Web Kazıma Yöntemiyle Toplanan Veriler Üzerinden CatBoost Algoritması ile İkinci El Araç Fiyat Tahmini: İzmir İli Örneği

Adem YAVUZ
Yönetim Bilişim Sistemleri
Dokuz Eylül Üniversitesi
İzmir, Türkiye
adem.yavuz@ogr.deu.edu.tr

Sezer YİĞİT
Yönetim Bilişim Sistemleri
Dokuz Eylül Üniversitesi
İzmir, Türkiye
sezer.yigit@ogr.deu.edu.tr

Oktay ONAR
Yönetim Bilişim Sistemleri
Dokuz Eylül Üniversitesi
İzmir, Türkiye
oktay.onar@ogr.deu.edu.tr

ÖZET

Bu çalışmada, web kazıma yöntemiyle toplanan güncel ikinci el araç ilan verileri kullanılarak, CatBoost algoritması ile araç fiyat tahmini gerçekleştirilmiştir. Çalışma kapsamında, Türkiye'de yaygın olarak kullanılan çevrim içi otomobil ilan platformlarından biri üzerinden İzmir ili özelinde filtrelenmiş 2500 adet benzersiz araç ilanı toplanmıştır. Dinamik web yapısı nedeniyle Puppeteer tabanlı tarayıcı otomasyonu kullanılarak veriler asenkron biçimde elde edilmiş ve çok boyutlu bir veri seti oluşturulmuştur.

Toplanan veri seti; araçların marka, seri, model, üretim yılı, kilometre, yakıt tipi, vites tipi, kasa tipi, motor hacmi, motor gücü ve çekiş tipi gibi hem kategorik hem de sayısal değişkenleri içermektedir. Veri ön işleme aşamasında eksik değerlerin yönetimi, aykırı gözlemlerin temizlenmesi ve öznitelik türetimi gibi işlemler uygulanmıştır. Modelleme sürecinde farklı regresyon algoritmalarının başlangıç performansları karşılaştırılmış ve kategorik değişkenleri doğrudan işleyebilme yeteneği, dengeli genelleme performansı ve düşük hata değerleri nedeniyle CatBoost algoritması nihai model olarak seçilmiştir.

Hiperparametre optimizasyonu sonrasında elde edilen CatBoost modeli, test veri seti üzerinde RMSE = 240.383 TL, MAE = 122.701 TL ve R^2 = 0.9297 performans değerlerine ulaşmıştır. Bu sonuçlar, geliştirilen modelin ikinci el araç fiyatlarındaki varyansın yaklaşık %93'ünü açıklayabildiğini göstermektedir. Çalışma, yalnızca bir modelleme yaklaşımı sunmakla kalmayıp; veri toplama, model eğitimi ve web tabanlı bir uygulama aracılığıyla konuşlandırma süreçlerini içeren uçtan uca bir karar destek sistemi ortaya koyarak literatüre ve pratik uygulamalara katkı sağlamaktadır.

Anahtar Kelimeler: İkinci El Araç Fiyat Tahmini, Web Kazıma, CatBoost, Makine Öğrenmesi, Regresyon Analizi, Karar Destek Sistemleri, Otomotiv Piyasası

1. GİRİŞ (*INTRODUCTION*)

İkinci el araç piyasası, hem bireysel kullanıcılar hem de ticari işletmeler için ekonomik değeri yüksek ve dinamik bir yapıdır. Bir aracın piyasa değerini belirlemek; markası, modeli, yaşı, kilometresi, yakıt tipi ve hasar geçmişi gibi çok sayıda değişkenin bir arada değerlendirilmesini gerektiren karmaşık bir süreçtir. Geleneksel yöntemlerle yapılan fiyatlandırmalar genellikle subjektif kalabilmekte veya piyasanın anlık değişimlerini yakalamakta yetersiz kalmaktadır. Özellikle araç piyasasına derinlemesine hâkim olmayan kullanıcılar için doğru fiyatı belirlemek ciddi bir belirsizlik yaratmaktadır.

Bu çalışmanın temel motivasyonu, bu belirsizliği ortadan kaldırmak amacıyla veri odaklı ve objektif bir fiyat tahminleme mekanizması geliştirmektir. Çalışma kapsamında, manuel veri toplama süreçlerinin zorluklarını aşmak adına modern web kazıma (web scraping) teknikleri kullanılarak İzmir iline ait güncel araç verileri toplanmıştır. Toplanan bu zengin veri seti, kategorik verilerle çalışma başarısı kanıtlanmış olan CatBoost algoritması ile modellenmiştir.

Projenin literatürdeki benzer çalışmalardan ayrılan en önemli yönü, sadece bir modelleme çalışması olmayıp; verinin kaynaktan asenkron olarak çekilmesi, ön işleme tabi tutulması, modelin eğitilmesi ve son aşamada bir uygulama programlama arayüzü (API) aracılığıyla canlıya alınması süreçlerini içeren "uçtan uca" (end-to-end) bir mühendislik çözümü sunmasıdır. Makalenin ilerleyen bölümlerinde veri toplama metodolojisi, kullanılan makine öğrenmesi algoritmaları ve sistemin performans metrikleri detaylı olarak ele alınacaktır.

1.1. Literatür Taraması *(Literature Review)*

Literatürde ikinci el araç fiyatlarının tahmin edilmesi üzerine yapılan çalışmalar incelendiğinde, genellikle istatistiksel yöntemler ve makine öğrenmesi temelli yaklaşımların ağırlıkta olduğu görülmektedir. İlk dönem araştırmalarında, bağımlı ve bağımsız değişkenler arasındaki ilişkileri doğrusal bir düzlemde inceleyen **Çoklu Doğrusal Regresyon (Multiple Linear Regression)** modelleri yaygın

olarak kullanılmıştır **[1]**. Ancak otomobil piyasasının karmaşık yapısı ve verilerin doğrusallıktan uzaklaşması, araştırmacıları daha esnek ve yüksek boyutlu verileri işleyebilen algoritmalara yöneltmiştir.

Bu bağlamda yapılan çalışmalarda, **Rassal Orman (Random Forest)** ve **Destek Vektör Makineleri (Support Vector Machines)** gibi yöntemlerin, özellikle araç yaşı ve kat edilen mesafe (kilometre) gibi temel değişkenler üzerinde yüksek açıklayıcılık oranına sahip olduğu kanıtlanmıştır **[2]**. Bununla birlikte, ikinci el araç ilanlarının doğası gereği barındırdığı "marka", "yakıt tipi" ve "vites tipi" gibi yüksek kardinaliteye sahip kategorik değişkenler, geleneksel algoritmaların bu verileri işlerken veri kaybı yaşamasına veya aşırı öğrenme (overfitting) problemine yol açmasına neden olabilmektedir **[3]**.

Son yıllarda gerçekleştirilen literatür taramaları, **Gradyan Artırıcı Karar Ağaçları (Gradient Boosting Decision Trees - GBDT)** mimarilerinin, yapılandırılmış (tabüler) veriler üzerinde derin öğrenme modellerinden daha tutarlı sonuçlar sergilediğini ortaya koymuştur **[4]**. Bu çalışmada tercih edilen **CatBoost** algoritması, literatürdeki benzerlerinden (XGBoost, LightGBM) farklı olarak kategorik değişkenleri ön işleme aşamasına gerek duymadan "Ordered Boosting" ve simetrik ağaç yapısı ile doğrudan işleyebilmektedir. Bu özellik, hem tahmin doğruluğunu artırmakta hem de modelin genelleme yeteneğini güçlendirmektedir **[5]**.

Son olarak, mevcut çalışmaların büyük bir kısmının Kaggle gibi platformlardan alınan statik ve geçmiş tarihli veri setleri üzerinden yürütüldüğü gözlemlenmiştir. Bu çalışma, **Puppeteer** tabanlı dinamik veri kazıma mimarisi sayesinde literatürdeki "anlık ve bölgesel veriyle tahminleme" ihtiyacına pratik bir çözüm sunarak mevcut literatüre katkı sağlamaktadır.

