

# Konut Fiyatlarının Tahmini

## Estimation of Housing Prices

Buse Yener  
Biliřim Sistemleri Mühendislięi  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
busee.yener@gmail.com

**Özetçe**—Bu dökümanda İstanbul’da bulunan satılık konut fiyatlarının *bs4* kütüphanesiyle verisi çekilerek yapay zeka modelinin eğitilmesi anlatılmaktadır.

**Anahtar Kelimeler** —*bs4, veri çekme ,regresyon, makine öğrenmesi, ön işleme, arayüz.*

**Abstract**— This document explains how to train the artificial intelligence model by capturing data of house prices for sale in Istanbul with the BS4 library.

**Keywords** — *bs4, data extraction, regression, machine learning, pre-processing, interface.*

### I. GİRİř

Bu raporda, İstanbul'daki konut fiyatlarını tahmin etmek amacıyla geliştirilen yapay zeka projesinin detayları ele alınmaktadır. Projede, veri toplama aşamasında emlakjet.com sitesi kullanılarak İstanbul'un tüm ilçelerinden yaklaşık 3600 veri çekilmiştir. Toplanan veriler, düzenleme ve etiketleme işlemlerinin ardından çeşitli regresyon modelleri ile analiz edilmiştir. Kullanılan modeller arasında Gradient Boosting Regresyon, Multilinear Linear Regresyon, Random Forest Regresyon, Destek Vektör Regresyonu ve Polynomial Regresyon yer almaktadır. Yapılan değerlendirmeler sonucunda en iyi performansı Gradient Boosting Regresyon modeli göstermiştir. Bu model ile eğitim süreci tamamlanmış ve bir kullanıcı arayüzü tasarlanmıştır. Projenin tüm aşamaları Jupyter Lab ortamında gerçekleştirilmiştir.

### II. VERİ ÇEKME

Veriler <https://www.emlakjet.com/> sitesinden çekilmiştir. İlk olarak, requests, BeautifulSoup, pandas ve re kütüphaneleri projeye dahil edilmiştir. Requests kütüphanesi web sayfalarına HTTP istekleri göndermek için kullanılırken, BeautifulSoup HTML ve XML belgelerini ayrıştırmak için kullanılmıştır. Pandas kütüphanesi veri manipülasyonu ve analizi için

kullanılmış ve re kütüphanesi düzenli ifadelerle çalışmak için kullanılmıştır.

İkinci adımda, İstanbul'daki çeşitli ilçelerdeki satılık konut ilanlarının sayfa URL'leri siteler adlı bir listeye eklenmiştir. Bu liste, her ilçedeki ilanların bulunduğu birkaç sayfanın URL'lerini içermektedir. Örneğin, Kadıköy ilçesindeki ilanlar üç sayfada yer almaktadır ve her sayfanın URL'si listeye eklenmiştir.

Üçüncü adımda, ilan verilerinin saklanacağı boş bir liste olan emlakjetİlan oluşturulmuştur. Bu liste, her bir ilan için çekilen verileri saklayacaktır.

Dördüncü adımda, her URL için bir döngü başlatılarak, sayfa içerięi requests kütüphanesi ile çekilmiş ve BeautifulSoup ile ayrıştırılmıştır. Bu adımda, sayfadaki ilanları içeren belirli bir HTML div elementi bulunmuş ve bu elementten tüm ilanlar çıkarılmıştır.

Beşinci adımda, her bir ilan için detay sayfasına gidilmiş ve ilanla ilgili daha ayrıntılı bilgiler çekilmiştir. Her ilan için detay sayfasının URL'si oluşturulmuş ve bu sayfa tekrar requests ve BeautifulSoup ile çekilip ayrıştırılmıştır.

Altıncı adımda, ilan adının alınması ve bu addan ilçenin belirlenmesi işlemi gerçekleştirilmiştir. İlan adı içeren HTML elemanı bulunmuş ve metin içerięi çekilmiştir. Bu metin, " - " karakterine göre bölünerek ilçenin adı alınmıştır.

Yedinci adımda, ilan fiyatı düzenli ifade kullanılarak çekilmiştir. Fiyat bilgisi içeren HTML elemanı bulunmuş ve metin içerięi düzenli ifade ile işlenerek sadece rakamlar alınmıştır.

Sekizinci adımda, ilanla ilgili diğer detaylar çekilmiştir. Bu detaylar, ilan sayfasındaki belirli HTML div elemanlarından alınmış ve bir sözlük yapısında saklanmıştır. Her bir anahtar-değer çifti, ilan detaylarını içermektedir.

Dokuzuncu adımda, tüm ilan bilgilerini içeren bir pandas DataFrame oluşturulmuştur. Bu DataFrame, her bir ilan için ilçeyi, fiyatı ve diğer detayları içermektedir.

Son adımda, oluşturulan DataFrame bir CSV dosyasına yazılmıştır. Bu işlem, pandas kütüphanesinin to\_csv

fonksiyonu ile gerçekleştirilmiş ve dosya UTF-8 formatında kaydedilmiştir. Bu adımlarla, belirli URL'lerden çekilen ilan verileri düzenli bir formatta saklanmış ve analiz edilebilir hale getirilmiştir.

