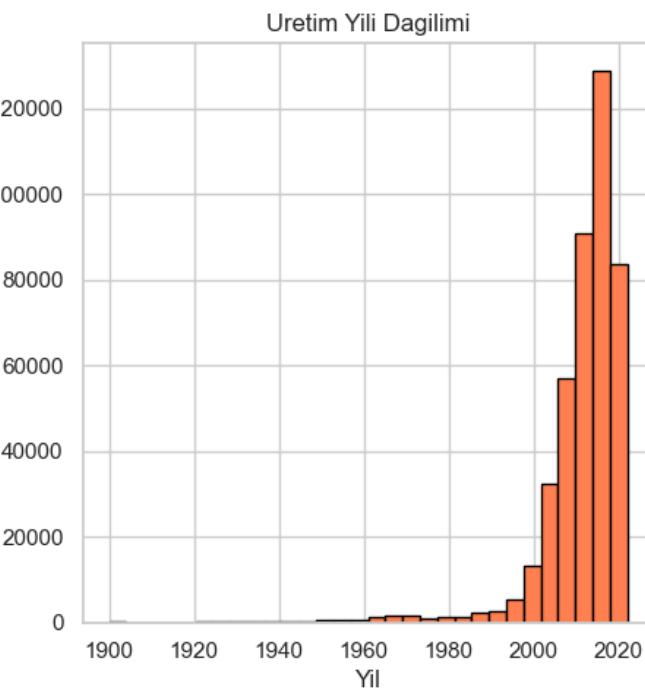
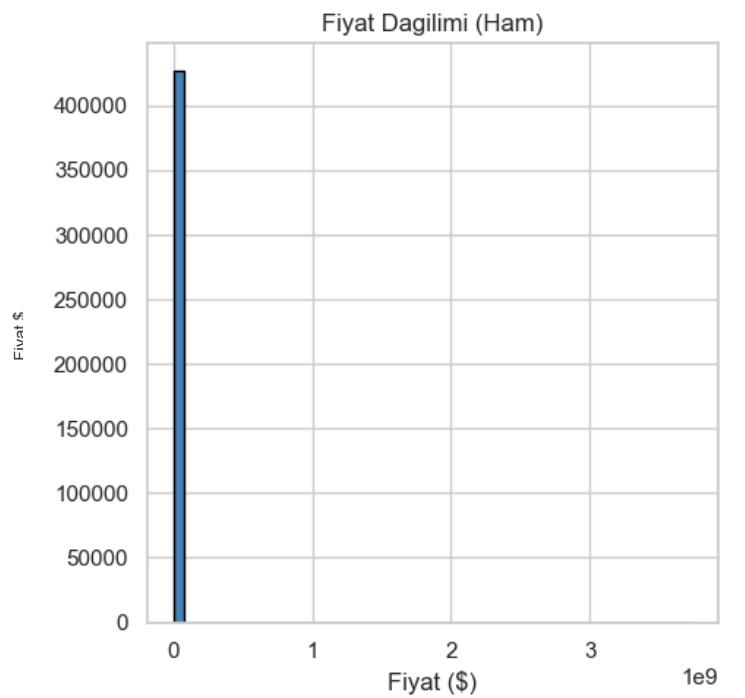
- Toplam satır sayısı: 426.880
- Toplam sütun sayısı: 26
- Dosya boyutu: ~1.45 GB (CSV)