2. MATERYAL VE YÖNTEM *(MATERIALS AND METHODS)*

2.1. Veri Toplama ve Web Kazıma *(Web Scraping)*

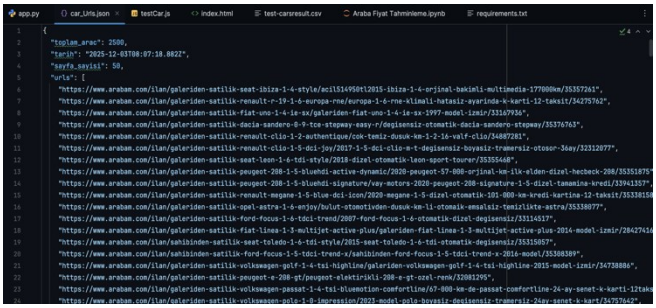
Bu çalışmada kullanılan veri seti, Türkiye'de yaygın olarak kullanılan çevrim içi otomobil ilan platformlarından biri olan **arabam.com** üzerinden elde edilmiştir. Veri toplama süreci, İzmir ili özelinde filtrelenmiş otomobil ilanlarını kapsamaktadır. Hedef platformun dinamik web içeriğine sahip olması nedeniyle, geleneksel statik web kazıma yöntemleri yerine tarayıcı otomasyonuna dayalı bir yaklaşım benimsenmiştir. Bu kapsamda, **Puppeteer** kütüphanesi kullanılarak veri kazıma işlemleri gerçekleştirilmiştir.

Veri toplama süreci, asenkron mimariye sahip iki temel aşamadan oluşmaktadır:

2.1.1. İlan URL'lerinin Hiyerarşik Olarak Toplanması

İlk aşamada, veri setinin temel yapısını oluşturacak olan araç ilan bağlantılarının toplanması hedeflenmiştir. Bu amaçla geliştirilen `collectCarUrl.js` betiği aracılığıyla, platformun arama sonuç sayfaları sistematik biçimde taranmıştır. Node.js çalışma ortamında yürütülen bu süreçte, toplamda 50 farklı arama sonuç sayfası ziyaret edilmiş; her bir sayfadan 50 adet ilan bağlantısı çekilerek **2500 adet benzersiz (unique) araç ilan URL'si** elde edilmiştir.

Toplanan bu bağlantılar, veri tekrarını önlemek amacıyla ayıklanmış ve yapılandırılmış bir JSON formatında `car_Urls.json` dosyasına kaydedilmiştir. Bu dosya, çalışmanın ilerleyen aşamalarında gerçekleştirilen detaylı veri çıkarım sürecinin girdi (input) kaynağını oluşturmuştur.

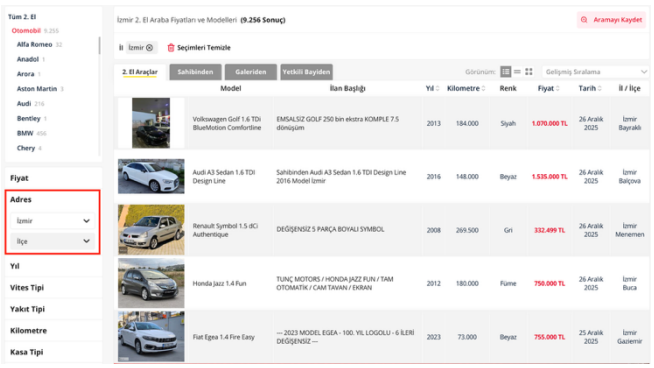


Şekil 1. *Toplanan araç* ***ilan*** *URL'lerinin yapılandırıldığı* `car_Urls.json` *dosyasından örnek bir görünüm.*

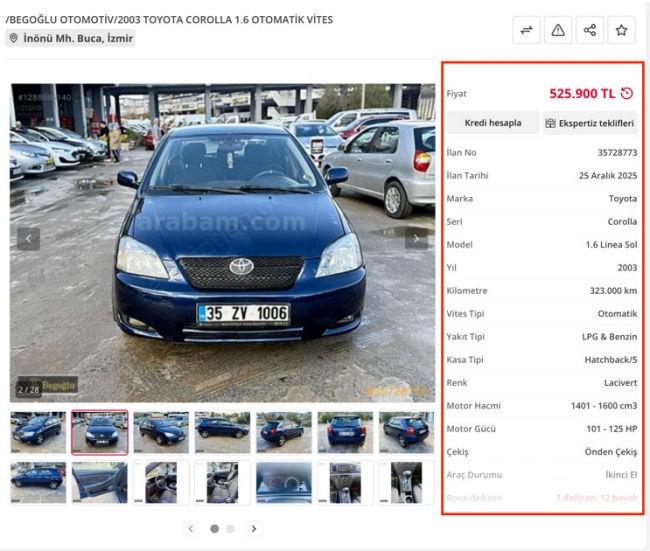
2.1.2. Detaylı Öznitelik Çıkarımı ve DOM Manipülasyonu

İkinci aşamada, elde edilen ilan URL'leri `testCar.js` betiği kullanılarak döngüsel bir işleme tabi tutulmuştur. Her bir URL için tarayıcı üzerinde yeni bir sayfa açılmış ve sayfanın **DOM (Document Object Model)** yapısı üzerinde CSS seçicileri yardımıyla ayrıntılı veri çıkarımı gerçekleştirilmiştir.

Bu aşamada yalnızca fiyat bilgisi değil; aracın markası, modeli, üretim yılı, kilometresi, hasar durumu, yakıt türü, vites tipi ve teknik donanımları gibi **toplam 14 farklı öznitelik (feature)** eş zamanlı olarak toplanmıştır. Böylece, çok boyutlu ve analiz açısından zengin bir veri seti oluşturulmuştur.



Şekil 2. *İzmir ili özelinde filtrelenmiş arama sonuçları sayfalarından toplanan araç ilanlarının hiyerarşik liste yapısını gösteren örnek ekran görüntüsü.*



Şekil 3. *Araç ilan detay sayfasında CSS seçicileri kullanılarak fiyat ve teknik özniteliklerin çıkarıldığı bilgi panelinin örnek görünümü.*

2.1.3. Güvenlik ve Anti-Bot Önlemlerinin Aşılması

Web kazıma sürecinde karşılaşılan en önemli teknik zorluklardan biri, hedef platformda uygulanan bot tespit ve engelleme mekanizmalarıdır. Bu engelleri aşmak ve veri toplama sürecinin sürekliliğini sağlamak amacıyla aşağıdaki stratejiler uygulanmıştır:

- **Stealth Plugin Kullanımı:** `puppeteer-extra-plugin-stealth` eklentisi kullanılarak, tarayıcının otomasyon aracı olduğu bilgisi gizlenmiş ve gerçek bir kullanıcı davranışı simüle edilmiştir.
- **Asenkron Gecikmeler:** Her sayfa yüklemesinden sonra `setTimeout` fonksiyonları ile ortalama 8 saniyelik bekleme süreleri tanımlanmıştır. Bu yaklaşım sayesinde, sunucuya yapılan istek yoğunluğu azaltılmış ve insan benzeri gezinme davranışı elde edilmiştir.
- **Hata Yönetimi:** Olası ağ kopmaları, zaman aşımı veya sayfa yükleme hatalarına karşı `try-catch` blokları kullanılarak, sürecin kesintiye uğramadan devam etmesi sağlanmıştır.

2.2. Veri Seti *(Data Set)*

Bu çalışmada oluşturulan veri seti, ikinci el otomobil fiyat tahmini bağlamında model giriş değişkenleri ile hedef değişkeni kapsamaktadır. Veri seti, otomobil ilanlarından elde edilen **araçlara ait fiziksel, teknik ve kullanım özelliklerini** temsil eden çok boyutlu değişkenlerden meydana gelmektedir. Toplanan veriler aracın özelliklerini açıklayan niteliklerden oluşmakta ve istatistiksel/makine öğrenmesi modelleri için uygun yapıda düzenlenmektedir.

Veri setindeki **bağımsız değişkenler (özellikler / features)** aşağıdaki gibi sınıflandırılmıştır:

- **Tanımlayıcı Bilgiler:** Marka, Seri, Model
- **Kullanım & Performans:** Yıl (üretim yılı), Kilometre
- **Mekanik & Teknik Özellikler:** Vites Tipi, Yakıt Tipi, Motor Hacmi, Motor Gücü, Çekiş Tipi
- **Gövde & Donanım:** Kasa Tipi, Renk
- **Enerji & Tüketim:** Yakıt Deposu Kapasitesi, Ortalama Yakıt Tüketimi

Bu değişkenler, araç fiyatı üzerinde belirleyici olabilecek hem kategorik hem sayısal özellikleri kapsamaktadır. Bağımsız değişkenlere dahil edilen her bir sütun, ilan detay sayfasından doğrudan veya ilgili DOM (Document Object Model) etiketleri üzerinden çıkarılarak veri setine eklenmiştir.

