

Makine Öğrenmesi Dönem Projesi

Amerika'da Araç Fiyatını Etkileyen Faktörler

Görkem GÜDER - 210911086

Oğuzhan Enes OZAT - 210911089

Mehmet Burak DURDU - 210911059

Batuhan KÜÇÜKKISA - 210911056

Veri setinin kolonları:

İd : Her bir kayıt için benzersiz kimliktir.

Vin :Araç tanımlama numarasıdır.

Price :Araç fiyatıdır.

Miles: Aracın kullanıldığı mil sayıdır.

Stock no :Araç stoğu için belirlenmiş numaradır.

Year : Araç model yılıdır.

Make:Araç üretici/marka ismidir.

Model : Aracın model ismidir.

Trim: Aracın donanım seviyesidir.

Body_type : Aracın gövde tipidir.

vehicle_type: Aracın tipidir.

Drivetrain: Aracın aktarma bileşenlerinin ismidir.

Transmission: Aracın şanzıman türüdür.

fuel_type: Aracın yakıt türüdür.

Engine_size : Aracın motor hacmi değeridir.

Engine_block : Aracın motor tipidir.

Seller_name : Satıcı firmanın ismidir.

Street :Satıcı firmanın adres bilgisidir.

City : Satıcı firmanın bulunduğu şehir bilgisidir.

State : Satıcı firmanın bulunduğu eyalet bilgisidir.

Zip : Satıcı firmanın posta kodu bilgisidir.

Veri Setimizin İlk 5 Kaydı:

id	vin	price	miles	stock_no	year	make	model	trim	body_type	vehicle_type	drivetrain	transmission	fuel_type	engine_size	engine_block	seller_name	street	city	state	zip
38b2f52e-8f5d	1GCMGFCF3F1284719	20998.0	115879.0	W1T503169C	2015.0	Chevrolet	Express Cargo	Work Van	Cargo Van	Truck	RWD	Automatic	EB5 / Unleaded	4.8	V	niisan elicott city	8569 Baltimore National Pike	Ellicott City	MD	21043
97ba4955-c40	WBV7Z8C59V887514	27921.0	7339.0	P33243	2018.0	BMW	i3	s	Hatchback	Car	RWD	Automatic	Electric / Premium Unleaded	0.6	I	hendrick honda pompano beach	5381 N Federal Highway	Pompano Beach	FL	33064
be1da6f4-0f04	ML32F4FJ2JH610325	11055.0	39798.0	WM02091A	2018.0	Mitsubishi	Mirage G4	SE	Sedan	Car	FWD	Automatic	Unleaded	1.2	I	rust darrow toyota	2700 West Washington St.	West Bend	WI	53095
84327445-82b6	1GCPTE15K1291189	52997.0	28568.0	9UJ2V425A	2019.0	Chevrolet	Colorado	Z52	Pickup	Truck	4WD	Automatic	Diesel	2.8	I	young kia	308 North Main Street	Layton	UT	84041
cd4691c3-91dd	1G0AL1B007312093	NaN	188485.0	T38625A	2000.0	Pontiac	G3	Base	Coupe	Car	FWD	Automatic	Unleaded	2.2	I	pappas toyota	10011 Spencer Rd	Saint Peters	MO	63376

Scikit-learn'i Tanıyalım:

1. Kullanıcı Dostu Arayüz:

- Scikit-learn, çeşitli makine öğrenimi modelleri için basit ve tutarlı bir API sunar. Bu, kullanıcıların her biri için tamamen yeni bir arayüz öğrenmek zorunda kalmadan farklı algoritmalar arasında geçiş yapmalarını sağlar.

2. Kapsamlı Belgeler:

- Scikit-learn'in kapsamlı ve iyi belgelenmiş fonksiyonları vardır. Bu, kullanıcıların farklı makine öğrenimi tekniklerini ve algoritmalarını nasıl kullanacaklarını anlamaları için mükemmel bir kaynak yapar.

3. Topluluk Desteği:

- Geniş bir kullanıcı kitlesi olan bir kütüphane olması nedeniyle Scikit-learn'in büyük ve aktif bir topluluğu vardır. Bu, çeşitli sorunları halletmiş olan başka birinin bulunduğu bir sorunla karşılaşırsanız çözüm bulma olasılığını artırır.