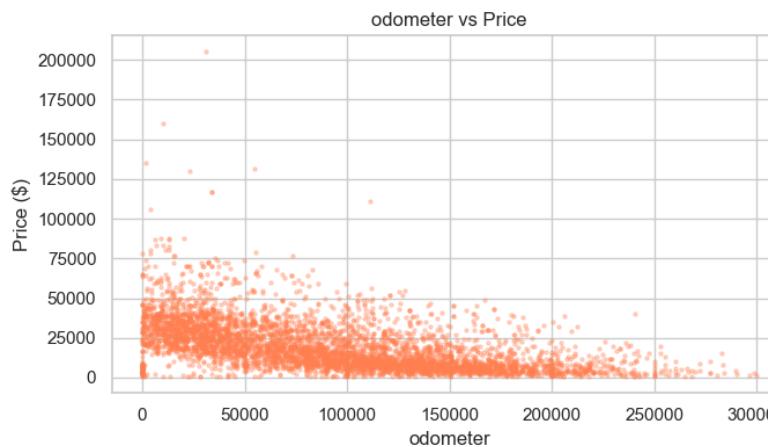
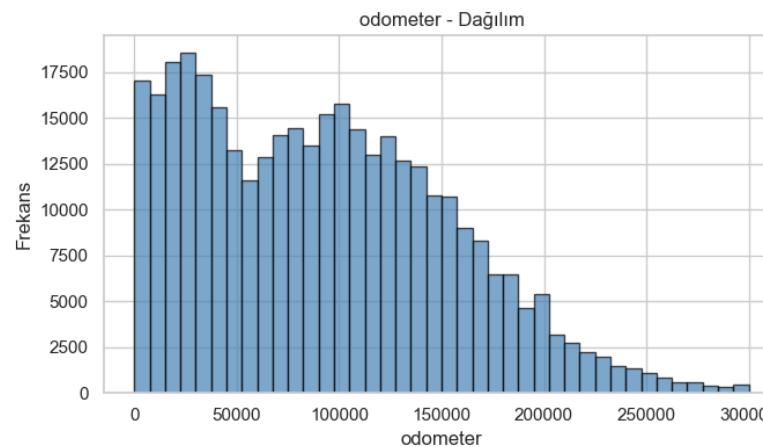
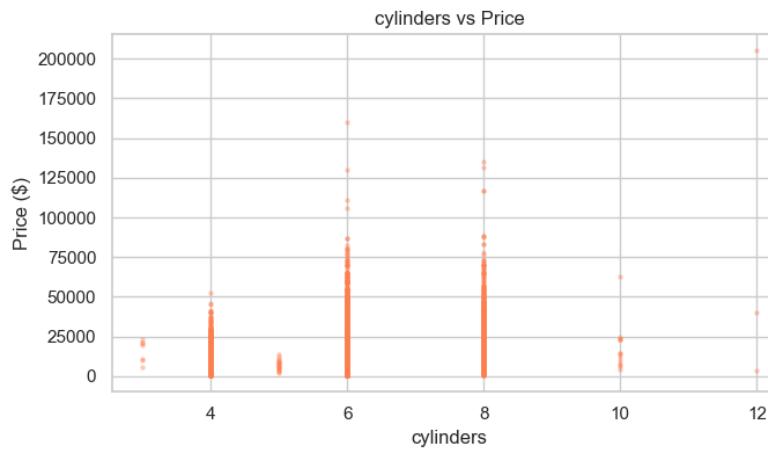
