VERİ ANALİZİ UYGULAMASI

Hazırlayan : **BERRİN GÖÇER**
Öğrenci No : **090150502**

Teslim Tarihi : **05.06.2022**

Ders : **MAT 492**
Danışman : **ÖĞR. GÖR. EVREN TANRIÖVER**

İçindekiler Tablosu

Şekil Tablosu

1. Giriş

Makine öğrenmesi yazılım programlarının programlama durumu olmaksızın sonuçların tahmin edilmesinde daha doğru olmasını sağlayan algoritmalar bütünü olarak açıklanan ve algoritma oluşumları sonrasındaki her güncellemede bilgilerin giriş yapılmasına gerek olmadan analiz kullanımıdır. Giriş verisiyle algoritmalar oluşturulmaktadır. Sonrasında yeni verilerin ortaya çıkmasıyla güncelleme işlemleri otomatik yapılırken çıktı tahmin durumlarında ise istatistiksel analizlerden yararlanmaktadır.

Bu proje kapsamında data kaynağı olarak: *https://www.hepsiburada.com* web sitesi kullanılmaktadır. Proje kapsamında buzdolabı fiyatlarını etkileyen önemli kriterlerin neler olduğu incelenir ve bu özellikler kullanılarak en doğru tahmini yapabilmek adına modelleme yapılarak buzdolabı fiyat tahmini yapılması amaçlanır.

Proje kapsamında Python yazılım dili kullanılmaktadır. Dataların web sitesinden çekilmesi adına Python *BeatifulSoup* kütüphanesi kullanılarak datalar otomatize olarak web sitesinden çekilir ve ardından çekilen bu datalar proje ortamı olarak kullanılan Jupyter Notebook'a aktarılır. Sonrasında çekilen datalar Python *Numpy-Pandas* kütüphaneleri kullanılarak dataframe haline getirilir ve üzerinde analizler gerçekleştirilir.

Elde edilen analizleri güçlendirmek ve doğru bir şekilde incelemek adına *seaborn* ve *matplotlib* kütüphaneleri ile bu analizler görsel grafik haline getirilir.

Datalar üzerindeki işlemler tamamlandıktan sonra elde edilen data incelendiğinde amacımız olan buzdolabı fiyatını etkileyen faktörlerin aslında fiyata doğrusal bir şekilde etki ettiği gözlemlenir ve ardından datalar Train-test olmak üzere 2 parçaya ayrılır. Sonrasında test datası üzerinde alınan tahminlerin doğruluğunun daha iyi analiz edilmesi adına *cross validation* yapılarak test datası da kendi içinde parçalara ayrılır. Son aşamada eğitim için ayrılan data *LinearRegression, Ridge* ve *Lasso makine* modelleri ile eğitilerek ve bu modellerde test datası kullanılarak tahminler yapılır ve modelin doğruluk değeri ölçülür.

2. Web Sitesinden Datanın Elde Edilmesi

Bu proje kapsamında Hepsi Burada web sitesinde bulunan 358 adet buzdolabı 16 sayfalık web sayfasında yer almaktadır. (Web sitesinde birçok sayfa ve her sayfada da birçok ürün bulunmaktadır.)