4. Tutarlı Veri Dönüşümleri:

- Scikit-learn, veri ön işleme için araçlar sağlar. Bu araçlar arasında eksik değerleri ele alma, özellikleri ölçeklendirme ve kategorik değişkenleri kodlama bulunur. Kodunuzda kullandığınız `'train_test_split'` yöntemi de veri setlerini eğitim ve test setlerine bölme konusunda yardımcı olur.

5. Verimli Uygulamalar:

- Scikit-learn, sayısal işlemler için optimize edilmiş NumPy ve SciPy üzerine inşa edilmiştir. Bu, temel hesaplamaların etkili olduğunu ve büyük veri kümeleriyle çalışmak için uygun olduğunu garanti eder.

6. Model Değerlendirme ve Metrikler:

- Scikit-learn, model performansını değerlendirmek için çeşitli metrikler içerir. Sizin kodunuzda kullandığınız ortalama kare hatası (`'mean_squared_error'`) gibi. Ayrıca çapraz doğrulama ve hiperparametre ayarlama için araçlar sunar.

7. Zengin Algoritma Seti:

- Scikit-learn, sınıflandırma, regresyon, kümeleme ve daha fazlası için geniş bir makine öğrenimi algoritma yelpazesini destekler. Bu esneklik, kolayca farklı modellerle deneme yapmanızı sağlar.

8. Diğer Kütüphanelerle Entegrasyon:

- Scikit-learn, Pandas, Matplotlib ve Seaborn gibi diğer popüler Python kütüphaneleriyle iyi entegre olur. Bu, scikit-learn'i mevcut veri analizi ve görselleştirme iş akışlarınıza kolayca entegre etmenizi sağlar.

9. Aktif Geliştirme:

- Scikit-learn aktif olarak bakım ve güncelleme alır, bu da onun Python'un en son sürümleriyle uyumlu olmasını ve makine öğrenimi görevleri için güvenilir bir seçenek olmasını sağlar.

10. Açık Kaynak ve Ücretsiz:

- Scikit-learn açık kaynaklıdır ve ücretsiz olarak kullanılabilir. Bu da onu geniş bir kitleye erişilebilir kılar. Bu, açık bilim ve işbirlikçi geliştirme ilkeleriyle uyumludur.

Scikit-learn'in popülerliğindeki nedenler, kullanıcı dostu arayüzü, kapsamlı belgeleri, topluluk desteği, tutarlı veri dönüşümleri, etkili uygulamalar, model değerlendirme metrikleri, zengin algoritma seti, diğer kütüphanelerle uyumlu olması, aktif geliştirme süreci ve açık kaynak doğasıdır.

Veri Hazırlık Adımları

```
def preprocess_data(self):  
    self.df.dropna(subset=['price'], inplace=True)  
  
    columns_with_mixed_types = ['year']  
    self.df[columns_with_mixed_types] = self.df[columns_with_mixed_types].apply(pd.to_numeric, errors='coerce')  
  
    self.df[self.cat_cols] = self.encoder.fit_transform(self.df[self.cat_cols])  
  
    self.df = self.df.apply(pd.to_numeric, errors='coerce')  
    self.df.dropna(inplace=True)  
  
    print(self.df.head())
```

1. Eksik Değerleri Temizleme:

- o **self.df.dropna(subset=['price'], inplace=True):** Fiyat sütununda eksik değer içeren satırların silinmesi. Fiyat, bağımlı değişken olduğu için bu satırların model eğitiminde kullanılmaması amaçlanmıştır.

2. Yıl (Year) Değişkeninin Sayısal Formata Çevrilmesi:

- o **columns_with_mixed_types = ['year']**
- o **self.df[columns_with_mixed_types] = self.df[columns_with_mixed_types].apply(pd.to_numeric, errors='coerce'):** year değişkeninin içinde sayısal olmayan değerlerin sayısal formata çevrilmesi. 'coerce' parametresi, dönüşüm sırasında hatalı değerlerle karşılaşılması durumunda NaN değeri atanmasını sağlar.

3. Kategorik Değişkenlerin Sayısal Formata Çevrilmesi:

- o **self.df[self.cat_cols] = self.encoder.fit_transform(self.df[self.cat_cols]):** Kategorik değişkenlerin, OrdinalEncoder kullanılarak sayısal formata çevrilmesi. Bu, kategorik değişkenlerin modelde kullanılabilir hale getirilmesini sağlar.