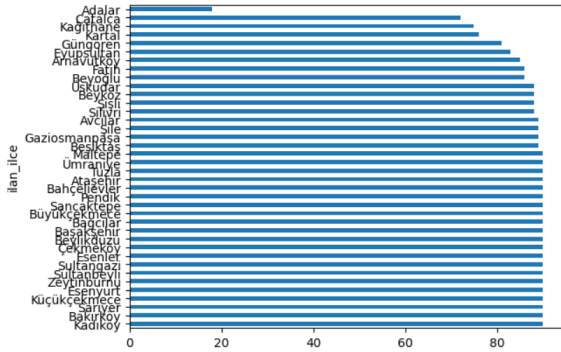
### III. VERİ İNCELEME VE DÜZENLEME

Veri düzenleme aşamasında ilk olarak, pandas, seaborn ve matplotlib kütüphaneleri içe aktarılır. CSV dosyası pd.read\_csv() fonksiyonu ile okunarak bir DataFrame (df) oluşturulur.

```
RangeIndex: 3260 entries, 0 to 3259
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ilan_ilce              3260 non-null   object
1   ilan_fiyat             3260 non-null   int64
2   net_metrekare          3260 non-null   object
3   oda_sayısı            3260 non-null   object
4   bulunduđu_kat         3260 non-null   object
5   ısıtma_tipi           3260 non-null   object
6   krediye_uygunluk      3260 non-null   object
7   banyo_sayısı         3260 non-null   object
8   tipi                  3260 non-null   object
9   brüt_metrekare        3260 non-null   object
10  binanın_yaşı          3260 non-null   object
11  binanın_kat_sayısı    3260 non-null   int64
12  site_içerisinde       3260 non-null   object
dtypes: int64(2), object(11)
memory usage: 331.2+ KB
```

Oluşturulan DataFrame üzerinden değerler düzenlenmiştir.

#### A. ilan\_ilce



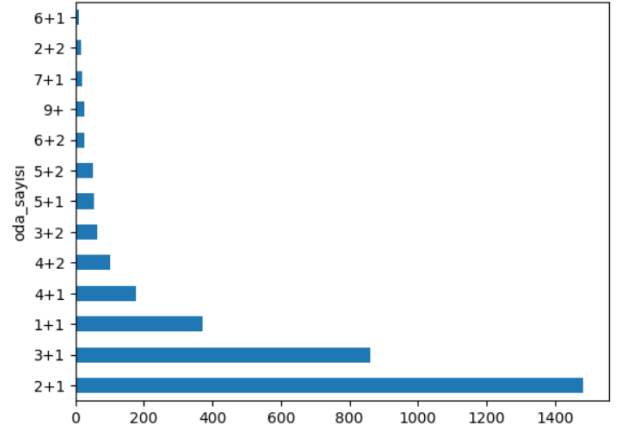
İstanbul'un 39 ilçesinden veri çekilmiştir. Verilerin dağılımı yukarıdaki grafikte mevcuttur.

#### B. net\_metrekare ve brüt\_metrekare

brüt\_metrekare ve net\_metrekare sütunlarındaki M2 metni kaldırılmıştır. Bu işlemler sonucunda, brüt\_metrekare ve net\_metrekare sütunlarında yalnızca sayısal değerler kalır ve birim bilgileri gibi metin parçaları kaldırılmış olur. Bu sayede, bu sütunlardaki değerler sayısal işlemler veya analizler için daha uygun hale gelmiştir.

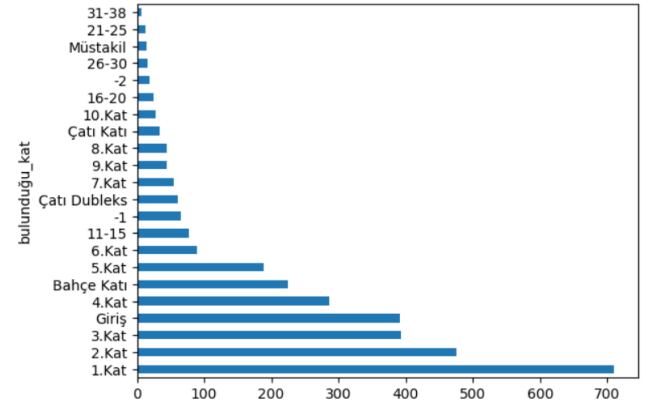
#### C. oda\_sayısı

Verilerin çok dağınık olması sebebiyle az bulunan değerler en yakında bulunan değere atanmıştır. Örneğin, "8+3" değeri "9+" olarak, "1 Oda" değeri ise "1+1" olarak değiştirilir. Son dağılımı aşağıdaki gibidir.

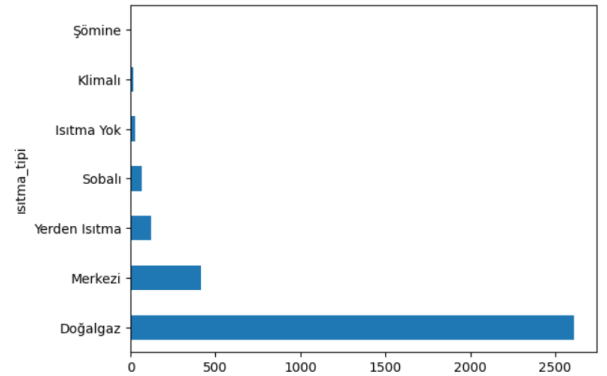


#### D. bulunduđu\_kat

Verilerin dağınık ve az olması sebebiyle az olan değerler belirli bir aralığa alınmıştır. Örneğin "Kot 2 (-2).Kat" değeri "-2" olarak, 16 ile 20 arasındaki kat numaraları ise "16-20" olarak gruplandırılır. Düzenlenmiş haliyle dağılımı aşağıdaki grafikte mevcuttur.