Sayfa içerisindeki datalar içinde birçok ürün bulunmaktadır ve ürün bazında özgün linkler bulunmaktadır. Bu linklerin yerleri belli modele göre oluşturulmaktadır ve bir düzeni vardır. Ürün özelinde dataların çekilmesi adına yapılan işlemler ilk olarak tek bir ürün özelinde uygulanır ve elde edilen bu kurallar(algoritma) döngü içine sokularak bir fonksiyon haline getirilerek çoklu işleme dönüştürülür. Bu kapsamda ilk olarak gitmek istenilen web sayfasına *request* metodunun içerisine web sayfasının linkini yerleştirerek sayfaya istek atılır. Ardından *get metodu* yardımı ile data çekme isteği bildirilir. Bu şekilde sayfa özelinde bütün datalar *HTML* dilinde *parse* olarak elde edilir. Sayfa üzerindeki tek bir ürünün linkine erişim adına *FindAll methodu* ile elde edilen bu toplu data içerisinden tüm ürünlerin linklerine erişilir. Bu aşamada bir ürünün içerisine girilen noktada bir fonksiyon oluşturulur ve sayfalar üzerinden geçiş sağlamak adına linklerin sonundaki sayfa numaralarını kullanarak sayfalar içinde geçişler ve sayfalardaki ürünlere erişim sağlanır. Son kısım olarak yine tek bir ürün bazındaki özelliklerin yerleri tespit edilir, *html* dilinde bu konumlar belirli başlıklar altında belirli kurallara uygun olarak yerleştirilmişlerdir. Bu dizin yerlerini göstererek çekilen bu fonksiyona *product* ismini verilir(Fonksiyon ismi opsiyoneldir, değiştirilebilir). Bu yerleri öğrenmek adına web sitesinde erişilmek istenen özellik(yazı) üzerinde fare ile sağ tık yapılarak *özellikler* yazısına tıklanır ve *HTML dilinde* bu özelliklerin konumlarının nerede olduğu gözlemlenir ardından *html.find* metodu içerisine bu konumlar bildirilerek bu datalarda çekilir. Sonrasında bu işlem tekrar bir fonksiyon içerisine yerleştirilir. Artık tek seferde toplu olarak ürün üzerinde datalar çekilmek üzere hazır bir komut oluşturulur.

```
In [2]: def getAndParseURL(url):
            result = requests.get(url,headers={"User-Agent":"Mozilla/5.0"})
            soup = bts(result.text, 'html.parser')
            return soup

In [3]: TOTAL_LINK = []
        for totalLink in range(1,17):
            TOTAL_LINK.append("https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:256

In [5]: ALL_PRODUCT_URL = []
        for products in TOTAL_LINK[::]:
            html= getAndParseURL(products)
            for link in html.findAll("li",{"class":"productListContent-item"}):
                ALL_PRODUCT_URL.append("https://www.hepsiburada.com"+link.a["href"])

                print(TOTAL_LINK)
```

```
['https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=1',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=2',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=3',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=4',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=5',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=6',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=7',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=8',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=9',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=10',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=11',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=12',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=13',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=14',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=15',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=16']
['https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=1',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=2',
 'https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=3',
```

Şekil 1: Siteden Çekilen HTML Tipi Data

```
In [6]: result = []
        for details in ALL_PRODUCT_URL[::]:
            html = getAndParseURL(details)

            Urun_Adi = html.find("h1",{"itemprop":"name"}).text.strip()

            try:
                Uretici = html.find("div",{"id":"productTechSpecContainer"}).find(text=re.compile("Üretici")).findNext().text.strip()
            except:
                Uretici = np.nan

            Fiyat = html.find("span",{"data-bind":"markupText:'currentPriceBeforePoint'"}).text.strip()

            try:
                Degerlendirme = html.find("a",{"id":"productReviewsTab"}).text.strip().split("(")[1].split(")")[0]
            except:
                Degerlendirme = np.nan

            try:
                Hacim = html.find("div", class_ = "key-properties").find(text=" Hacim (L) ").findNext().text.strip()
            except:
                Hacim = np.nan

            try:
                Enerji_Sinifi = html.find("div",{"id":"productTechSpecContainer"}).find(text="AB Yeni Enerji Sınıfı").findNext().text.str
            except:
                Enerji_Sinifi = np.nan
            try:
                Yillik_Enerji_Tuketimi = html.find("div",{"id":"productTechSpecContainer"}).find(text="Yıllık Enerji Tüketimi (kWh)").fir
            except:
                Yillik_Enerji_Tuketimi = np.nan

            try:
                Renk = html.find("div",{"id":"productTechSpecContainer"}).find(text=re.compile("Renk")).findNext().text.strip()
            except:
                Renk = np.nan

            Fiyat = html.find("span",{"data-bind":"markupText:'currentPriceBeforePoint'"}).text.strip()

            result.append([Urun_Adi,Uretici,Fiyat,Degerlendirme,Hacim,Enerji_Sinifi, Yillik_Enerji_Tuketimi,Renk])

            kolon_name = ['Urun_Adi','Uretici','Degerlendirme','Hacim','Enerji_Sinifi', 'Yillik_Enerji_Tuketimi', 'Renk','Fiyat']
```

Şekil 2: Çekilen Data İçeriği

3. Veri Toplanması ve Manipülasyon

3.1 Genel Data Bilgisi

İlk aşamada web sitesinden çekilen datalar üzerinde analizler gerçekleştirebilmek adına data *pd.DataFrame metodu* ile data *DataFrame* haline getirilir.(Çekilen datalar *string tipi* formundadırlar).

```
In [9]: df = pd.DataFrame.from_records(result,columns=kolon_name)
        df
```

Out[9]:

	Urun_Adi	Uretici	Degerlendirme	Hacim	Enerji_Sinifi	Yillik_Enerji_Tuketimi	Renk	Fiyat
0	Vestel NF52101 No-Frost Buzdolabı	Vestel	6.998	28	NaN	F	312	Beyaz
1	Profilo BD2155WFNN 453 L No-Frost Çift Kapılı ...	Profilo	7.789	105	453	F	336	Beyaz
2	Samsung RT46K6000WW/TR No-Frost Buzdolabı	Samsung	8.895	936	468	F	323	Beyaz
3	Vestel NF45001 No-Frost Buzdolabı	Vestel	6.799	208	403	F	310	Beyaz
4	Samsung RB50RS334SA/TR No-Frost Buzdolabı	Samsung	13.199	286	520	F	340	Gri
...	...	...	...	...	...	...	...	...
364	Beko 983650 Eı 630 Lt D Enerji Leke Tutmayan I...	Beko	20.150	0	NaN	D	250	Inox
365	Hoover 3'lü Antrasit Çeyiz Seti (Hoce 4T618EX ...	Hoover	20.189	0	NaN	E	256	Inox
366	Teka RFD 77820 GBK Solo Gardrop Tipi NoFrost B...	Teka	50.300	0	500	E	333	Siyah
367	Teka RBF 78720 SS Solo Kombi NoFrost Buzdolabı...	Teka	14.999	0	510	E	322	Açık Gri
368	Liebherr TSL1414 127 Lt A+ Tezgah Altı Buzdola...	NaN	7.250	1	NaN	A	171	Gümüş

369 rows × 8 columns

Şekil 3: Elde Edilen DataFrame

Şekil 3 den de gördüğümüz gibi ilk olarak çektiğimiz data buzdolaplarına dair:

Ürün Adı- Üretici Firma- Web sitedeki Değerlendirme Sayısı- Enerji Sınıfı- Yıllık Enerji Tüketimi- Renk- Fiyat özellikleri çekilmişlerdir. Data 8 kolon ve satır (ürün sayısını temsil eder)'dan oluşmaktadır. Ardından *Fiyat özelliği* içerisindeki tüm datalar üzerinde işlem yapmak adına *apply metodu* ile ondalık cinsindeki datadan nokta kaldırılarak *tamsayı* tipine çevrilir. Bu data özelinde işlem sonrasında *Yıllık Enerji Tüketimi* ve *Değerlendirme* özellikleri üzerinde de apply metodu ile aynı işlem bu kolonlara da uygulanır. Diğer bir özelliğimiz olan *Hacim* kolonu da ondalık *tipine* çevrilir.

```
In [10]: df['Fiyat'] = df['Fiyat'].apply(lambda x : str(x).replace(".","")).astype(int)
         df['Yillik_Enerji_Tuketimi'] = df['Yillik_Enerji_Tuketimi'].apply(lambda x : str(x).replace(".",",")).astype(int)
         df['Degerlendirme'] = df['Degerlendirme'].astype(int)
         df['Hacim'] = df['Hacim'].astype(float)
```

Şekil 4: Kolon Bazında Manipülasyonlar

3.2 Hacim Kolonu

Hacim: İlk olarak hacim kolonundaki veriler float(ondalık) cinsine çevrilir, ardından *isnull metodu* ile kaç adet boş *null*(boş) değer olduğu sorgulanır. Ardından bu *null* değerlere bu kolonun ortalama değeri atanır.

```
df['Hacim'] = df['Hacim'].astype(float)
df['Hacim'].isnull().value_counts()
df['Hacim'].fillna((df['Hacim'].mean()), inplace = True)
df = df.round({'Hacim': 0})
```

Şekil 5: Hacim Kolonundaki İşlemler

3.3 Renk Kolonu

Renk: Bu kolondaki özgün değerler sorgulandığında fiyat aralığı çok değişmeyen, renk olarak aynı segmente hitap eden değerler incelenir ve ardından renk kolonundaki çeşitlilikler kendi içinde gruplanır. Siyah- Gri- Beyaz- Renk olmak üzere 4 grupta toparlanırlar.