Çalışmanın **hedef değişkeni (dependent variable)**, piyasadaki aracın **satış fiyatı** olup Türk Lirası (TL) cinsinden ifade edilmiştir. Fiyat, tahmin modellerinin çıktısı olarak değerlendirilmiş ve bağımsız değişkenlerle olan ilişkisi modeller aracılığıyla incelenmiştir.

Bu çerçevede, İzmir ili özelinde filtrelenmiş arama sonuçlarından elde edilmiş **2500 adet benzersiz gözlem birimi** veri setini oluşturmuştur. Her bir gözlem, bir araç ilanına karşılık gelmekte ve eksiksiz veri yapısı ile modelleme sürecine uygun hale getirilmiştir.

Tablo 1'de, çalışmada kullanılan veri setine ait bağımsız ve hedef değişkenler ile bu değişkenlerin veri tipleri ve açıklamaları sunulmuştur. Bağımsız değişkenler, araçların fiziksel, teknik ve kullanım özelliklerini temsil ederken; hedef değişken, araç satış fiyatını ifade etmektedir.

Tablo 1. *Veri Setinde Kullanılan Bağımsız ve Hedef Değişkenler, Veri Tipleri ve Açıklamaları*

Değişken Adı	Değişken Türü	Veri Tipi	Açıklama
Marka	Bağımsız	Kategorik	Aracın üretici markası
Seri	Bağımsız	Kategorik	Aracın ait olduğu seri
Model	Bağımsız	Kategorik	Aracın model adı
Yıl	Bağımsız	Sayısal (Integer)	Aracın üretim yılı
Kilometre	Bağımsız	Sayısal (Integer)	Aracın toplam kullanım mesafesi (km)
Vites Tipi	Bağımsız	Kategorik	Aracın vites türü (Manuel, Otomatik vb.)

Değişken Adı	Değişken Türü	Veri Tipi	Açıklama
Yakıt Tipi	Bağımsız	Kategorik	Aracın kullandığı yakıt türü
Kasa Tipi	Bağımsız	Kategorik	Aracın gövde tipi (Sedan, Hatchback vb.)
Renk	Bağımsız	Kategorik	Aracın dış renk bilgisi
Motor Hacmi	Bağımsız	Sayısal (Integer)	Motor silindir hacmi (cc)
Motor Gücü	Bağımsız	Sayısal (Integer)	Motor gücü (HP)
Çekiş Tipi	Bağımsız	Kategorik	Aracın çekiş sistemi (Önden, Arkadan vb.)
Yakıt Deposu	Bağımsız	Sayısal (Integer)	Yakıt deposu kapasitesi (Litre)
Ortalama Yakıt Tüketimi	Bağımsız	Sayısal (Float)	Ortalama yakıt tüketimi (L/100 km)
Fiyat	**Hedef**	Sayısal (Integer)	Araç ilanında belirtilen satış fiyatı (TL)

Bu çalışma kapsamında, belirlenen problem tanımı ve vurgulanan özellikler doğrultusunda **veri madenciliği proje döngüsü (CRISP-DM)** yaklaşımı benimsenmiştir. CRISP-DM süreci; işin ve verinin anlaşılması, veri hazırlama, modelleme, değerlendirme, konuşlandırma ve kontrol aşamalarından oluşmakta olup, çalışmada bu aşamalar sırasıyla uygulanmıştır [6][7].

CRISP-DM yönteminin ilk aşamasında, karşılaşılması muhtemel problemler analiz edilmiş ve ilgili literatür taranarak çalışmada kullanılacak yazılım dilleri, kütüphaneler ve araçlar belirlenmiştir. Bu kapsamda kullanılan yazılım teknolojileri Tablo 2'te sunulmaktadır. İkinci aşamada, elde edilen verinin **kalitesi**, **erişilebilirliği** ve **sürdürülebilirliği** değerlendirilmiş; eksik, tutarsız veya hatalı veriler ön işleme adımlarıyla giderilmiştir.

Üçüncü aşamada, ön işleme süreci sonucunda elde edilen nihai veri seti kullanılarak makine öğrenmesi modelleri oluşturulmuş ve modelleme aşaması gerçekleştirilmiştir. Ardından, belirlenen yöntemler ve analiz yaklaşımları doğrultusunda modeller değerlendirilmiş ve performans ölçütleri üzerinden karşılaştırmalar yapılmıştır.

Değerlendirme aşamasında elde edilen sonuçlar doğrultusunda, kurulan modellerin başarısı ve uygulanabilirliği analiz edilmiştir. Sonuçların doğruluk ve genellenebilirlik düzeyleri incelenerek, ilerleyen çalışmalarda geliştirilecek bir web tabanlı sistem aracılığıyla kullanıcıdan alınan verilerin bu modelle entegre biçimde kullanılabileceği öngörülmüştür.

Tablo 2. *CRISP-DM Süreci Kapsamında Kullanılan Teknolojiler ve Görev Tanımları*

Süreç	Kullanılan Teknolojiler	Tanım / Görev
Veri Toplama (Web Kazıma)	Node.js, Puppeteer, Stealth Plugin	Dinamik web içeriğinden araç ilanlarına ait verilerin asenkron olarak toplanması
Veri Ön İşleme	Python, Pandas, NumPy	Eksik verilerin yönetimi, aykırı değerlerin temizlenmesi ve özellik mühendisliği işlemleri
Modelleme (Makine Öğrenmesi)	CatBoost Regressor, Scikit-learn	Kategorik ve sayısal değişkenleri içeren regresyon modelinin eğitilmesi
Konuşlandırma (Deployment)	FastAPI, Uvicorn, Pydantic	Eğitilen modelin REST API aracılığıyla servis hâline getirilmesi
Geliştirme Ortamı	Jupyter Notebook, Visual Studio Code	Veri analizi, model geliştirme ve kodlama süreçlerinin yürütülmesi

Tablo 2'de, CRISP-DM metodolojisi kapsamında veri toplama, ön işleme, modelleme ve konuşlandırma aşamalarında kullanılan yazılım teknolojileri ve bu teknolojilerin çalışma içerisindeki görev tanımları sunulmuştur.

2.3. Veri Ön İşleme *(Data Preprocessing)*

Web kazıma yöntemiyle elde edilen ham veri seti, doğası gereği eksik değerler, gürültü (noise) ve aykırı gözlemler içermektedir. Bu tür veriler, doğrudan modelleme aşamasında kullanıldığında tahmin performansını olumsuz yönde etkileyebilmektedir. Bu nedenle, modelin genellenebilirliğini ve tahmin tutarlılığını artırmak amacıyla kapsamlı bir veri ön işleme süreci uygulanmıştır. Gerçekleştirilen ön işleme adımları aşağıda ayrıntılı olarak açıklanmaktadır.

2.3.1. Eksik Veri Yönetimi

Eksik verilerin ele alınmasında değişkenin yapısına ve sektörel bilgiye dayalı bir yaklaşım benimsenmiştir:

- **Model** değişkenindeki eksik değerler, ilgili aracın **seri** bilgisi kullanılarak doldurulmuştur.

- **Motor hacmi**, **motor gücü**, **ortalama yakıt tüketimi** ve **yakıt deposu kapasitesi** gibi sayısal değişkenlerdeki eksik değerler, öncelikle aynı **seri** grubunun medyan değeri ile doldurulmuştur. Eğer ilgili seri için yeterli gözlem bulunmuyorsa, değişkenin genel medyan değeri kullanılmıştır.
- **Çekiş tipi** değişkenindeki eksik değerler, ilgili seri içindeki en sık görülen değer (mod) ile tamamlanmıştır. Seri bazında mod değeri bulunmayan durumlarda, piyasa dağılımı göz önünde bulundurularak varsayılan olarak *"Önden Çekiş"* atanmıştır.