#### E. ısıtma\_tipi



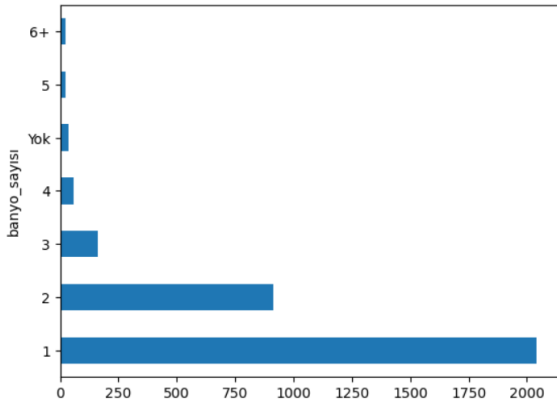
Bu sütundaki çeşitli ısıtma türleri daha genel veya yaygın kategorilere dönüştürülmüştür. Bu, veri setindeki çeşitliliği azaltarak analiz ve görselleştirme süreçlerini kolaylaştırmak amacıyla yapılmıştır. Örneğin, "Kombi Doğalgaz", "Merkezi Doğalgaz", "Kombi Fueloil", "Güneş Enerjisi", "Elektrikli Radyatör", "Fancoil Ünitesi" ve "VRV" gibi farklı ısıtma tipleri "Doğalgaz" kategorisi altında birleştirilmiştir.

#### F. krediye\_uygunluk

DataFrame'deki "krediye\_uygunluk" sütununda "Bilinmiyor" olarak işaretlenmiş olan satırlar "Krediye Uygun Değil" olarak değiştirilmiştir. Yani, eğer bir satırın "krediye\_uygunluk" değeri "Bilinmiyor" ise, o satırın "krediye\_uygunluk" değeri "Krediye Uygun Değil" olarak güncellenir. Bu, veri setindeki belirsizlikleri ele almak ve veri analizi için daha sağlam sonuçlar elde etmek için yapılan bir düzenlemedir.

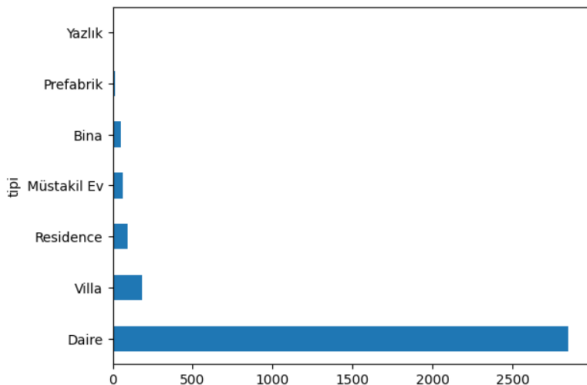
#### G. banyo\_sayisi

Bu sütunda veriler üzerinde düzenleme yapılmamıştır. Dağılımı aşağıdaki grafikte verilmiştir.



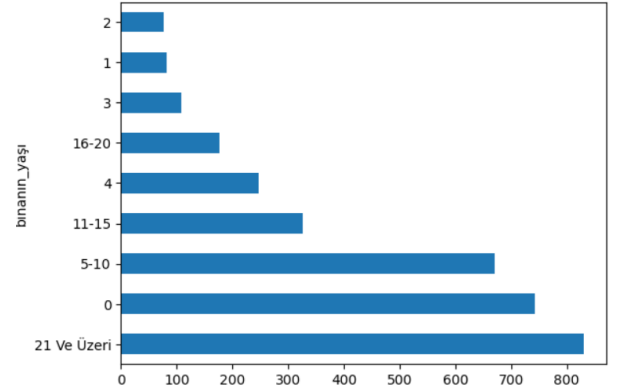
#### H. tipi

Sütunda belirli tiplere sahip olan satırlar "Villa" olarak değiştirilmiştir. Özellikle "Dağ Evi", "Köy Evi", "Yalı Dairesi", "Yalı", "Çiftlik Evi" ve "Köşk" gibi az olan veya daha spesifik tipler, genel bir kategori altında toplanarak "Villa" olarak güncellenir. Bu, veri setindeki çeşitli ev tiplerini daha genel bir kategori altında birleştirmek ve analiz için daha anlamlı bir formata getirmek için kullanılır.



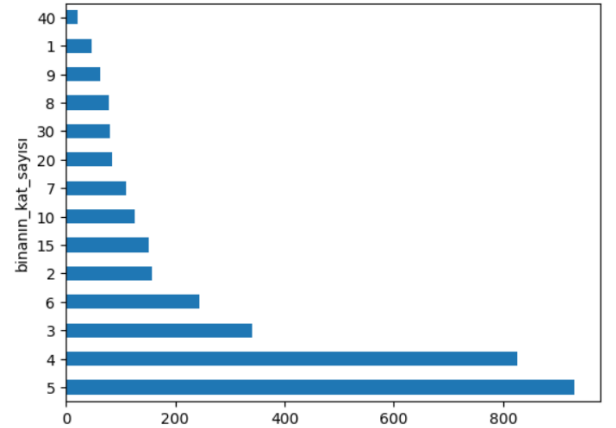
#### I. binanın\_yaşı

Bu sütunda çok fazla değişiklik olmamıştır. Sadece yazım hataları düzenlenmiştir. Son dağılımı aşağıdaki grafikte verilmiştir.