4. Veri Tipi Dönüşümleri ve Eksik Değerleri Temizleme:

- o **self.df = self.df.apply(pd.to_numeric, errors='coerce'):** Veri setindeki tüm değerlerin sayısal formata çevrilmesi. 'coerce' parametresi, dönüşüm sırasında hatalı değerlerle karşılaşılması durumunda NaN değeri atanmasını sağlar.
- o **self.df.dropna(inplace=True):** Kalan eksik değerlerin silinmesi. Model eğitimi ve değerlendirmesi için temiz veri seti elde edilmiş olur.

5. Özetleme:

- o Elde edilen veri setinin başı, **print(self.df.head())** ile gösterilmiştir.

Bu adımlar, veri setimizdeki eksik veya sayısal olmayan değerleri temizlemek ve modele uygun formata getirmek amacıyla gerçekleştirilmiştir. Bu ön işleme adımları, modelin daha güvenilir ve etkili tahminler yapabilmesini sağlamak için uygulanmıştır.

Veri Setini Yükleme

```
def __init__(self, data_path):
    self.data_path = data_path
    self.df = None
    self.cat_cols = ['vin', 'year', 'make', 'model', 'trim', 'body_type', 'vehicle_type',
                     'drivetrain', 'transmission', 'fuel_type', 'engine_size', 'engine_block',
                     'seller_name', 'street', 'city', 'state', 'zip']
    self.encoder = OrdinalEncoder()
    self.X_train, self.X_test, self.y_train, self.y_test = None, None, None, None
    self.model = RandomForestRegressor(n_estimators=100)
    self.y_pred = None
    self.rmse = None
    self.df_pred_actual = None
```

- data_path özelliği, veri setinin dosya yolunu içerir.
- df özelliği, veri setini depolayan bir DataFrame'i temsil eder.
- encoder özelliği, kategorik değişkenleri sayısal formata dönüştürmek için kullanılacak olan OrdinalEncoder sınıfının bir örneğini içerir.
- cat_cols özelliği, kategorik sütun isimlerini içeren bir liste.
- X_train, X_test, y_train, ve y_test özellikleri, veri setini eğitim ve test setlerine ayırmak için kullanılır.
- model özelliği, RandomForestRegressor sınıfından bir örnektir ve başlangıçta 100 ağaç içeren bir Random Forest Regressor modeli ile başlatılır.
- y_pred özelliği, modelin eğitildikten sonra yapılan tahminleri içerir.
- rmse özelliği, tahminlerin gerçek değerlere göre RMSE (Root Mean Squared Error) değerini içerir.
- df_pred_actual özelliği, tahmin edilen ve gerçek değerleri içeren bir DataFrame'i temsil eder.

```
def load_data(self):
    try:
        self.df = pd.read_csv(self.data_path, dtype={'vin': str}, low_memory=False)
        print(self.df.info())
        self.df.drop(columns=['Unnamed: 0'], inplace=True, errors='ignore')
        print(self.df.head())
    except FileNotFoundError:
        print(f"Error: File {self.data_path} not found.")
```

1. Veri Setinin Yüklenmesi:

- pd.read_csv fonksiyonu kullanılarak belirtilen dosya yolundaki CSV dosyası okunur.
- 'vin' sütunu str (string) veri tipine dönüştürülür.
- low_memory=False parametresi ile bellek kullanımı daha etkili hale getirilir.

2. Temel Bilgilerin Kontrol Edilmesi:

- info() metodu kullanılarak veri setinin temel bilgileri ekrana yazdırılır.

- Bu bilgiler arasında sütun sayısı, sütun isimleri, non-null değer sayıları ve veri tipleri yer alır.
- 3. **Gereksiz Sütunların Kaldırılması:**
 - Eğer veri setinde 'Unnamed: 0' adında bir sütun bulunuyorsa, bu sütun drop metodu ile silinir.
 - inplace=True parametresi ile değişiklik veri setine doğrudan uygulanır.
 - errors='ignore' parametresi ile 'Unnamed: 0' sütunu bulunamazsa hata mesajı gösterilmez.
- 4. **Veri Setinin İlk Beş Satırının Görüntülenmesi:**
 - head() metodu kullanılarak veri setinin ilk beş satırı ekrana yazdırılır.
 - Bu adım, veri setinin nasıl görüldüğünü hızlı bir gözden geçirme imkanı sağlar.
- 5. **Hata Kontrolü:**
 - FileNotFoundError durumunda bir hata mesajı ekrana yazdırılır.
 - Bu durum, belirtilen dosya yolunda veri setinin bulunamaması durumunu işaret eder.