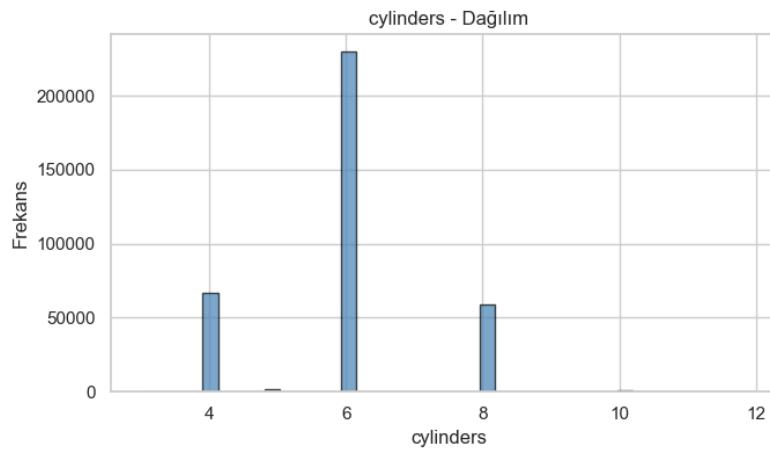
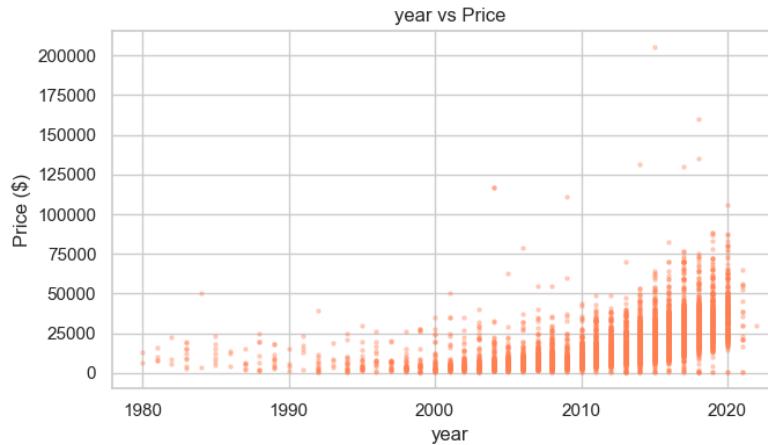
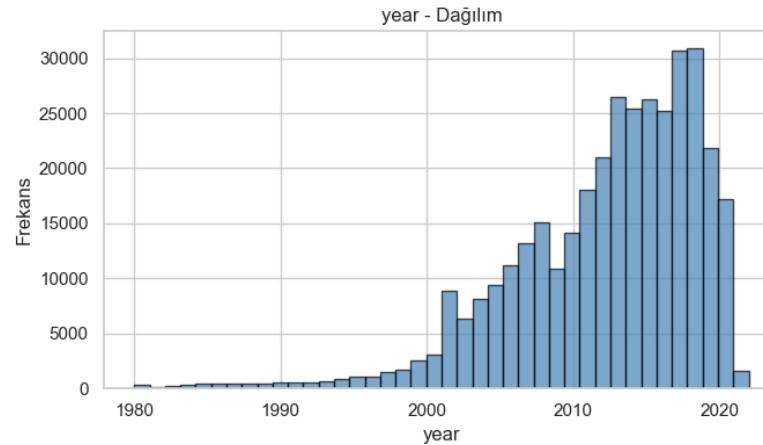
Veri Seti Özellik Tipleri