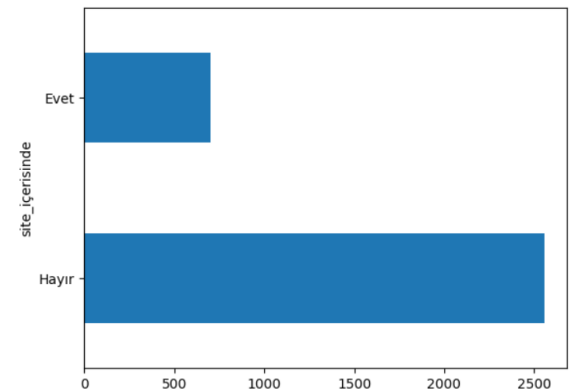
#### J. binanın\_kat\_sayisi

Verilerin çok dağınık olması sebebiyle az bulunan değerler en yakında bulunan aralığa atanmıştır. Örneğin, 65, 32, 44, 34, 52, 35, 50, 38, 40, 45 ve 33 gibi değerler 40 olarak düzeltilmiştir.



#### K. site\_içerisinde

Bu sütunda veriler üzerinde düzenleme yapılmamıştır. Dağılımı aşağıdaki grafikte verilmiştir.



Bu işlemler, ilan verilerinin daha tutarlı ve analiz edilebilir bir formatta olmasını sağlar. Bu düzenlemeler yapıldıktan sonra, temizlenmiş ve düzenlenmiş veri seti "emlakjet.csv" dosyasına kaydedilir.

#### IV. ÖN İŞLEME

Bu bölümde, veri setindeki kategorik özelliklerin Label Encoding işlemiyle sayısal değerlere dönüştürülmesi gerçekleştirilmiştir. Öncelikle, *pandas* kütüphanesi aracılığıyla veri seti *emlakjet.csv* dosyasından okunmuştur. Daha sonra, *sklearn.preprocessing* altındaki *LabelEncoder* sınıfı kullanılarak her bir kategorik özelliğe ayrı ayrı bir Label Encoding işlemi uygulanmıştır.

Özellikle, *ilan\_ilce*, *oda\_sayısı*, *bulunduğu\_kat*, *ısıtma\_tipi*, *krediye\_uygunluk*, *banyo\_sayısı*, *tipi*, *binanın\_yaşı*, *binanın\_kat\_sayısı* ve *site\_içerisinde* sütunları Label Encoding işlemine tabi tutulmuştur.

Label Encoding işlemi, kategorik değerleri alfabetik sıraya göre sıralayarak her bir farklı değere bir tamsayı atar. Bu şekilde, makine öğrenimi modelleri gibi sayısal girdi gerektiren algoritmalar için kategorik verilerin işlenmesi sağlanır.