Bu yöntemle, hem veri kaybı önlenmiş hem de istatistiksel olarak daha dengeli bir dağılım elde edilmiştir.

```
# Veri setindeki her bir sütun için eksik (NaN) değerlerin sayısını hesaplar
df.isnull().sum()
```

```
Vites Tipi            81
Yakıt Tipi            81
Kasa Tipi             81
Renk                  81
Motor Hacmi          120
Motor Gücü           113
Çekiş                109
Araç Durumu           81
Ort. Yakıt Tüketimi  830
Yakıt Deposu         739
Takasa Uygun         624
Kimden                81
dtype: int64
```

Şekil 4. *Veri setindeki değişkenlere ait eksik (NaN) değerlerin sütun bazında dağılımı.*

Şekil 4' te, `df.isnull().sum()` fonksiyonu kullanılarak veri setinde yer alan her bir değişkende bulunan eksik gözlem sayıları gösterilmektedir. Görselden de anlaşılacağı üzere, özellikle **ortalama yakıt tüketimi** ve **yakıt deposu kapasitesi** gibi sayısal değişkenlerde yüksek sayıda eksik değer bulunmaktadır. Bu durum, veri ön işleme aşamasında değişkenin yapısına ve sektörel bilgiye dayalı uygun doldurma yöntemlerinin uygulanmasını gerekli kılmıştır.

2.3.2. Gereksiz Değişkenlerin Kaldırılması

İlk aşamada, araç fiyatını doğrudan veya dolaylı olarak etkilemediği değerlendirilen değişkenler veri setinden çıkarılmıştır. Bu kapsamda **URL**, **İlan No**, **İlan Tarihi**, **İl**, **İlçe** ve **Takasa Uygun** değişkenleri modelleme sürecine katkı sağlamayacağı ve gürültü oluşturabileceği gerekçesiyle veri setinden silinmiştir. Böylece modelin daha anlamlı özniteliklere odaklanması sağlanmıştır.

2.3.3. Yinelenen Kayıtların Temizlenmesi

Veri setinde bulunan yinelenen gözlemler incelenmiş ve tespit edilen mükerrer kayıtlar ilk gözlem korunacak şekilde veri setinden çıkarılmıştır. Bu adım, modelin aynı bilgiyi birden fazla kez öğrenerek yanlı tahminler üretmesini önlemek amacıyla gerçekleştirilmiştir.

2.3.4. Öznitelik Türetimi (Feature Engineering)

Veri setinde yer alan **Yıl** değişkeni aracın üretim yılını ifade etmektedir. Araç fiyatı üzerinde daha açıklayıcı bir etkiye sahip olduğu düşünülen **araç yaşı** değişkeni, mevcut yıl ile üretim yılı arasındaki fark alınarak türetilmiştir. Bu işlem sonrasında **Yıl** değişkeni veri setinden kaldırılmıştır. Bu yaklaşım, modelin araçların değer kaybını daha doğru şekilde öğrenmesine katkı sağlamaktadır.

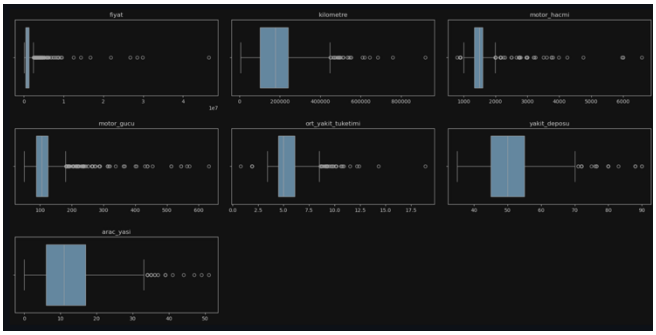
2.3.5. Veri Tipi Düzenlemeleri ve Kategorik Değişkenler

Veri setindeki değişkenlerin veri tipleri kontrol edilerek gerekli dönüşümler yapılmıştır. Kategorik değişkenler için klasik kodlama yöntemleri yerine, kategorik verileri doğrudan işleyebilen **CatBoost** algoritmasının yapısından faydalanılmıştır. Bu nedenle, kategorik değişkenler ayrıca sayısal formata dönüştürülmemiş; ilgili değişkenler model eğitim aşamasında **kategorik öznitelik** olarak tanımlanmıştır. Bu yaklaşım, bilgi kaybını önleyerek model performansını artırmıştır.

2.3.6. Aykırı Değerlerin Temizlenmesi

Araç fiyatı ve kilometre değişkenleri üzerinde yapılan incelemelerde, piyasa gerçekleriyle uyumsuz bazı gözlemler tespit edilmiştir. Örneğin; olağan dışı düşük fiyatlı lüks araçlar veya hatalı girişlerden kaynaklanan aşırı yüksek kilometre değerleri aykırı gözlem olarak değerlendirilmiştir.

Aykırı değerlerin belirlenmesinde **Kutu Grafiği (Box Plot)** analizi ve **Çeyrekler Arası Aralık (Interquartile Range – IQR)** yöntemi kullanılmıştır. Bu yönteme göre, alt ve üst sınırların dışında kalan gözlemler veri setinden çıkarılmıştır. Böylece, modelin aşırı uç değerlere bağlı olarak yanlılık (bias) geliştirmesi engellenmiş ve daha dengeli bir veri dağılımı elde edilmiştir.



Şekil 5. *İkinci el araç veri setinde yer alan sayısal değişkenlere (fiyat, kilometre, motor hacmi, motor gücü, ortalama yakıt tüketimi, yakıt deposu ve araç yaşı) ait kutu grafikleri (boxplot) ve aykırı gözlem dağılımları.*

2.4. **Modelleme** *(Modelling)*

Bu çalışmada araç fiyat tahmini problemi, denetimli öğrenme kapsamında bir regresyon problemi olarak ele alınmıştır. Modelleme sürecinde, farklı algoritmaların veri seti üzerindeki öğrenme kapasitelerini karşılaştırmak ve en uygun modeli belirlemek amacıyla geniş bir model havuzu oluşturulmuştur. Bu havuzda hem doğrusal regresyon temelli yöntemler hem de ağaç tabanlı ve topluluk (ensemble) öğrenme algoritmaları yer almaktadır.

Modelleme sürecine dahil edilen algoritmalar aşağıdaki şekilde gruplandırılabilir:

- **Doğrusal modeller:** Linear Regression, Ridge, Lasso ve ElasticNet
- **Ağaç tabanlı modeller:** Decision Tree ve Random Forest
- **Topluluk ve gradyan artırma yöntemleri:** AdaBoost, Gradient Boosting, XGBoost, LightGBM ve CatBoost
- **Mesafe tabanlı ve kernel yöntemleri:** K-Nearest Neighbors (KNN) ve Support Vector Regression (SVR)

Bu çeşitlilik, farklı model ailelerinin veri setindeki doğrusal ve doğrusal olmayan ilişkileri ne ölçüde yakalayabildiğini gözlemleyebilmek amacıyla tercih edilmiştir. Tüm modeller, adil bir karşılaştırma sağlamak amacıyla aynı eğitim ve test veri setleri üzerinde eğitilmiş ve değerlendirilmiştir.

Model karşılaştırma sürecinin ilk aşamasında, tüm algoritmalar **herhangi bir hiperparametre optimizasyonu uygulanmadan**, varsayılan parametre değerleriyle eğitilmiştir. Bu yaklaşım, modellerin veri seti üzerindeki doğal performanslarının (baseline performans) gözlemlenmesini ve daha sonraki optimizasyon adımlarının etkisinin net biçimde analiz edilmesini amaçlamaktadır.

2.4.1. Başlangıç (Baseline) Model Performanslarının Karşılaştırılması

Başlangıç modelleme aşamasında elde edilen sonuçlar, modellerin eğitim ve test veri setleri üzerindeki hata metrikleri (MSE, MAE, RMSE) ve açıklayıcılık ölçütü (R^2) kullanılarak değerlendirilmiştir. Elde edilen performans değerleri, modellerin genelleme yeteneklerini ve aşırı öğrenme eğilimlerini analiz etmek amacıyla birlikte yorumlanmıştır.

Sonuçlar incelendiğinde, doğrusal regresyon temelli modellerin test verisi üzerinde görece düşük R^2 değerleri ürettiği görülmektedir. Bu durum, araç fiyatlarını etkileyen değişkenler arasındaki ilişkilerin büyük ölçüde doğrusal olmayan bir yapıya sahip olduğunu ve doğrusal modellerin bu karmaşıklığı yeterince temsil edemediğini göstermektedir. Özellikle Lasso ve ElasticNet modellerinde test performansının oldukça zayıf olması dikkat çekmektedir.

Ağaç tabanlı ve gradyan artırma temelli yöntemler ise doğrusal modellere kıyasla belirgin biçimde daha başarılı sonuçlar üretmiştir. Gradient Boosting, LightGBM ve CatBoost modelleri test veri setinde daha düşük hata değerleri ve daha yüksek R^2 skorları elde etmiştir. Bu durum, topluluk öğrenme yaklaşımlarının veri setindeki karmaşık örüntüleri daha etkin biçimde yakalayabildiğini ortaya koymaktadır.