- Sayısal özellikler:
price (hedef değişken), year, odometer, cylinders, latitude, longitude
- Kategorik özellikler:
manufacturer, fuel, transmission, drive, type, paint_color, title_status, state, region
- Ordinal özellikler:
condition (durum seviyesi)

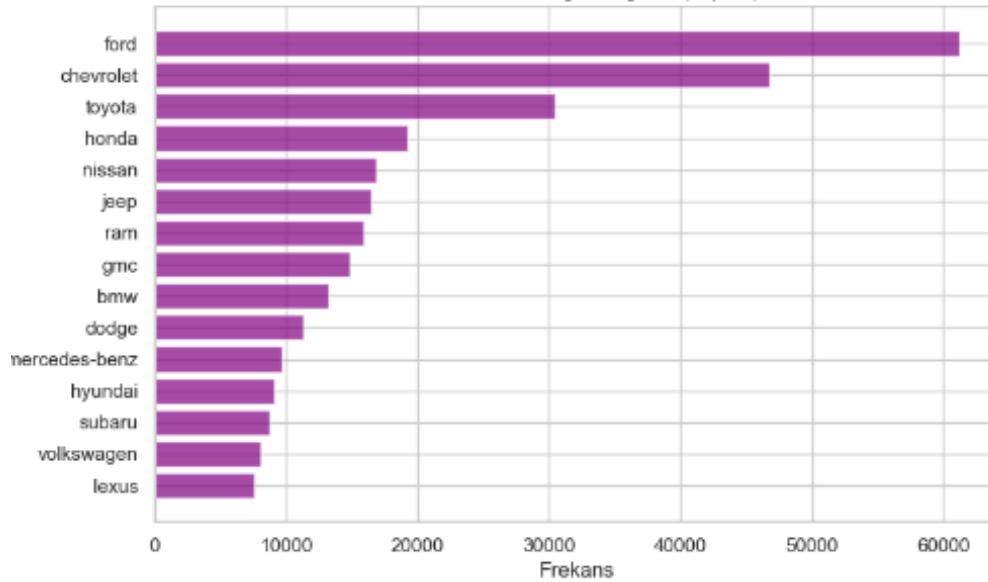


EDA öncesi kolonlardaki null değer yüzde grafiği

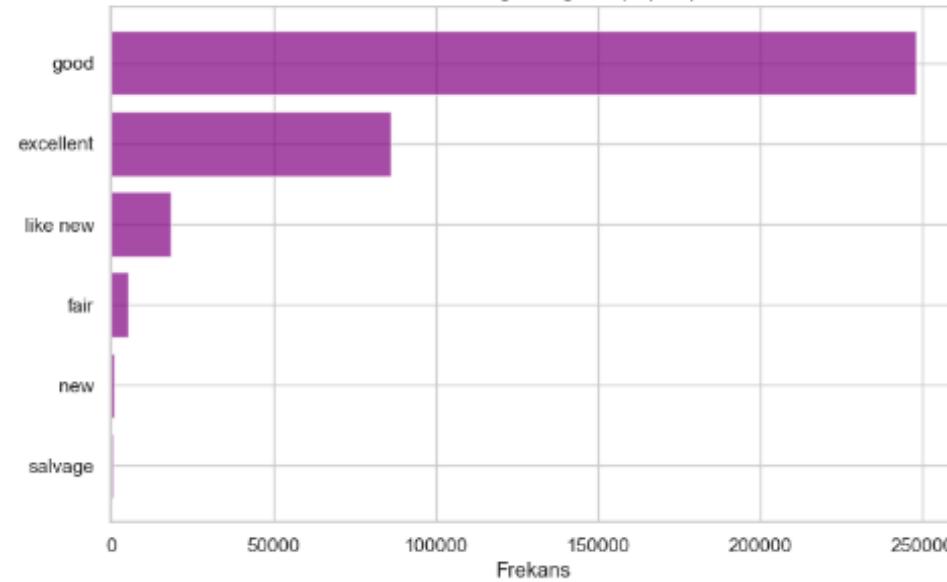




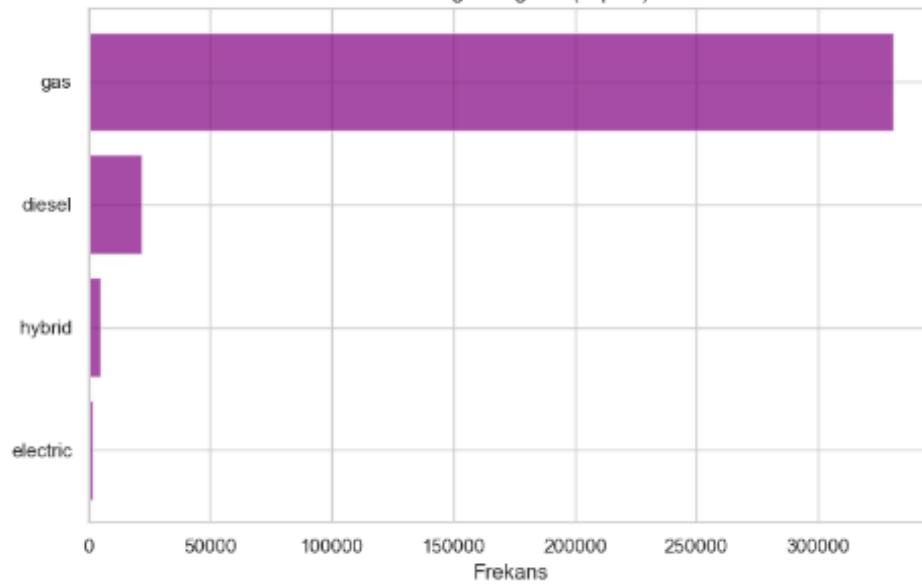
manufacturer - Değer Dağılımı (Top 15)