Veri setini Bölme

```
def split_data(self):
    self.X_train, self.X_test, self.y_train, self.y_test = train_test_split(
        self.df.drop(columns=['price']),
        self.df['price'],
        test_size=0.2,
        random_state=42
    )
```

1. **Bağımsız ve Bağımlı Değişkenlerin Ayarlanması:**
 - self.df.drop(columns=['price']) ifadesi ile bağımlı değişken olan 'price' sütunu hariç tüm sütunlar bağımsız değişkenleri oluşturur.
 - self.df['price'] ifadesi ile 'price' sütunu bağımlı değişkeni oluşturur.
2. **Veri Setinin Bölünmesi:**
 - train_test_split fonksiyonu kullanılarak veri seti eğitim ve test setlerine ayrılır.
 - test_size=0.2 parametresi ile test setinin oranı belirlenir (yüzde 20).
 - random_state=42 parametresi ile tekrarlanabilirlik sağlanır.
3. **Eğitim ve Test Setlerinin Atanması:**
 - Bağımsız değişkenler (self.X_train ve self.X_test) ve bağımlı değişkenler (self.y_train ve self.y_test) atama işlemleri gerçekleştirilir.

Modelin eğitilmesi

```
def train_model(self):
    self.model.fit(self.X_train, self.y_train)
```

1. **Modelin Eğitilmesi:**

- o `self.model.fit` metodu kullanılarak önceden belirlenmiş model (`self.model`) eğitilir.
- o Eğitim seti (`self.X_train` ve `self.y_train`) üzerinde `fit` metodu çağrılarak model eğitilir.
- o Bu adım, modelin bağımlı değişkeni öğrenmesini sağlar.

Bu fonksiyon, modelin eğitilmesi adımını gerçekleştirir. Modelin başarılı bir şekilde eğitilmesi, veri setindeki desenleri anlamasını ve gelecekteki gözlemleri tahmin etmesini sağlar.

Tahmin Yapılması

```
def predict(self):  
    self.y_pred = self.model.predict(self.X_test)
```

1. Tahmin Yapma:

- o `self.model.predict` metodu kullanılarak eğitilen model (`self.model`) üzerinde tahminler yapılır.
- o Test seti (`self.X_test`) üzerinde `predict` metodu çağrılarak fiyat tahminleri (`self.y_pred`) elde edilir.
- o Bu adım, modelin eğitildikten sonra test setindeki gözlemler için fiyat tahminleri üretmesini sağlar.

Bu fonksiyon sayesinde, model eğitildikten sonra test seti üzerindeki gözlemler için fiyat tahminleri elde edilir. Bu tahminler, modelin performansının değerlendirilmesi ve gerçek değerlerle karşılaştırılması için kullanılır.

Hata Ölçümü

```
def calculate_rmse(self):  
    self.rmse = mean_squared_error(self.y_test, self.y_pred, squared=False)  
    print('RMSE:', self.rmse)
```

1. RMSE Hesaplama:

- o `mean_squared_error` fonksiyonu kullanılarak gerçek fiyatlar (`self.y_test`) ile modelin tahmin ettiği fiyatlar (`self.y_pred`) arasındaki hata hesaplanır.
- o `squared=False` parametresi, RMSE'nin karesiz (kök) olarak hesaplanmasını sağlar.
- o Hesaplanan RMSE değeri, `self.rmse` özelliğine atanır.

2. Sonuçların Yazdırılması:

- o Hesaplanan RMSE değeri, ekrana yazdırılarak kullanıcıya bilgi verilir.

Bu fonksiyon, modelin performansını ölçmek için yaygın olarak kullanılan bir metrik olan RMSE değerini hesaplar. RMSE, modelin tahminlerinin ne kadar doğru olduğunu

değerlendirmek için kullanılır. Düşük RMSE değerleri, modelin daha iyi performans gösterdiğini gösterir.

Sonuçların Analiz Edilmesi

```
def analyze_results(self):
    self.df_pred_actual = pd.DataFrame({
        'Actual Price': self.y_test.values,
        'Predicted Price': self.y_pred
    })
    print(self.df_pred_actual.head(10))
```

1. DataFrame Oluşturma:

- o `pd.DataFrame` kullanılarak bir DataFrame oluşturulur.
- o Bu DataFrame, 'Actual Price' (Gerçek Fiyatlar) ve 'Predicted Price' (Tahmin Edilen Fiyatlar) sütunlarını içerir.
- o Gerçek fiyatlar (`self.y_test.values`) ve tahmin edilen fiyatlar (`self.y_pred`) bu sütunlara atanır.