Bununla birlikte, bazı modellerde eğitim ve test performansları arasında belirgin farklar gözlemlenmiştir. Özellikle Decision Tree, Random Forest ve XGBoost modellerinde eğitim R^2 değerlerinin çok yüksek olmasına karşın test R^2 değerlerinin daha düşük seviyelerde kalması, bu modellerin varsayılan parametrelerle aşırı öğrenme eğilimi gösterdiğini düşündürmektedir.

CatBoost modeli ise hiperparametre ayarlaması yapılmadan dahi dengeli bir performans sergilemiştir. Eğitim ve test hata metrikleri arasındaki farkın sınırlı olması, modelin güçlü bir genelleme yeteneğine sahip olduğunu göstermektedir. Ayrıca CatBoost'un kategorik değişkenleri doğrudan işleyebilme yeteneği, ek kodlama adımlarına ihtiyaç duyulmamasını sağlamış ve modelleme sürecini sadeleştirmiştir.

Bu bulgular doğrultusunda, başlangıç performansları, genelleme kabiliyeti ve veri setinin yapısına uyumluluğu dikkate alınarak CatBoost algoritması nihai model için en uygun aday olarak belirlenmiştir. Bir sonraki aşamada, bu model üzerinde hiperparametre optimizasyonu gerçekleştirilmiştir.

Tablo 3. Eğitim (Train) Veri Seti Üzerindeki Başlangıç Model Performansları

(MSE: $\times 10^9$, MAE & RMSE: $\times 10^3$)

Model	Train MSE	Train MAE	Train RMSE	Train R^2
GradientBoostingRegressor	34.25	103.6	185.1	0.97
CatBoostRegressor	8.17	57.2	90.4	0.99
LinearRegression	439.80	170.2	663.2	0.64
Ridge	435.31	170.0	659.8	0.64
LGBMRegressor	33.70	71.9	183.6	0.97
XGBRegressor	1.47	24.5	38.4	1.00
RandomForestRegressor	24.82	49.7	157.5	0.98
SVR	154.44	105.5	393.0	0.87
AdaBoostRegressor	156.35	186.0	395.4	0.87
KNeighborsRegressor	192.99	149.4	439.3	0.84
DecisionTreeRegressor	0.00	0.06	0.95	1.00
ElasticNet	869.96	498.0	932.7	0.29
Lasso	1296.52	570.2	1138.6	-0.06

Tablo 3'te , hiperparametre optimizasyonu uygulanmadan önce farklı regresyon modellerinin eğitim veri seti üzerindeki başlangıç performansları sunulmaktadır.

Tablo 4. Test Veri Seti Üzerindeki Başlangıç Model Performansları

(MSE: ×10⁹, MAE & RMSE: ×10³)

Model	Test MSE	Test MAE	Test RMSE	Test R^2
GradientBoostingRegressor	118.33	146.1	344.0	0.86
CatBoostRegressor	121.24	127.6	348.2	0.85
LinearRegression	129.26	160.9	359.5	0.84
Ridge	129.73	161.1	360.2	0.84
LGBMRegressor	151.17	138.7	388.8	0.82
XGBRegressor	192.20	151.8	438.4	0.77
RandomForestRegressor	195.85	153.1	442.6	0.76
SVR	214.09	145.0	462.7	0.74
AdaBoostRegressor	242.20	209.0	492.1	0.71
KNeighborsRegressor	346.46	209.0	588.6	0.58
DecisionTreeRegressor	582.29	238.5	763.1	0.29
ElasticNet	801.01	494.8	895.0	0.03
Lasso	901.02	538.3	949.2	-0.10

Modellerin test veri seti üzerindeki başlangıç performansları Tablo 4'te sunulmuştur.

2.4.2. CatBoost Algoritmasının Temel Özellikleri

CatBoost (Categorical Boosting), gradyan artırma (gradient boosting) yaklaşımına dayanan ve özellikle kategorik değişkenlerin etkin biçimde işlenmesi amacıyla geliştirilmiş bir makine öğrenmesi algoritmasıdır. Geleneksel gradyan artırma yöntemlerinde kategorik değişkenlerin modele dahil edilebilmesi için önceden çeşitli kodlama tekniklerinin (örneğin one-hot encoding veya label encoding) uygulanması gerekmektedir. Bu tür ön işlemler, veri boyutunun artmasına ve bazı durumlarda bilgi kaybına yol açabilmektedir. CatBoost ise kategorik değişkenleri doğrudan işleyebilme yeteneği sayesinde bu sınırlamaları büyük ölçüde ortadan kaldırmaktadır.

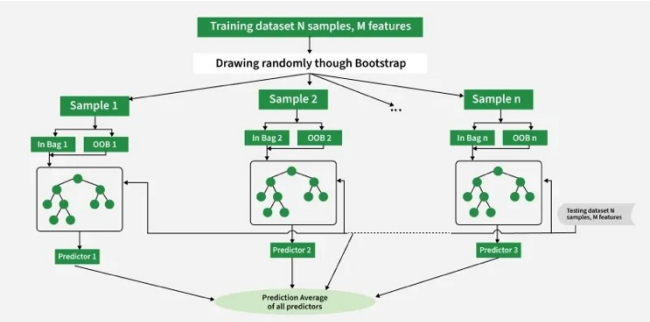
CatBoost algoritmasının temelinde, bootstrap örnekleme yöntemi ile oluşturulan çoklu karar ağaçlarından oluşan bir **ensemble öğrenme yapısı** bulunmaktadır. Eğitim verisinden rastgele örnekler alınarak oluşturulan bu yapı sayesinde model, farklı alt örnekler üzerinde öğrenme gerçekleştirmekte ve daha sağlam tahminler üretebilmektedir. CatBoost'un ensemble yapısı ve çoklu karar ağaçlarının birlikte çalışma prensibi Şekil 6'te şematik olarak gösterilmektedir.

Algoritmanın en önemli yeniliklerinden biri, kategorik değişkenler için kullanılan **hedefe dayalı istatistiksel kodlama (target-based encoding)** yaklaşımıdır. Bu yöntemde kategorik değişkenler, hedef değişkenle olan ilişkileri dikkate alınarak sayısal temsillere dönüştürülmektedir. Ancak klasik hedef kodlama yaklaşımlarında karşılaşılan **veri sızıntısı (target leakage)** problemi, CatBoost'ta geliştirilen **ordered boosting** mekanizması ile giderilmektedir. Ordered boosting yaklaşımında, her bir gözlem için hesaplanan istatistikler yalnızca kendisinden önceki gözlemler kullanılarak elde edilmekte; böylece modelin geleceğe ait bilgileri öğrenmesi engellenmektedir. Bu süreç, gradyan artırma adımlarının ardışık biçimde hata minimizasyonu sağlamasını mümkün kılmakta olup, bu mekanizma Şekil 8'de görsel olarak sunulmaktadır.

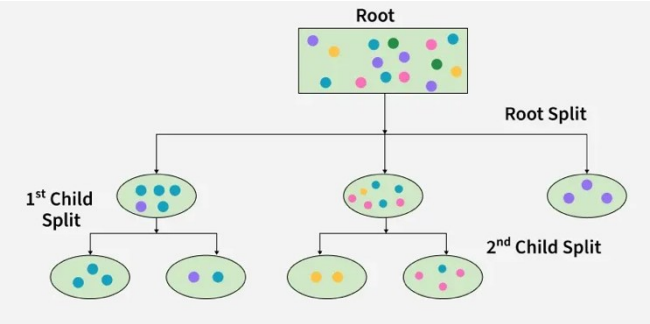
CatBoost'un bir diğer ayırt edici özelliği, **simetrik (oblivious) karar ağaçları** kullanmasıdır. Bu ağaç yapısında, ağacın her seviyesinde aynı bölme kuralı uygulanmakta ve böylece modelin yapısal karmaşıklığı kontrol altında tutulmaktadır. Simetrik karar ağaçlarının bölünme yapısı ve hiyerarşik organizasyonu Şekil 7'te gösterilmektedir. Bu yapı, modelin hem eğitim sürecinin daha kararlı olmasını sağlamakta hem de genelleme yeteneğini artırmaktadır.

CatBoost'un sahip olduğu bu yapısal ve algoritmik özellikler, varsayılan hiperparametrelerle dahi dengeli ve güçlü bir performans elde edilmesine olanak tanımaktadır. Özellikle çok sayıda kategorik değişken içeren ve doğrusal olmayan ilişkilerin baskın olduğu veri setlerinde, CatBoost manuel ön işleme ihtiyacını azaltarak modelleme sürecini sadeleştirmekte ve tahmin başarısını artırmaktadır.