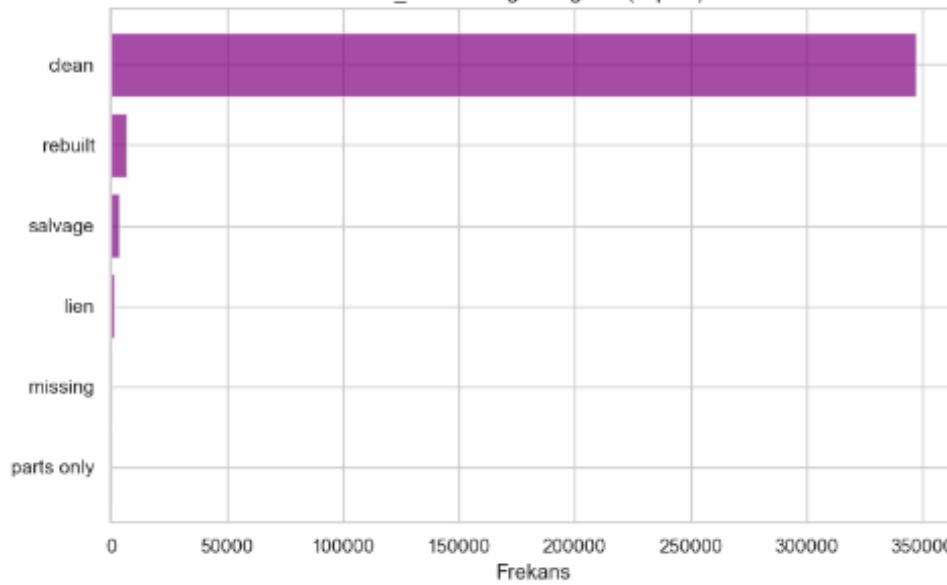
condition - Değer Dağılımı (Top 15)



fuel - Değer Dağılımı (Top 15)



title_status - Değer Dağılımı (Top 15)



Her bir ekip üyesi farklı feature engineering modelleri + sabit bir baseline model kullanarak farklı feature engineering tekniklerinin direkt olarak R2 , RMSE , MAE metriklerine etkisini test etmiştir. Kullanılan bazı feature engineering teknikleri:

```
# 1) Arac yası
df['car_age'] = 2025 - df['year']
print('car_age eklendi')
```

```
# 2) Yıllık km
df['mil_per_year'] = df['odometer'] / df['car_age'].replace(0, 1)
print('mil_per_year eklendi')
```

```
# 3) Condition düzeltme (km bazlı)
def correct_condition(row):
    odometer = row['odometer']
    if odometer <= 10000: return 'new'
    elif odometer <= 50000: return 'like new'
    elif odometer <= 100000: return 'excellent'
    elif odometer <= 150000: return 'good'
    else: return 'fair'
```

```
df['condition'] = df.apply(correct_condition, axis=1)
print('condition km bazlı düzeltildi')
```

```
def categorize_age(age):
    if age <= 2:
        return 'brand_new'
    elif age <= 5:
        return 'new'
    elif age <= 10:
        return 'mid_age'
    elif age <= 15:
        return 'old'
    else:
        return 'very_old'
```

```
df['age_category'] = df['car_age'].apply(categorize_age)
```

```
def categorize_engine(cylinders):
    if cylinders <= 4:
        return 'small_engine'
    elif cylinders <= 6:
        return 'medium_engine'
    else:
        return 'large_engine'
```

```
df['engine_category'] = df['cylinders'].apply(categorize_engine)
```

==== TYPE DOLDURMA (MODEL BİLGİSİ İLE) ====

Başlangıç type null: 86295

1) Model isminden type çıkarılıyor...