2. Analiz İçin İlk 10 Gözlem:

- o Oluşturulan DataFrame'in ilk 10 gözlemi, `head(10)` metodu kullanılarak ekrana yazdırılır.
- o Bu adım, gerçek ve tahmin edilen fiyatları karşılaştırarak modelin performansını daha yakından incelemenizi sağlar.

Bu fonksiyon, modelin gerçek ve tahmin edilen fiyatlarını karşılaştırmak ve analiz etmek için kullanılır. Bu tür bir analiz, modelin hangi durumlarda doğru veya yanlış tahminler yaptığını anlamınıza yardımcı olabilir.

Tahminleri Görselleştirme

```
def visualize_predictions(self):
    sns.set_style("darkgrid")
    plt.scatter(self.y_test, self.y_pred)
    plt.plot([min(self.y_test), max(self.y_test)],
             [min(self.y_test), max(self.y_test)], "--", color="red")
    plt.xlim([min(self.y_test), max(self.y_test)])
    plt.ylim([min(self.y_test), max(self.y_test)])
    plt.xlabel("Actual Price")
    plt.ylabel("Predicted Price")
    plt.title("Predicted vs Actual Price (RMSE={:.2f})".format(self.rmse))
    plt.show()
```

1. Görselleştirmenin Ayarlanması:

- o `sns.set_style("darkgrid")` komutu, seaborn kütüphanesini kullanarak grafiklerin stilini belirler.
- 2. **Saçılım Grafiği Oluşturma:**
 - o `plt.scatter` fonksiyonu, gerçek fiyatlar (`self.y_test`) ile tahmin edilen fiyatları (`self.y_pred`) saçılım grafiği üzerinde gösterir.
- 3. **45 Derece Çizgisi Ekleme:**
 - o `plt.plot` fonksiyonu, 45 derece açılı bir kırmızı çizgi ekler. Bu çizgi, mükemmel bir tahmin durumunu temsil eder.
- 4. **Eksen Ayarları:**
 - o `plt.xlim` ve `plt.ylim` fonksiyonları, x ve y eksenini sınırlarını belirler.
- 5. **Eksen Etiketleri:**
 - o `plt.xlabel` ve `plt.ylabel` fonksiyonları, x ve y eksenini etiketlerini belirler.
- 6. **Grafik Başlığı:**
 - o `plt.title` fonksiyonu, grafik başlığını belirler ve RMSE değerini içerir.
- 7. **Görselin Gösterilmesi:**
 - o `plt.show` fonksiyonu, oluşturulan grafiği ekrana getirir.

Bu fonksiyon, modelin gerçek ve tahmin edilen fiyatları arasındaki ilişkiyi görsel olarak değerlendirmenizi sağlar. Saçılım grafiği, modelin performansını daha iyi anlamana ve potansiyel desenleri gözlemlemenize yardımcı olur.

Önemli Özellikleri Görselleştirme

```
def visualize_feature_importance(self, top_n=10):
    feat_importances = pd.Series(self.model.feature_importances_, index=self.X_train.columns)
    top_feats = feat_importances.nlargest(top_n)

    top_feats.plot(kind='barh')
    plt.xlabel("Feature Importance")
    plt.ylabel("Features")
    plt.title(f"Top {top_n} Important Features")
    plt.show()
```

1. **Özellik Önem Düzeylerini Almak:**
 - o `pd.Series` kullanılarak, modelin özellik önem düzeylerini (`self.model.feature_importances_`) ve özellik isimlerini (`self.X_train.columns`) içeren bir pandas Serisi oluşturulur.
2. **En Önemli Özellikleri Seçmek:**
 - o `nlargest` metodu ile en önemli özellikler belirlenir ve `top_n` kadar seçilir.
3. **Çubuk Grafik Oluşturma:**
 - o `top_feats.plot(kind='barh')` komutu, seçilen en önemli özellikleri içeren bir yatay çubuk grafik oluşturur.
4. **Eksen Ayarları ve Başlık:**
 - o `plt.xlabel`, `plt.ylabel`, ve `plt.title` fonksiyonları, çubuk grafik üzerinde eksen etiketleri ve başlık belirler.

5. Görselin Gösterilmesi:

- o `plt.show` fonksiyonu, oluşturulan çubuk grafik görselini ekrana getirir.