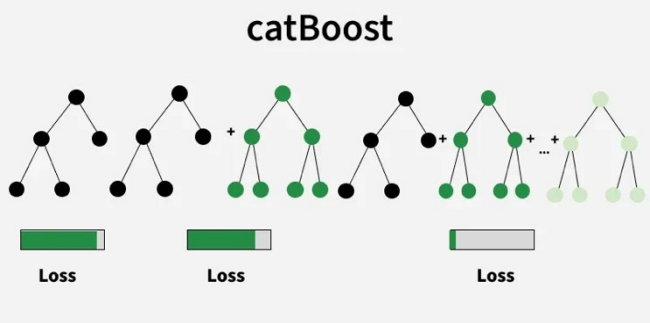
Bu çalışmada kullanılan araç fiyat veri seti; marka, seri, model, yakıt tipi, vites tipi, kasa tipi ve çekiş türü gibi çok sayıda kategorik değişken içermektedir. Baseline modelleme sonuçları da dikkate alındığında, CatBoost'un diğer algoritmalara kıyasla daha dengeli bir eğitim–test performansı sergilediği ve genelleme kabiliyetinin daha yüksek olduğu gözlemlenmiştir. Bu nedenlerle, çalışmanın devamında CatBoost algoritması nihai model olarak seçilmiş ve performansının artırılması amacıyla hiperparametre optimizasyonu aşamasına geçilmiştir.



Şekil 6. *CatBoost algoritmasında bootstrap örnekleme ile oluşturulan çoklu karar ağaçlarının ensemble yapısı. Kaynak: GeeksforGeeks, "CatBoost ML".*



Şekil 7. *CatBoost modelinde kullanılan simetrik karar ağaçlarının (oblivious trees) bölünme yapısının şematik gösterimi.*
Kaynak: GeeksforGeeks, "CatBoost ML".



Şekil 8. *CatBoost algoritmasında gradyan artırma sürecinin ardışık ağaçlar üzerinden kayıp fonksiyonunu azaltma prensibi.*
Kaynak: GeeksforGeeks, "CatBoost ML".

2.4.3. CatBoost Modeli için Hiperparametre Optimizasyonu

Makine öğrenmesi modellerinin tahmin performansı, algoritmanın öğrenme sürecini yönlendiren hiperparametrelerin doğru biçimde yapılandırılmasına doğrudan bağlıdır. Bu nedenle, bu çalışmada CatBoost modelinin genelleme yeteneğini artırmak ve tahmin hatalarını (MAE ve RMSE) minimize etmek amacıyla sistematik bir hiperparametre optimizasyonu süreci yürütülmüştür.

2.4.3.1. Optimize Edilen Hiperparametre Seti

Modelin eğitim sürecinde kritik rol oynayan ve performans üzerinde doğrudan etkisi bulunan temel hiperparametreler belirlenmiş ve optimize edilmiştir. Bu parametreler aşağıda özetlenmektedir:

- **İterasyon Sayısı (Iterations):** Modelin oluşturacağı toplam karar ağacı sayısını ifade etmektedir. Bu çalışmada iterasyon sayısı 1000 olarak belirlenmiş olup, modelin araç fiyatlarını etkileyen karmaşık ve doğrusal olmayan ilişkileri öğrenebilmesi için yeterli model kapasitesi sağlamaktadır.
- **Öğrenme Oranı (Learning Rate):** Her bir ağacın nihai modele katkı düzeyini belirleyen bu parametre, modelin öğrenme hızını kontrol etmektedir. Öğrenme oranı 0.1 olarak seçilmiş; bu değer, hızlı yakınsama sağlarken aşırı öğrenme riskini sınırlamak amacıyla tercih edilmiştir.
- **Ağaç Derinliği (Depth):** Karar ağaçlarının maksimum dallanma seviyesini belirleyen bu parametre, modelin karmaşıklığını doğrudan etkilemektedir. Aşırı yüksek derinlik değerlerinin overfitting'e yol açabileceği göz önünde bulundurularak, veri setinin hacmi ve yapısına uygun bir derinlik seviyesi belirlenmiştir.
- **L2 Regülarizasyonu (L2_leaf_reg):** Yaprak düğümlerindeki ağırlıkların büyüklüğünü kontrol ederek model karmaşıklığını sınırlandıran bu parametre, aşırı öğrenmenin önlenmesi amacıyla yapılandırılmıştır.

2.4.3.2. Optimizasyon Yöntemi

Hiperparametre optimizasyonu sürecinde, belirlenen parametre uzayındaki en uygun kombinasyonu tespit etmek amacıyla **Grid Search (ızgara araması)** yöntemi kullanılmıştır. Bu yöntemde, tanımlanan hiperparametre aralıklarındaki tüm olası kombinasyonlar sistematik olarak denenmiş ve her bir kombinasyon çapraz doğrulama (cross-validation) sonuçlarına göre değerlendirilmiştir. Elde edilen sonuçlar arasından, en düşük hata değerlerini ve en kararlı genelleme performansını sunan hiperparametre seti nihai model için seçilmiştir.

2.4.3.3. Aşırı Öğrenmenin (Overfitting) Kontrolü

Modelin yalnızca eğitim verisini ezberlemesini önlemek ve genelleme kabiliyetini artırmak amacıyla, CatBoost algoritmasının dahili **aşırı öğrenme belirleyici (overfitting detector)** mekanizmasından yararlanılmıştır. Eğitim sürecinde, doğrulama verisi üzerindeki hata belirli bir iterasyon boyunca iyileşme göstermediğinde, eğitim otomatik olarak durdurulmuş (early stopping) ve en iyi performansın elde edildiği ağaç yapısı korunmuştur. Bu yaklaşım sayesinde, modelin eğitim ve test performansları arasındaki fark minimize edilerek daha dengeli ve güvenilir bir tahmin yapısı elde edilmiştir.

2.5. Değerlendirme *(Evaluation)*

Bu bölümde, geliştirilen ikinci el araç fiyat tahminleme modelinin performans sonuçları ve elde edilen bulgular detaylı biçimde analiz edilmektedir. Modelin tahmin doğruluğu, genelleme kapasitesi ve hata davranışı; nicel performans metrikleri ve istatistiksel değerlendirmeler aracılığıyla ele alınmıştır.

Modelin genelleme yeteneğinin güvenilir bir şekilde ölçülebilmesi amacıyla veri seti, **%80 eğitim** ve **%20 test** olacak biçimde iki bağımsız kümeye ayrılmıştır. Bu oran, literatürde regresyon problemleri için yaygın olarak kullanılan bir veri bölünme stratejisi olup, modelin yeterli miktarda veriyle öğrenmesini sağlarken aynı zamanda daha

önce görülmemiş veriler üzerinde objektif bir performans değerlendirmesi yapılmasına olanak tanımaktadır. Veri ön işleme, özellik mühendisliği ve hiperparametre optimizasyonu adımlarının tamamlanmasının ardından CatBoost modeli yalnızca eğitim verisi üzerinde eğitilmiş, modelin başarımı test veri seti üzerinden raporlanmıştır.

Regresyon modelinin tahmin performansını değerlendirmek amacıyla **Ortalama Mutlak Hata (MAE)**, **Kök Ortalama Kare Hata (RMSE)** ve **Belirlilik Katsayısı (R^2)** metrikleri kullanılmıştır. Bu metrikler, tahmin edilen değerler ile gerçek değerler arasındaki sapmayı farklı açılardan ölçerek modelin başarımını bütüncül biçimde değerlendirmeye imkân tanımaktadır.

Ortalama Mutlak Hata (MAE), tahmin edilen değerler ile gerçek değerler arasındaki mutlak farkların ortalamasını ifade etmekte olup aşağıdaki şekilde hesaplanmaktadır:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \quad | Y_i - \hat{Y}_i |$$

Kök Ortalama Kare Hata (RMSE), tahmin hatalarının karelerinin ortalamasının karekökü alınarak hesaplanmakta ve büyük tahmin hatalarına daha duyarlı bir metrik olarak kullanılmaktadır:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \quad (Y_i - \hat{Y}_i)^2}$$

Bu denklemlerde n, toplam gözlem sayısını; $Y_i$ve $\hat{Y}_i$ise sırasıyla i'inci gözleme ait gerçek ve tahmin edilen değerleri temsil etmektedir. MAE ve RMSE değerlerinin daha küçük olması, modelin tahmin performansının daha başarılı olduğunu göstermektedir.