Sonrası null: 49988

2) Manufacturer bazlı mode...

Sonrası null: 0

3) Kalan null'lar 'unknown' yapılıyor...

Final null: 0

Type doldurma tamamlandı!

Type dağılımı:

```
type
sedan      117634
SUV        90340
pickup     74027
truck       33492
other       21246
coupe      20649
hatchback   17236
van         10720
wagon      10532
convertible  7152
mini-van    4682
offroad     570
bus         314
Name: count, dtype: int64
```

Toplam canonical model sayısı: 665

Toplam model varyant sayısı: 1050

=====
MODEL NORMALİZASYON İŞLEMİ
=====

Ham model NaN sayısı: 4,452

Normalize edilmiş model sayısı: 364,044

Sözlükte olmayan model sayısı: 202,516

Bu modeller 'unknown' olarak işaretlenecek (silinmeyecek)

Final dataframe boyutu: 368,498 satır (veri kaybı yok!)

Model kolonu hazır: 364,044 normalize edilmiş model

- Bilinen modeller: 165,982

- Unknown modeller: 202,516

=====
MODEL NORMALİZASYON TAMAMLANDI
=====

=====
UNKNOWN MODEL DEĞERLERİNİ DOLDURMA (Manufacturer'a göre)
=====

Doldurulacak unknown model sayısı: 202,516

Manufacturer bazında en popüler modeller bulundu: 40 manufacturer

Doldurma sonuçları:

- Doldurulan model sayısı: 202,100

- Kalan unknown model sayısı: 416

- Doldurma oranı: 99.8%

- Kalan 416 unknown değer 'other' olarak işaretlendi

=====
UNKNOWN MODEL DOLDURMA TAMAMLANDI
=====

Train Test Split Oranı : %80 Train %20 Test

Kullanılan Performans metrikleri : R2 , MAE , RMSE

Proje boyunca yapılan deneyler sonucunda, hem eğitim sürecinde GPU desteği sayesinde hızlı bir şekilde eğitilebilmesi hem de test verisi üzerinde en iyi performans metriklerini sağlaması nedeniyle **XGBoost** modeli en başarılı model olarak belirlenmiştir.

Model geliştirme sürecinde , hiperparametre optimizasyonu için en güçlü rakibi GridSearchCV den çok daha hızlı olmasına rağmen yakın verimlilikte sonuç vermesi sebebiyle çoğunlukla RandomizedSearchCV kullanıldı. RandomizedSearchCV ile optimizasyon sürecinde ağaç sayısı, maksimum derinlik, öğrenme oranı ve örneklemeye oranı gibi temel hiperparametreler ayarlanmıştır.

Modelin eğitilmesinde kullanılan feature seti; aracın üretim yılı, kilometre bilgisi, motor özellikleri, yakıt türü, şanzıman tipi, araç tipi, çekiş türü, üretici bilgisi ve coğrafi konum gibi hem sayısal hem de kategorik özelliklerden oluşmaktadır. Kategorik değişkenler uygun encoding yöntemleri ile dönüştürülmüş, sayısal değişkenler ise ölçeklendirilerek modele dahil edilmiştir.