Bu fonksiyon, modelin hangi özelliklere daha fazla önem verdiğini anlamak için kullanılır. Özellik önem düzeyleri, modelin eğitilirken hangi özelliklere daha fazla ağırlık verdiğini gösterir.

Model Eğitim ve Değerlendirme Aşamaları

```
model = RegressionModel('us-dealers-used.csv')
model.load_data()
model.preprocess_data()
model.split_data()
model.train_model()
model.predict()
model.calculate_rmse()
model.analyze_results()
model.visualize_predictions()
model.visualize_feature_importance(top_n=21)
```

1. Proje Amaçları ve Kapsamı:

- Bu proje, kullanılmış araçların fiyatlarını tahmin etmek için makine öğrenimi kullanmayı amaçlamaktadır.
- Kullanılan veri seti, ABD'deki satıcıların kullanılmış araç ilanlarını içermektedir.

2. Veri Yüklemesi (`load_data` Fonksiyonu):

- Proje, "us-dealers-used.csv" adlı veri seti ile başlar.
- Veri seti, araç özellikleri, fiyatlar ve satıcı bilgilerini içerir.
- Pandas kütüphanesi kullanılarak veri seti yüklenir ve temel bilgiler ekrana yazdırılır.

3. Veri Ön İşleme (`preprocess_data` Fonksiyonu):

- Veri seti üzerinde ön işleme adımları gerçekleştirilir:
 - o Fiyat bilgisi olmayan satırlar düşürülür.
 - o Karışık veri tiplerine sahip sütunlar (örneğin, 'year') sayısal formata çevrilir.
 - o Kategorik sütunlar Ordinal Encoder kullanılarak sayısallaştırılır.
 - o Tüm veri seti sayısal formata dönüştürülerek eksik değerler temizlenir.

4. Veri Seti Bölme (`split_data` Fonksiyonu):

- Veri seti, eğitim ve test setleri olarak ayrılır.

- `train_test_split` fonksiyonu kullanılarak, eğitim seti (%80) ve test seti (%20) oluşturulur.

5. Model Eğitimi (`train_model` Fonksiyonu):

- Random Forest Regressor algoritması kullanılarak model eğitilir.
- Eğitim seti (`self.X_train` ve `self.y_train`) üzerinde `fit` metodu kullanılarak model eğitilir.

6. Tahmin Yapma (`predict` Fonksiyonu):

- Eğitilen model kullanılarak test seti üzerinde fiyat tahminleri yapılır.
- `self.y_pred` değişkenine tahminler atanır.

7. Hata Ölçümü (`calculate_rmse` Fonksiyonu):

- Mean Squared Error (MSE) kullanılarak Root Mean Squared Error (RMSE) hesaplanır.
- RMSE değeri, modelin tahmin performansını ölçer.

8. Sonuçları Analiz Etme (`analyze_results` Fonksiyonu):

- Gerçek ve tahmin edilen fiyatları içeren bir DataFrame oluşturulur.
- İlk 10 gözlem ekrana yazdırılarak modelin performansı detaylı bir şekilde incelenir.

9. Tahminleri Görselleştirme (`visualize_predictions` Fonksiyonu):

- Scatter plot ile gerçek ve tahmin edilen fiyatlar karşılaştırılır.
- 45 derece kırmızı çizgi, mükemmel tahmin durumunu temsil eder.
- RMSE değeri başlıkta gösterilerek görselleştirmeye ek bilgi sağlanır.

10. Özellik Önemini Görselleştirme (`visualize_feature_importance` Fonksiyonu):

- Modelin belirlediği en önemli özellikler, yatay çubuk grafik ile görsel olarak ifade edilir.
- Toplamda 21 özellik gösterilir, ancak sayı isteğe bağlı olarak değiştirilebilir.