Belirlilik Katsayısı (R^2), bağımlı değişkendeki toplam varyansın ne kadarının model tarafından açıklandığını ifade eden bir ölçüt olup aşağıdaki şekilde tanımlanmaktadır:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \quad (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n} \quad (Y_i - \bar{Y})^2}$$

Burada $\bar{Y}$, gerçek değerlerin ortalamasını ifade etmektedir. R^2 değerinin 1'e yaklaşması, modelin veriye olan uyumunun arttığını ve tahmin gücünün yüksek olduğunu göstermektedir.

Hiperparametre optimizasyonu tamamlanmış CatBoost modelinin test veri seti üzerindeki performansı incelendiğinde, **RMSE = 240.383 TL**, **MAE = 122.701 TL** ve **R^2 = 0.9297** değerleri elde edilmiştir. Bu sonuçlar, modelin ikinci el araç fiyatlarındaki değişkenliğin yaklaşık **%92.97'sini** açıklayabildiğini ve düşük hata paylarıyla güvenilir tahminler üretebildiğini göstermektedir. Özellikle yüksek R^2 değeri, geliştirilen modelin piyasa dinamiklerini başarılı biçimde öğrendiğini ve fiyat oluşum mekanizmasını büyük ölçüde yakalayabildiğini ortaya koymaktadır.

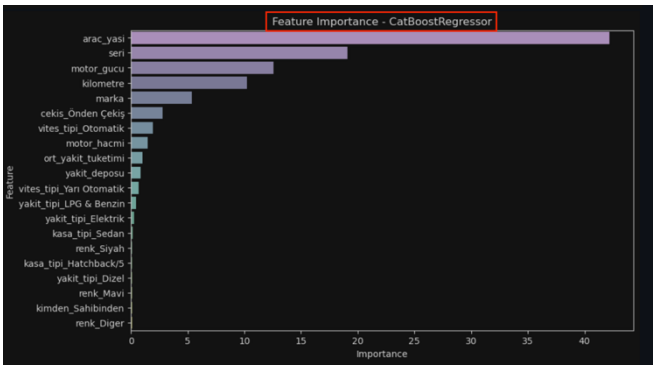
Tablo 4. *CatBoost Modeli için Optimizasyon Öncesi–Sonrası Performans Karşılaştırması (Test)*

Model	Test RMSE (TL)	Test MAE (TL)	Test R^2
CatBoost (Baseline)	348,199	127,582	0.85
CatBoost (Optimize)	240,383	122,701	0.9297

CatBoost modelinin hiperparametre optimizasyonu öncesi ve sonrası test performansları Tablo 4'de karşılaştırmalı olarak sunulmuştur.

Hata davranışı incelendiğinde, modelin düşük ve orta segment araçlarda oldukça yüksek tahmin doğruluğu sergilediği; buna karşılık lüks segment veya nadir bulunan araçlarda hata değerlerinin görece arttığı gözlemlenmiştir. Bu durum, söz konusu araçlara ait veri sayısının sınırlı olması ve fiyat dağılımındaki uç değerlerin (outliers) model üzerinde daha baskın etki oluşturması ile açıklanabilir. Bununla birlikte, genel hata seviyelerinin düşük olması ve yüksek açıklayıcılık oranı, modelin pratik uygulamalarda bir **karar destek sistemi** olarak kullanılabileceğini göstermektedir.

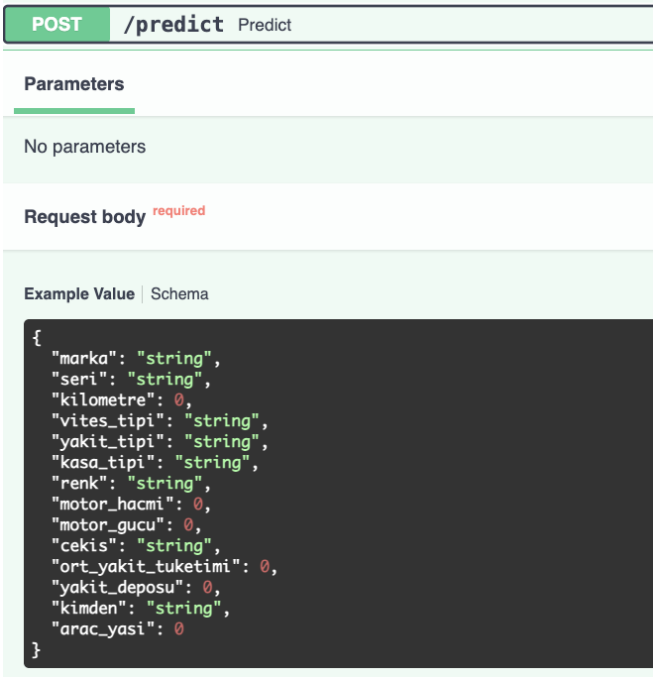
Modelin karar verme sürecinin daha iyi anlaşılabilmesi amacıyla, hiperparametre optimizasyonu sonrasında elde edilen **özellik önem sıralamaları (feature importance)** analiz edilmiştir. Bu analiz, araç yaşı, kilometre, motor gücü ve seri gibi değişkenlerin fiyat tahmininde belirleyici rol oynadığını ortaya koymaktadır. Elde edilen bulgular, ikinci el araç piyasasına ilişkin ekonomik ve sektörel gerçeklerle yüksek düzeyde uyum göstermekte olup, geliştirilen modelin yalnızca yüksek doğruluk sunmakla kalmayıp aynı zamanda **açıklanabilir yapay zeka (Explainable AI – XAI)** perspektifine de katkı sağladığını göstermektedir.



Şekil 9. *Hiperparametre optimizasyonu sonrası CatBoost regresyon modeli için hesaplanan öznitelik önem (feature importance) değerleri.*

2.6. Web Tabanlı Araç Fiyat Tahminleme Uygulaması

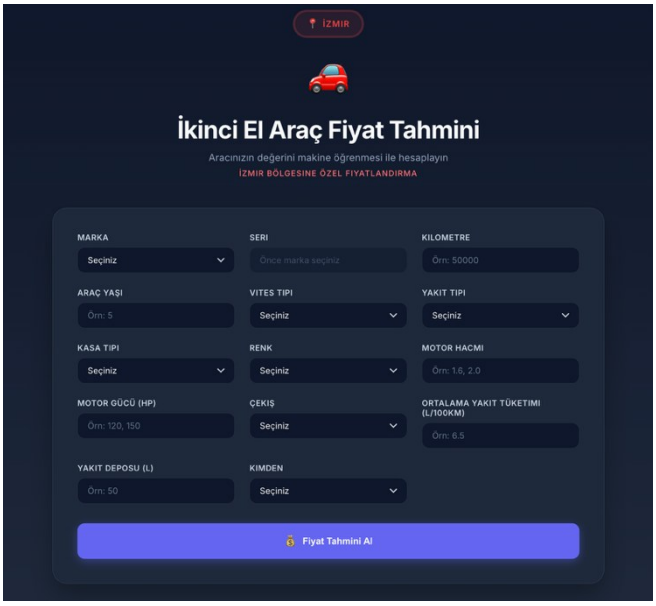
Bu çalışma kapsamında geliştirilen makine öğrenmesi modeli, son kullanıcıların kolaylıkla erişebileceği web tabanlı bir tahminleme uygulamasına entegre edilmiştir. Uygulama, uçtan uca bir yazılım mimarisi üzerine inşa edilmiş olup modern REST API yaklaşımını temel almaktadır. Arka planda FastAPI tabanlı bir sunucu çalışmakta ve eğitilmiş CatBoost modeli (best_model.cbm) sistem belleğinde hazır tutularak gelen kullanıcı isteklerine düşük gecikme süresiyle yanıt verilmektedir. Kullanıcılar, araç markası, modeli, kilometresi, üretim yılı ve motor gücü gibi teknik özellikleri uygulama arayüzü aracılığıyla sisteme iletmekte; bu veriler modelin eğitim sürecinde kullanılan ön işleme adımlarına uygun şekilde asenkron olarak işlenerek fiyat tahmini gerçekleştirilmektedir. Ayrıca uygulama, yalnızca tek bir tahmin değeri sunmak yerine, modelin performans metriği olan Ortalama Mutlak Hata (MAE: 122.701 TL) esas alınarak bir tahmin aralığı üretmektedir. Bu yaklaşım, piyasa koşullarındaki belirsizlikleri dikkate alarak kullanıcılara daha gerçekçi ve güvenilir bir değerlendirme imkânı sunmakta, aynı zamanda karar destek sürecinin daha objektif ve analitik bir temele oturmasına katkı sağlamaktadır.