Son olarak, Label Encoding işlemi uygulanmış veri seti *emlakjetisleme.csv* dosyasına kaydedilir ve tekrar okunarak *df* değişkenine atanır. Bu işlem veri setinin sayısal olarak kodlanmış sürümünü oluşturur ve makine öğrenimi modellerinin eğitiminde kullanılabilir hale getirilir.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3260 entries, 0 to 3259
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   ilan_ilce              3260 non-null   int64
1   ilan_fiyat             3260 non-null   int64
2   net_metrekare          3260 non-null   int32
3   oda_sayısı            3260 non-null   int64
4   bulunduğ_kat          3260 non-null   int64
5   ısıtma_tipi            3260 non-null   int64
6   krediye_uygunluk       3260 non-null   int64
7   banyo_sayısı          3260 non-null   int64
8   tipi                   3260 non-null   int64
9   brüt_metrekare         3260 non-null   int32
10  binanın_yaşı           3260 non-null   int64
11  binanın_kat_sayısı     3260 non-null   int64
12  site_içerisinde        3260 non-null   int64
dtypes: int32(2), int64(11)
memory usage: 305.8 KB
```

#### V. MAKİNE ÖĞRENMESİ

Konut fiyatlarını tahmin etmek için çeşitli regresyon modelleri kullanılmıştır. Bu modellerin her biri farklı algoritmalar ve tekniklerle çalışmakta olup, her birinin avantajları ve dezavantajları bulunmaktadır. Kullanılan regresyon modelleri ve ilgili kütüphaneler aşağıda detaylandırılmıştır:

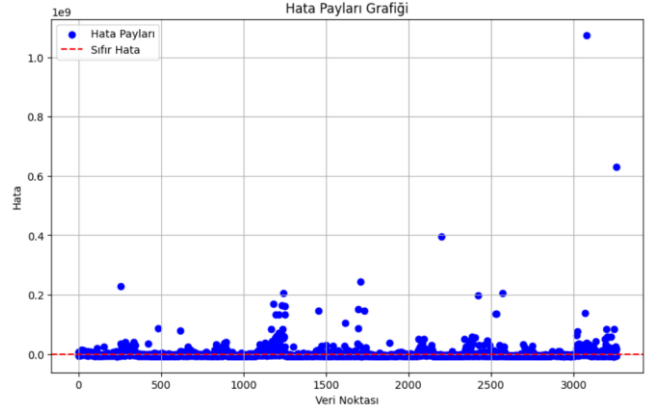
##### A. Çoklu Doğrusal Regresyon (Multilinear Linear Regression)

İlk olarak, "emlakjetisleme.csv" dosyasından veriler okunur ve bir doğrusal regresyon modeli oluşturulur. Çoklu Doğrusal Regresyon, bağımsız değişkenlerin bağımlı değişkenle doğrusal ilişkisini modelleyen basit ve etkili bir yöntemdir. Bu model, verilerin doğrusal ilişkilerle iyi temsil edildiği durumlarda hızlı ve güvenilir sonuçlar sağlar. Ancak, doğrusal olmayan ilişkilerde performansı düşebilir. Burada model, gayrimenkul ilanlarının fiyatlarını tahmin etmek için kullanılmıştır.

Daha sonra, modelin tahmin yeteneğini değerlendirmek için çeşitli metrikler kullanılır. Bu metrikler arasında modelin genel performansını ölçen R-kare (R-squared) skoru, ortalama mutlak hata (MAE), ortalama kare hata (MSE), ve ortalama mutlak yüzde hata (MAPE) bulunur.

- Ortalama Mutlak Hata (MAE): 3900944.0649752165
- Ortalama Kare Hatası (MSE): 98503053553011.28
- Ortalama Mutlak Yüzde Hata (MAPE): 64.34678957029942
- R-kare (R-squared): 0.871686182816037

Modelin gerçek değerlerle tahmin edilen değerler arasındaki farkları incelemek için bir hata grafiği çizilmiştir.



Bu grafik, her bir veri noktası için tahmin edilen ve gerçek fiyatlar arasındaki farkları gösterir.

##### B. Gradient Artırma Regresyonu (Gradient Boosting Regressor)

Gradient Artırma Regresyonu, zayıf tahmin edicilerin bir araya getirilerek güçlü bir tahmin edici oluşturulduğu bir ensemble öğrenme yöntemidir. Bu model, her bir tahmin edicinin hatalarını azaltarak performansını artırır. Gradient Artırma özellikle karmaşık veri setlerinde yüksek doğruluk sağlar ve genellikle en iyi performans gösteren modeller arasında yer alır.

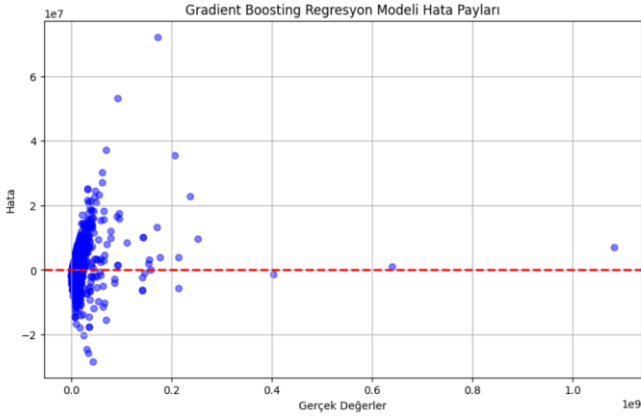
Projede GradientBoostingRegressor sınıfından bir regresyon modeli oluşturulur. Bu model, veri setindeki bağımsız değişkenler (ilan özellikleri) ile bağımlı değişken (ilan fiyatı)

arasındaki ilişkiyi öğrenir. fit yöntemi kullanılarak model, veri setindeki ilgili özellikler ve fiyatlar arasındaki ilişkiyi öğrenir.

Modelin performansı, ortalama mutlak hata (MAE), ortalama kare hata (MSE), ortalama mutlak yüzde hata (MAPE) ve R-kare (R-squared) skoru gibi çeşitli metrikler kullanılarak değerlendirilir. Bu metrikler, modelin ne kadar iyi çalıştığını ve tahminlerinin gerçek değerlere ne kadar yakın olduğunu belirlemek için kullanılır.