Kod Çıktıları :

```
Data columns (total 21 columns):
#   Column      Dtype
---  -
0   id          object
1   vin          object
2   price        float64
3   miles        float64
4   stock_no     object
5   year         float64
6   make         object
7   model        object
8   trim         object
9   body_type    object
10  vehicle_type object
11  drivetrain   object
12  transmission object
13  fuel_type    object
14  engine_size  float64
15  engine_block object
16  seller_name  object
17  street       object
18  city         object
19  state        object
20  zip          object
```

```
   id      vin      price      miles      stock_no      year      ...      engine_block      seller_name      street      city      state      zip
0   38b2f52e-8f5d  1GCMGFCF3F1284719  20998.0  115879.0  W1T503168C  2015.0  ...      V      nissan ellicott city  8569 Baltimore National Pike  Ellicott City  MD  21043
1   97ba4955-ccf0  WBY7Z8C59JV887514  27921.0  7339.0  P33243  2018.0  ...      I  hendrick honda pompano beach  5381 N Federal Highway  Pompano Beach  FL  33064
2   be1da9fd-0f34  ML32F4FJ2JHF10325  11855.0  39798.0  W42091A  2018.0  ...      I      russ darrow toyota  2700 West Washington St.  West Bend  WI  53095
3   84327e45-6cb6  1GCPTEE15K1291189  52997.0  28568.0  9U2Y425A  2019.0  ...      I      young kia  308 North Main Street  Layton  UT  84041
4   cde691c3-91dd  1G2AL18F087312093  NaN  188485.0  T36625A  2008.0  ...      I      pappas toyota  10011 Spencer Rd  Saint Peters  MO  63376

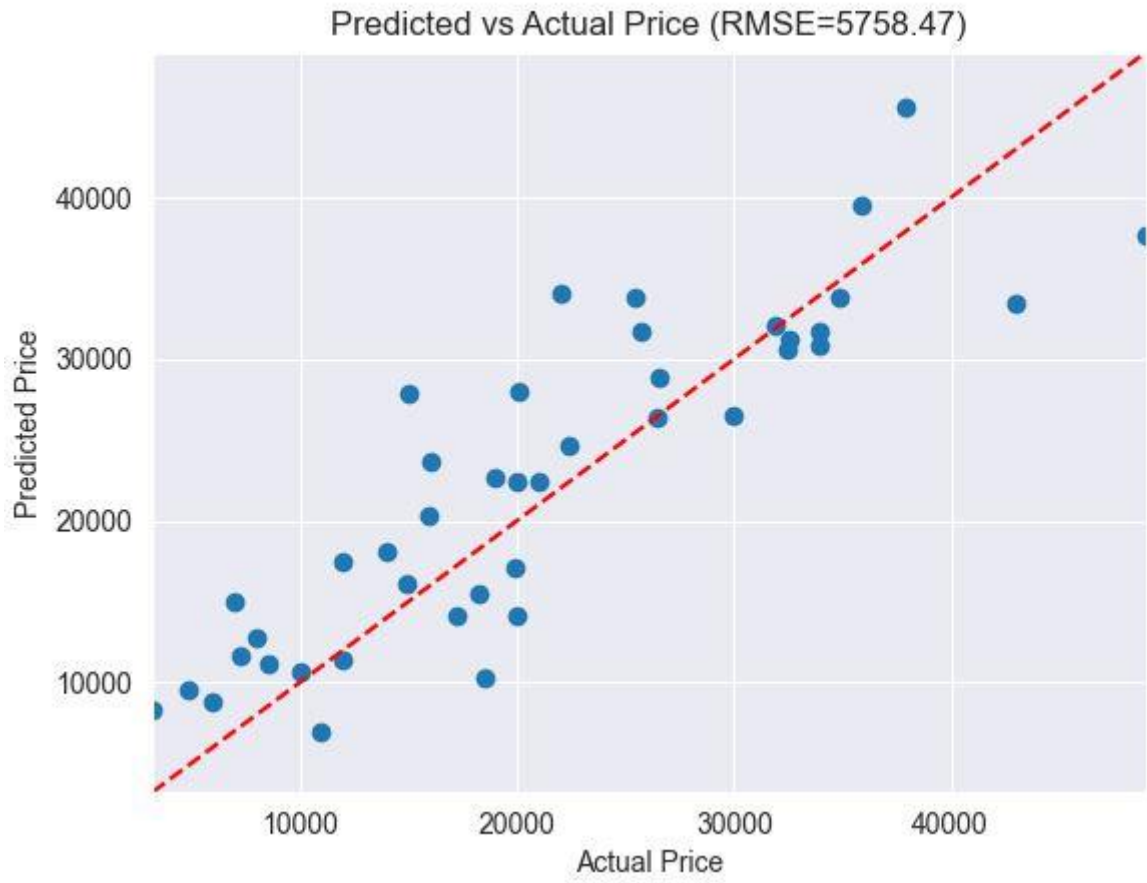
[5 rows x 21 columns]
```