Ekip Üyesi	Kullanılan Model	Hyperparameter Tuning Yöntemi	En Optimum Parametreler	Kullanılan Performans Metrikleri
Buğra	Random Forest Regressor	RandomizedSearchCV (n_iter=12, cv=3)	n_estimators=400, max_depth=20, min_samples_split=5, min_samples_leaf=2, max_features="sqrt"	RMSE, MAE, R ²
Buğra	PyTorch Neural Network (MLP)	Manual Tuning	learning_rate=0.001, batch_size=32, epochs=50	RMSE, Loss, R ²
Emirhan	XGBoost Regressor	RandomizedSearchCV (n_iter=30, cv=3)	n_estimators=1000, max_depth=8, learning_rate=0.05, subsample=0.8	RMSE, MAE, R ²
Emirhan	LightGBM Regressor	RandomizedSearchCV (n_iter=30, cv=3)	n_estimators=500, max_depth=10, learning_rate=0.1, num_leaves=31	RMSE, MAE, R ²
Utku	Bagging Ensemble (KNN)	RandomizedSearchCV (n_iter=10, cv=3)	n_estimators=200, max_samples=1.0, learning_rate=0.1	RMSE, MAE, R ²
Utku	CatBoost	Randomized Search	n_estimators=800, max_depth=8, learning_rate=0.05, loss_function="RMSE"	RMSE, MAE, R ²
Alper	HistGradientBoostingRegressor	Manual Tuning (iterative parameter refinement)	max_depth=15, learning_rate=0.03, max_iter=800	RMSE, MAE, R ²
Alper	Stacking Regressor	Grid Search (meta-model & base model selection)	base_models=[RF, LGBM, XGB], final_estimator=LinearRegression	RMSE, MAE, R ²

R² Score (↑)	RMSE (\$) (↓)	MAE (\$) (↓)
XGBoost : 0.8869	Random Forest : 3,950.88	Bagging KNN : 2,033.50
Random Forest : 0.8831	XGBoost : 3,992.31	Random Forest : 2,220.24
LightGBM : 0.8698	LightGBM : 4,389.76	XGBoost : 2,393.11
Bagging KNN : 0.8380	Bagging KNN : 5,153.38	Stacking Regressor : 2,426.28
Stacking Regressor : 0.8202	Stacking Regressor : 5,375.21	LightGBM : 2,745.36
CatBoost : 0.8178	CatBoost : 5,473.67	CatBoost : 2,660.70
HistGradientBoosting : 0.7299	HistGradientBoosting : 7,602.93	HistGradientBoosting : 4,152.12
PyTorch ResNet MLP : 0.7152	PyTorch ResNet MLP : 7,956.78	PyTorch ResNet MLP : 4,423.55

Bu projede, ikinci el araç fiyatlarını tahmin etmeye yönelik bir regresyon problemi ele alınmış ve farklı makine öğrenmesi modelleri test verisi üzerinde R^2 , RMSE ve MAE metrikleri kullanılarak karşılaştırılmıştır. Elde edilen sonuçlara göre, boosting tabanlı XGBoost modeli, test verisi üzerinde en yüksek R^2 değerine ulaşarak fiyat değişkenindeki varyansın büyük bir kısmını açıklamayı başarmış ve genel olarak en verimli modeli oluşturmuştur.

Buna karşılık Random Forest modeli, RMSE ve MAE metriklerinde en düşük hata değerlerini vererek tahmin hatalarının minimize edilmesi açısından öne çıkmıştır. Bu durum, farklı modellerin farklı performans metrikleri açısından avantaj sağlayabildiğini göstermektedir. Buna karşılık Random Forest modeli, RMSE ve MAE metriklerinde en düşük hata değerlerini vererek tahmin hatalarının minimize edilmesi açısından öne çıkmıştır. Bu durum, farklı modellerin farklı performans metrikleri açısından avantaj sağlayabildiğini göstermektedir.

Hiperparametre optimizasyonu aşamasında, hesaplama maliyeti ve verimlilik dengesi gözetilerek RandomizedSearchCV yöntemi tercih edilmiştir. Ayrıca hedef değişkene uygulanan log dönüşümü, modelin daha stabil öğrenmesine katkı sağlamıştır. Genel değerlendirme sonucunda, proje kapsamında belirlenen performans kriterlerinin büyük ölçüde karşılandığı ve XGBoost modelinin en başarılı model olduğu sonucuna varılmıştır.