Şekil 9. *Makine öğrenmesi tabanlı fiyat tahmin servisine ait* `/predict` *uç noktasının Swagger UI üzerinden otomatik olarak oluşturulan dokümantasyonu. Uç noktanın aldığı parametreler, beklenen istek gövdesi ve örnek JSON yapısı gösterilmektedir.*

```python
@app.post("/predict")  sezeryigitt
async def predict(data: CarData):
    try:
        # Veriyi dataframe'e çevir ve düzenle
        df = pd.DataFrame([data.dict()])
        df = df[['marka', 'seri', 'kilometre', 'vites_tipi', 'yakit_tipi',
                 'kasa_tipi', 'renk', 'motor_hacmi', 'motor_gucu', 'cekis',
                 'ort_yakit_tuketimi', 'yakit_deposu', 'kimden', 'arac_yasi']]

        # Model tahmini
        prediction_value = model.predict(df)[0]
        pred = int(prediction_value)

        # Minimum fiyat kontrolü
        if pred < 50000:
            pred = 50000

        # MAE bazlı güven aralığı (MAE: 122,701 TL)
        mae = 122701
        min_price = max(50000, pred - mae)  # Minimum 50k olsun
        max_price = pred + mae

        return {
            "price": f"{pred:,}".replace( _old: ",", _new: ".") + " TL",
            "min": f"{min_price:,}".replace( _old: ",", _new: "."),
            "max": f"{max_price:,}".replace( _old: ",", _new: ".")
        }
    except Exception as e:
        return {
            "price": "Hata oluştu",
            "min": "-",
            "max": "-"
        }
```

Şekil 10. *`/predict` uç noktasının sunucu tarafında gerçekleştirdiği işlemleri gösteren kod yapısı. Gelen veriler veri çerçevesine dönüştürülmekte, model tahmini yapılmakta ve hata payı dikkate alınarak fiyat aralığı hesaplanmaktadır.*



Şekil 11. *Kullanıcıların araç özelliklerini girerek fiyat tahmini almasını sağlayan web tabanlı kullanıcı arayüzü. Girilen veriler, arka planda REST API aracılığıyla makine öğrenmesi modeline iletilmektedir.*

Bu bölümde, geliştirilen ikinci el araç fiyat tahmin modelinin gerçek bir uygulama ortamına nasıl entegre edildiği ayrıntılı olarak açıklanmıştır. Makine öğrenmesi modeli, REST tabanlı bir servis aracılığıyla dış dünyaya açılmış; Swagger UI ile dokümantasyonu sağlanmış ve kullanıcı dostu bir web arayüzü ile desteklenmiştir. Böylece model, yalnızca teorik bir çalışma olmaktan çıkarılarak gerçek kullanıcılar tarafından doğrudan kullanılabilir bir karar destek sistemine dönüştürülmüştür. Elde edilen bu yapı, makine öğrenmesi tabanlı fiyat tahmin sistemlerinin pratikte uygulanabilirliğini ve ölçeklenebilirliğini açıkça ortaya koymaktadır.

3. TARTIŞMA ve SONUÇ *(DISCUSSION and CONCLUSION)*

Bu çalışmada, ikinci el araç fiyatlarının tahmin edilmesi problemine veri odaklı ve uçtan uca bir mühendislik yaklaşımıyla çözüm sunulmuştur. Dinamik web kazıma yöntemleri kullanılarak elde edilen güncel ve bölgesel veri seti, literatürde sıklıkla kullanılan statik ve geçmiş tarihli veri setlerine kıyasla daha gerçekçi piyasa koşullarını yansıtmaktadır. Bu durum, geliştirilen modelin pratikte kullanılabilirliğini ve güncel piyasa dinamiklerine uyumunu önemli ölçüde artırmaktadır.

Modelleme aşamasında gerçekleştirilen karşılaştırmalı analizler, doğrusal regresyon temelli yöntemlerin ikinci el araç fiyatlarını etkileyen karmaşık ve doğrusal olmayan ilişkileri yakalamakta yetersiz kaldığını göstermiştir. Ağaç tabanlı ve topluluk öğrenme yöntemleri daha başarılı sonuçlar üretmiş olsa da, bazı modellerde eğitim ve test performansları arasındaki farklar aşırı öğrenme riskine işaret etmiştir. CatBoost algoritması ise kategorik değişkenleri doğrudan işleyebilme yeteneği, ordered boosting mekanizması ve simetrik ağaç yapısı sayesinde dengeli bir öğrenme süreci sergilemiş ve yüksek genelleme kabiliyeti göstermiştir.

Hiperparametre optimizasyonu sonrasında CatBoost modelinin test veri seti üzerindeki R^2 değerinin 0.9297 seviyesine ulaşması, modelin araç fiyatlarındaki değişkenliğin büyük bir bölümünü başarıyla açıkladığını ortaya koymaktadır. Düşük MAE ve RMSE değerleri, modelin tahmin hatalarının sınırlı olduğunu ve karar destek sistemi olarak güvenilir biçimde kullanılabileceğini göstermektedir. Özellikle düşük ve orta segment araçlarda yüksek tahmin doğruluğu elde edilmesi, modelin geniş kullanıcı kitlesi için pratik değer sunduğunu ortaya koymaktadır. Lüks ve nadir araç segmentlerinde görece daha yüksek hata değerleri ise bu araçlara ait veri sayısının sınırlı olmasından kaynaklanmakta olup, gelecekte daha büyük ve dengeli veri setleriyle bu durumun iyileştirilebileceği değerlendirilmektedir.

Çalışmanın önemli katkılarından biri, geliştirilen makine öğrenmesi modelinin FastAPI tabanlı bir web servisi aracılığıyla konuşlandırılmasıdır. Bu sayede model, teorik bir analiz olmaktan çıkarılarak gerçek kullanıcıların doğrudan faydalanabileceği bir web tabanlı fiyat tahminleme uygulamasına dönüştürülmüştür. Tahmin aralığı yaklaşımı kullanılarak kullanıcıya tek bir değer yerine belirsizliği de yansıtan bir çıktı sunulması, karar verme sürecini daha gerçekçi ve analitik hâle getirmektedir.

Sonuç olarak, bu çalışma; güncel veri toplama, gelişmiş makine öğrenmesi algoritmaları ve modern yazılım mimarilerinin birlikte kullanıldığı, uygulanabilir ve ölçeklenebilir bir ikinci el araç fiyat tahmin sistemi ortaya koymuştur. Gelecek çalışmalarda veri setinin farklı şehirleri kapsayacak şekilde genişletilmesi, makroekonomik değişkenlerin modele dahil edilmesi ve açıklanabilir yapay zekâ (XAI) yöntemlerinin daha derinlemesine uygulanmasıyla sistemin doğruluğu ve kullanıcı güveninin daha da artırılabileceği öngörülmektedir.

KAYNAKLAR (REFERENCES)

[1] Kuiper, S. (2008). "Modeling Used Car Prices," *Journal of Statistics Education*, vol. 16, no. 1.

[2] Listiani, M. (2009). "Support Vector Regression Analysis for Price Prediction of Used Cars," *University of Indonesia*.

[3] Gegic, E., et al. (2019). "Car Price Prediction of Used Cars Using Machine Learning Techniques," *TEM Journal*, vol. 8, no. 1, pp. 113-118.

[4] Prokhorenkova, L., et al. (2018). "CatBoost: unbiased boosting with categorical features," *Advances in Neural Information Processing Systems (NeurIPS)*.

[5] Hancock, J. T., & Khoshgoftaar, T. M. (2020). "CatBoost for big data: an interdisciplinary review," *Journal of Big Data*, vol. 7, no. 1.

[6] S. Huber, H. Wiemer, D. Schneider ve S. Ihlenfeldt, "DMME: Data mining methodology for engineering applications–a holistic," 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy. Doi: 10.1016/j.procir.2019.02.106

[7] A. F. Fahmy, H. K. Mohamed ve A.H. Yousef, "A data mining experimentation framework to improve six sigma projects," in 2017 13th International Computer Engineering Conference. Doi: 10.1109/ICENCO.2017.8289795