	id	vin	price	miles	stock_no	year	make	model	trim	...	transmission	fuel_type	engine_size	engine_block	seller_name	street	city	state	zip
12997	0.0	536129.0	15999.0	175375.0	1.346240e+05	35.0	9.0	419.0	2154.0	...	0.0	12.0	40.0	2.0	74.0	20914.0	4512.0	12.0	3148.0
48272	0.0	682415.0	6999.0	187325.0	6.980300e+05	27.0	28.0	546.0	1249.0	...	0.0	12.0	39.0	2.0	2790.0	7745.0	5102.0	43.0	1952.0
133477	0.0	1990062.0	25990.0	20110.0	2.000937e+09	38.0	5.0	1162.0	115.0	...	0.0	27.0	12.0	1.0	8594.0	35512.0	1937.0	18.0	5043.0
142270	0.0	1709235.0	27498.0	29753.0	2.141240e+07	37.0	33.0	576.0	96.0	...	0.0	27.0	12.0	1.0	39584.0	12352.0	2198.0	11.0	3566.0
166601	0.0	2067005.0	5495.0	71539.0	3.091000e+03	32.0	60.0	399.0	1129.0	...	0.0	27.0	12.0	1.0	14369.0	18670.0	5414.0	35.0	935.0

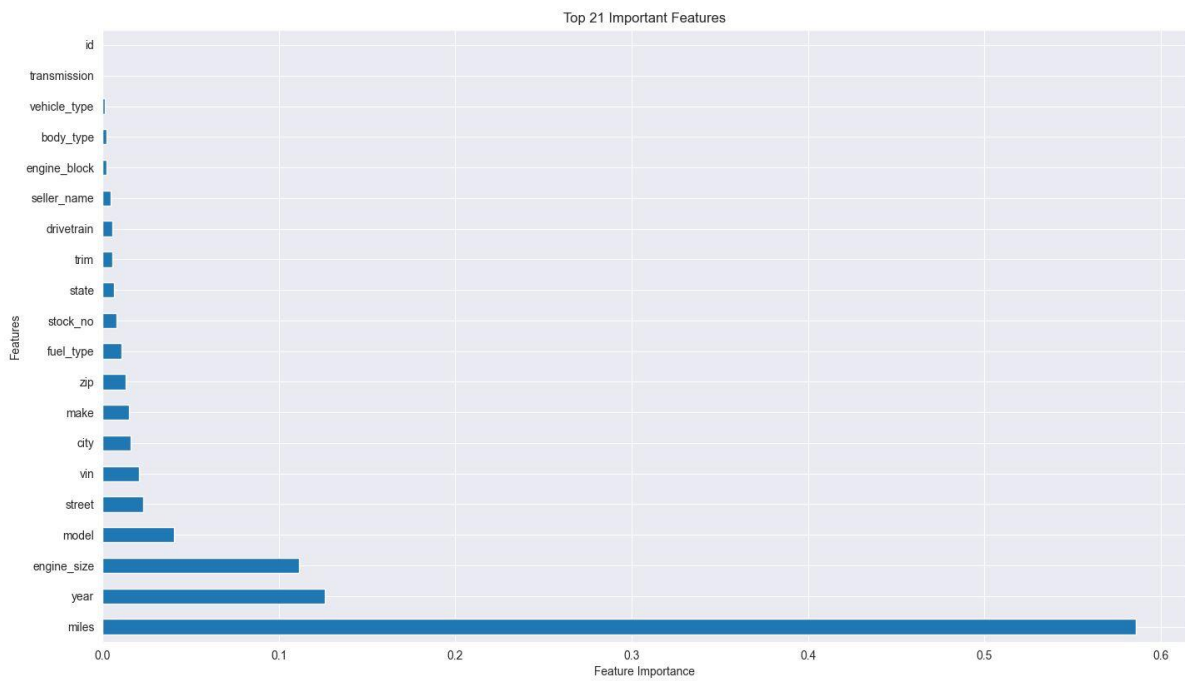
```
RMSE: 5919.716638866825
```

	Actual Price	Predicted Price
0	14900.0	16771.56
1	34888.0	34478.98
2	30000.0	26450.92
3	13995.0	17034.07
4	37897.0	43790.25
5	19999.0	14231.22
6	6999.0	14388.29
7	18499.0	10342.43
8	17200.0	13435.09
9	9998.0	10464.78

Grafikler:



Tahmini ve gerçek fiyat değerleri



En Önemli 21 features