- Ortalama Mutlak Hata (MAE): 2116035.294539453
- Ortalama Kare Hatası (MSE): 16983462115432.328
- Ortalama Mutlak Yüzde Hata (MAPE): 36.643442580237156
- R-kare (R-squared): 0.9778766974786466

Modelin tahmin yeteneğini daha iyi anlamak için hata paylarının grafiği çizilir. Bu grafik, her bir gerçek fiyatın, modelin tahmin ettiği fiyattan ne kadar sapma gösterdiğini görselleştirir.



### C. Rastgele Orman Regresyonu (Random Forest Regressor)

Rastgele Orman Regresyonu, birden çok karar ağacının birlikte kullanıldığı bir ensemble öğrenme yöntemidir. Bu model, overfitting'i azaltarak yüksek doğruluk sağlar ve değişkenler arasındaki karmaşık ilişkileri öğrenebilir. Rastgele Orman, büyük ve çeşitli veri setlerinde oldukça etkili sonuçlar verir.

RandomForestRegressor sınıfından bir regresyon modeli oluşturulur. Bu model, veri setindeki bağımsız değişkenler (ilan özellikleri) ile bağımlı değişken (ilan fiyatı) arasındaki ilişkiyi öğrenir. Ardından, fit yöntemi kullanılarak model, veri setindeki ilgili özellikler ve fiyatlar arasındaki ilişkiyi öğrenir.

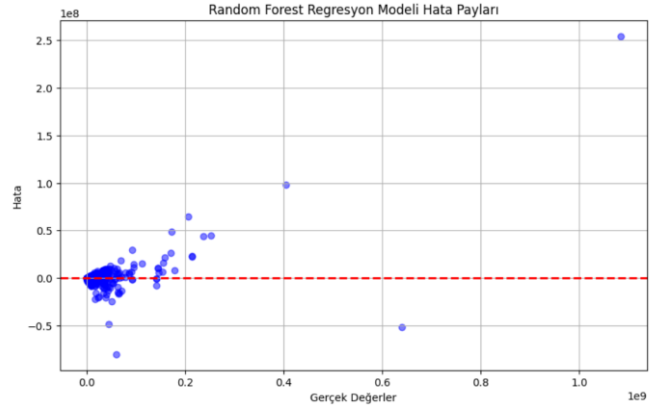
Model, eğitim veri seti üzerinde eğitildikten sonra, aynı veri seti üzerinde veya farklı bir veri seti üzerinde fiyat tahminleri yapabilir. predict yöntemi kullanılarak, belirtilen özelliklere dayanarak ilan fiyatlarını tahmin eder.

Modelin performansı, ortalama mutlak hata (MAE), ortalama kare hata (MSE), ortalama mutlak yüzde hata (MAPE) ve R-kare (R-squared) skoru gibi çeşitli metrikler kullanılarak değerlendirilir. Bu metrikler, modelin ne kadar iyi çalıştığını

ve tahminlerinin gerçek değerlere ne kadar yakın olduğunu belirlemek için kullanılır.

- Ortalama Mutlak Hata (MAE): 1044656.379194128
- Ortalama Kare Hatası (MSE): 23766914301078.82
- Ortalama Mutlak Yüzde Hata (MAPE): 12.802589391460979
- R-kare (R-squared): 0.9690403151307961

Gerçek fiyatlar ile tahmin edilen fiyatlar arasındaki hata payları görselleştirilir.



Bu grafik, modelin belirli fiyatları ne kadar doğru tahmin ettiğini veya ne kadar hata yaptığını gösterir.

### D. Destek Vektör Regresyonu (Support Vector Regression - SVR)

SVR, verileri yüksek boyutlu bir uzaya haritalayarak, bu uzayda en iyi doğrusal tahmin çizgisini bulmaya çalışan bir regresyon tekniğidir. SVR, özellikle karmaşık ve doğrusal olmayan veri ilişkilerini modellemeye başarılıdır. Doğru kernel seçimiyle yüksek performans elde edilebilir.

Destek Vektör Regresyonu (SVR) kullanılarak ilan fiyatlarını tahmin etmek amacıyla bir model oluşturulmuş ve değerlendirilmiştir. İlk olarak, gerekli kütüphaneler (SVR, StandardScaler, ve make\_pipeline) sklearn kütüphanesinden ithal edilmiştir. Bu araçlar, SVR modelini oluşturmak ve veri ölçeklendirmeyi kolaylaştırmak için kullanılmıştır.

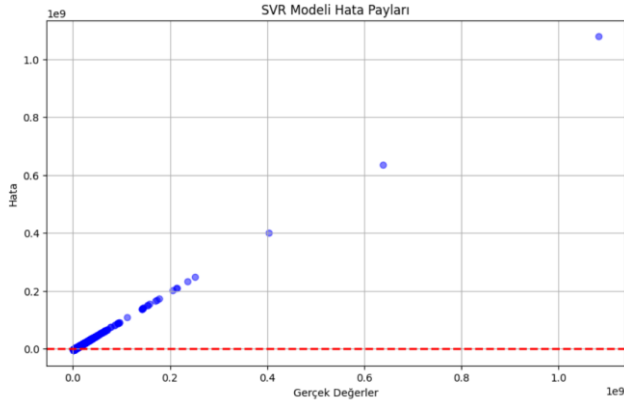
SVR modeli, make\_pipeline fonksiyonu kullanılarak oluşturulmuştur. Bu fonksiyon, veriyi önce ölçeklendiren (StandardScaler) ve ardından SVR modelini uygulayan bir boru hattı (pipeline) oluşturur. Bu adım, modelin eğitilmeden önce verinin uygun şekilde ölçeklendirilmesini sağlar. Ardından, SVR modeli belirlenen özelliklere (bağımsız değişkenler) ve ilan fiyatlarına (bağımlı değişken) göre eğitilmiştir. Bu adım, modelin eğitim veri setine dayalı olarak fiyat tahmin etmeyi öğrenmesini sağlar.

Eğitilen SVR modeli kullanılarak ilan fiyatları tahmin edilmiştir. Tahminlerin doğruluğunu değerlendirmek için çeşitli performans metrikleri kullanılmıştır. Ortalama Mutlak

Hata (MAE), gerçek değerler ile tahmin edilen değerler arasındaki ortalama mutlak farkı hesaplar. Ortalama Kare Hatası (MSE), gerçek değerler ile tahmin edilen değerler arasındaki kare farkların ortalamasını hesaplar. Ortalama Mutlak Yüzde Hata (MAPE), gerçek ve tahmin edilen değerler arasındaki mutlak yüzde farkın ortalamasını hesaplar. R-kare (R-squared) ise modelin veri üzerindeki doğruluk oranını belirler; R-kare değeri 1'e ne kadar yakınsa, model o kadar iyi demektir.

- Ortalama Mutlak Hata (MAE):  
5707505.996486804
- Ortalama Kare Hatası (MSE):  
788456069793323.2
- Ortalama Mutlak Yüzde Hata (MAPE):  
52.80295267980249
- R-kare (R-squared):  
-0.02707280990634464

Ek olarak, hata paylarını görselleştiren bir grafik eklemek amacıyla, gerçek fiyatlar ile tahmin edilen fiyatlar arasındaki hata payları hesaplanmış ve bu hata payları bir grafik üzerinde gösterilmiştir.



#### E. Polinomsal Regresyon (Polynomial Regression)

Polinomsal Regresyon, bağımsız değişkenlerin polinom terimlerle genişletilmesi ve bu genişletilmiş terimlerle doğrusal regresyon yapılmasıdır. Bu model, doğrusal olmayan veri ilişkilerini modellemede etkilidir ancak yüksek dereceli polinomlar overfitting'e neden olabilir. Uygun derece seçimiyle dengeli ve etkili sonuçlar alınabilir.

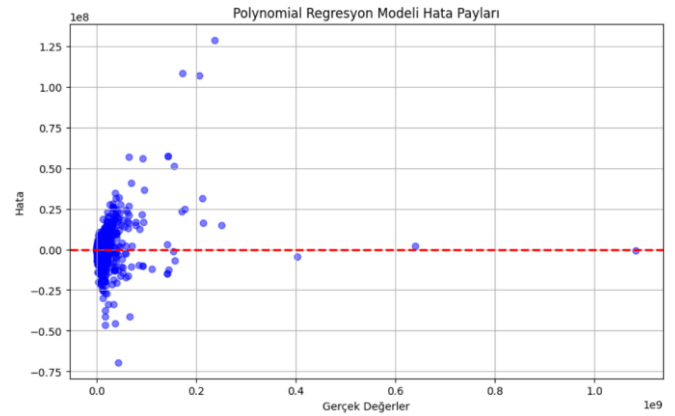
Polynomial Regresyon kullanarak ilan fiyatlarını tahmin etmek için bir model oluşturulur, modelin performansı değerlendirilir ve sonuçları grafikleştirilir. İlk olarak, gerekli kütüphaneler yüklenir. Polynomial Regresyon modelini oluşturmak için make\_pipeline fonksiyonu kullanılarak PolynomialFeatures ve LinearRegression bileşenleri birleştirilir. Model, bağımsız değişkenler ve bağımlı değişken kullanılarak eğitilir.

Model eğitildikten sonra, veri kümesi üzerindeki ilan fiyatları tahmin edilir. Tahmin edilen fiyatlar kullanılarak model performansı değerlendirilir. Bu değerlendirme, Ortalama

Mutlak Hata (MAE), Ortalama Kare Hatası (MSE), Ortalama Mutlak Yüzde Hata (MAPE) ve R-kare (R-squared) gibi performans metrikleri ile yapılır. Bu metrikler modelin doğruluğunu ve hata oranını ölçer.

- Ortalama Mutlak Hata (MAE):  
3243566.441691001
- Ortalama Kare Hatası (MSE):  
47273266593265.06
- Ortalama Mutlak Yüzde Hata (MAPE):  
57.27213596960347
- R-kare (R-squared):  
0.9384200482264996

Son olarak, modelin hata paylarını görselleştirmek için bir grafik çizildi. Grafik, gerçek ilan fiyatları ile modelin tahmin ettiği fiyatlar arasındaki farkları gösterir. Bu farklar bir scatter plot üzerinde görselleştirilir ve yatay kırmızı çizgi sıfır hatayı temsil eder. Bu grafik, modelin doğruluğunu ve hata dağılımını görsel olarak değerlendirmemizi sağlar.



Emlak ilan fiyatlarını tahmin etmek için kullanılan beş farklı regresyon modeli (Linear Regression, Gradient Boosting Regression, Random Forest Regression, Support Vector Regression ve Polynomial Regression) performansları açısından değerlendirildiğinde, Gradient Boosting Regression'un diğer modellerden daha mantıklı ve başarılı bir sonuç verdiği gözlemlenmiştir.

## VI. ARAYÜZ

Gradient Boosting Regression modeliyle kullanıcıların konut fiyatı tahmini yapabilmeleri için kullanıcı dostu bir arayüz tasarlanmıştır. Bu arayüz, Python'un tkinter kütüphanesi kullanılarak oluşturulmuştur.

Projenin başlangıcında, emlak ilan verilerini içeren bir CSV dosyası pandas kütüphanesi kullanılarak okunur. Veri seti, bağımlı (y) ve bağımsız (X) değişkenlere ayrılır. Bağımlı değişken ilan fiyatı (ilan\_fiyat), bağımsız değişkenler ise konutun çeşitli özellikleridir.



Veri seti, modeli eğitmek ve test etmek için ikiye ayrılır. Bu işlem, train\_test\_split fonksiyonu kullanılarak yapılır ve veri setinin %25'i test seti olarak ayrılır. Bu oran, modelin genelleştirme yeteneğini değerlendirmek için yaygın olarak kullanılan bir orandır.

Gradient Boosting Regression modeli, Gradient Boosting Machines (GBM) algoritmasının bir uygulamasıdır ve karar ağaçlarını kullanarak tahminler yapar. Modelin hiperparametrelerini optimize etmek için GridSearchCV kullanılır. Hiperparametre aralıkları param\_grid sözlüğü ile belirlenmiştir. GridSearchCV, belirli bir hiperparametre ızgarası üzerinde çapraz doğrulama yaparak en iyi parametre kombinasyonunu seçer.

En iyi parametreler ve model, grid\_search nesnesinden elde edilir. Modelin test seti üzerindeki performansı değerlendirilir ve  $R^2$  skoru ile ölçülür.

En iyi sonucu veren Gradient Boosting Regression modeli eğitildikten sonra, kullanıcı dostu bir arayüz tasarlanmıştır. Bu arayüz, kullanıcıdan çeşitli konut özelliklerini (ilçe, konut türü, net metrekare, oda sayısı vb.) girmesini sağlar. Arayüz, Tkinter kütüphanesi kullanılarak oluşturulmuştur ve kullanıcı deneyimini artırmak amacıyla çeşitli etkileşimli elemanlar içerir.

Kullanıcı arayüzünde yapılan işlemler ve açıklamaları aşağıda verilmiştir:

- İlçe Seçimi: Kullanıcı, Combobox aracılığıyla ilçe seçimi yapar. Seçilen ilçe, uygun bir kod ile ilişkilendirilir.
- Konut Türü Seçimi: Konut türü için bir Combobox kullanılır ve kullanıcı tarafından seçilen tür ilgili kodla eşleştirilir.

- Net Metrekare Girişi: Kullanıcı, konutun net metrekaresini Entry aracılığıyla girer ve bu değer kontrol edilir.
- Oda Sayısı Seçimi: Oda sayısı Combobox ile seçilir ve uygun kod ile ilişkilendirilir.
- Bina Yaşı Seçimi: Bina yaşı için bir Combobox kullanılır ve seçilen yaş aralığı uygun kod ile eşleştirilir.
- Site İçerisinde Olup Olmama Seçimi: Combobox ile site içerisinde olup olmadığı seçilir.
- Brüt Metrekare Girişi: Kullanıcı, konutun brüt metrekaresini Entry aracılığıyla girer.
- Isıtma Tipi Seçimi: Isıtma tipi Combobox ile seçilir ve uygun kod ile eşleştirilir.
- Banyo Sayısı Seçimi: Banyo sayısı Combobox ile seçilir ve uygun kod ile eşleştirilir.
- Dairenin Katı Seçimi: Dairenin bulunduğu kat Combobox ile seçilir ve uygun kod ile eşleştirilir.
- Binanın Kat Sayısı Seçimi: Binanın toplam kat sayısı Combobox ile seçilir.
- Kredi Durumu Seçimi: Konutun krediye uygun olup olmadığı Combobox ile seçilir ve uygun kod ile eşleştirilir.

Her bir giriş elemanı, kullanıcıdan veri alır ve bu veriyi ilgili değişkene atar. Bu işlem, ilgili değişkeni düzenleyen fonksiyonlar ile yapılır. Örneğin, ilçe seçimi için ilce\_düzenle fonksiyonu kullanılır. Benzer şekilde, diğer özellikler için de düzenleme fonksiyonları ve arayüz elemanları oluşturulmuştur.

Arayüz, kullanıcıların ilçe, konut türü, net metrekare, oda sayısı gibi çeşitli konut özelliklerini seçmelerine olanak tanır. Kullanıcılar girdileri tamamladıktan sonra, tahmin etme işlemini başlatmak için bir butona tıklarlar.

Kullanıcı girdileri, modelin tahmin fonksiyonuna iletilir ve tahmin edilen fiyat kullanıcıya gösterilir. Kullanıcı gerekli bilgileri eksiksiz girdiyse, tahmini fiyat bir mesaj kutusu ile gösterilir. Aksi takdirde, kullanıcı eksik bilgi girdiğinde bir uyarı mesajı alır.

Yapılan bir tahminin çıktısı aşağıdaki görselde verilmiştir.

Proje drive linki:

<https://drive.google.com/drive/folders/15LizEL4cVB-nNxmZyk3I3u4yqnwcR7q2?usp=sharing>

## KAYNAKLAR

- [1] <https://web.cs.hacettepe.edu.tr/~pinar/courses/VBM683/lectures/regression.pdf>
- [2] <https://learn.microsoft.com/tr-tr/training/modules/understand-regression-machine-learning/>
- [3] <https://pypi.org/project/beautifulsoup4/>
- [4] <https://pandas.pydata.org/>
- [5] <https://jupyter.org/>
- [6] [https://scikit-learn.org/stable/auto\\_examples/ensemble/plot\\_gradient\\_boosting\\_regression.html](https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html)
- [7] [https://scikit-learn.org/stable/modules/linear\\_model.html#polynomial-regression-extending-linear-models-with-basis-functions](https://scikit-learn.org/stable/modules/linear_model.html#polynomial-regression-extending-linear-models-with-basis-functions)
- [8] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>
- [9] <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>
- [10] <https://eneshazr.medium.com/python-ile-gui-geli%C5%9Ftirme-%C3%B6rnekleme-tkinter-51ca1b82166b>
- [11] <https://aws.amazon.com/tr/what-is/data-labeling/>