Ardından bu renk seçenekleri *DataFrame*'de kolon olarak eklenmesi adına *Dummies metodu* uygulanarak yeni kolonlar eklenir. Yeni kolonların oluşturulma nedeni modelin daha iyi öğrenmesi adına özellik sayısının artırılması ve datanın detaylandırılmasıdır.

```
new_df1['Renk'].unique()
new_df1['Renk'].value_counts()
```

```
Beyaz          171
Inox           120
Gri             25
Siyah           20
Gümüş           17
Kırmızı          3
Koyu Mavi        2
Mavi             2
Bordo            2
Koyu Gri         2
Siyah - Gri      2
Yeşil            2
Açık Siyah       1
Açık Gri         1
Bej              1
Siyah Inox       1
Turuncu          1
Name: Renk, dtype: int64
```

```
new_df1["Renk"].replace({"Açık Gri": "Gri"}, inplace=True)
new_df1["Renk"].replace({"Bej": "Gri"}, inplace=True)
new_df1["Renk"].replace({"Gri": "Gri"}, inplace=True)
new_df1["Renk"].replace({"Gümüş": "Gri"}, inplace=True)
new_df1["Renk"].replace({"Inox": "Gri"}, inplace=True)
new_df1["Renk"].replace({"Koyu Gri": "Gri"}, inplace=True)
new_df1["Renk"].replace({"Siyah - Gri": "Gri"}, inplace=True)
new_df1["Renk"].replace({"Bordo": "Renkli"}, inplace=True)
new_df1["Renk"].replace({"Koyu Mavi": "Renkli"}, inplace=True)
new_df1["Renk"].replace({"Kırmızı": "Renkli"}, inplace=True)
new_df1["Renk"].replace({"Mavi": "Renkli"}, inplace=True)
new_df1["Renk"].replace({"Turuncu": "Renkli"}, inplace=True)
new_df1["Renk"].replace({"Açık Siyah": "Siyah"}, inplace=True)
new_df1["Renk"].replace({"Siyah": "Siyah"}, inplace=True)
new_df1["Renk"].replace({"Beyaz": "Beyaz"}, inplace=True)
new_df1 = pd.get_dummies(new_df1, columns=['Renk'], drop_first = True)
```

Şekil 6: Renk Kolonundaki İşlemle

3.4 Üretici Kolonu

Üretici: Bu kolon bazında da yine özgün değerlere bakılır ardından bu özgün değerlerin hangisinden kaç tane olduğuna bakılır ve bu çeşitliliklerin fiyatı nasıl etkilediği analiz edilir.
İlk aşama ürün sayısı fazla olan üretici firmalar kendi içinde gruplanarak kolonlar oluşturulur ancak markadan ziyade aslında fiyatı etkileyen asıl noktanın üretici firmanın yerli olup olmaması durumunun önemli bir kriter olduğu analiz edilir. Bu noktada üreticiler yerli ve yabancı noktasında ve kendi içinde gruplanarak yeni kolonlar oluşturulur.

```
new_df2['Uretici'].unique()
new_df2['Uretici'].value_counts()
```

```
Vestel        74
Bosch         62
Arçelik       42
Siemens       39
Samsung       33
Profilo       33
LG            21
Beko          14
Teka          11
Liebherr      11
Regal          9
Sharp          7
Franke         5
Uğur           3
Hoover         3
Electrolux     2
Finlux         1
Dijitsu        1
Altus          1
Name: Uretici, dtype: int64
```

```
new_df2["Uretici"].replace({"Vestel": "Yerli"}, inplace=True)
new_df2["Uretici"].replace({"Profilo": "Yerli"}, inplace=True)
new_df2["Uretici"].replace({"Arçelik": "Yerli"}, inplace=True)
new_df2["Uretici"].replace({"Beko": "Yerli"}, inplace=True)
new_df2["Uretici"].replace({"Regal": "Yerli"}, inplace=True)
new_df2["Uretici"].replace({"Altus": "Yerli"}, inplace=True)
new_df2["Uretici"].replace({"Uğur": "Yerli"}, inplace=True)
```

```
new_df2.dropna(subset=['Uretici'], inplace=True)
new_df2 = pd.get_dummies(new_df2, columns=['Uretici'], drop_first = True)
```

Şekil 7: Üretici Kolonundaki İşlemler

3.5 Enerji Sınıfı Kolonu

Enerji Sınıfı: Bu kolonda da genel incemeler yapılır ve neticesinde enerji sınıfının kendi içinde hiyerarşik olduğu gözlemlenir. Ve bunu modele anlatabilmek adına *dummies* işlemi yapılarak

oluşturulan kolonlara *scale_mapper metodu* uygulanarak modele bu hiyerarşik durum aktarılır.

```
new_df3['Enerji_Sinifi'].value_counts()
new_df3['Enerji_Sinifi'] = new_df3['Enerji_Sinifi'].replace('Yok', np.nan)
new_df3.dropna(subset=['Enerji_Sinifi'], inplace=True)

new_df3["Enerji_Sinifi"].replace({"F": "F"}, inplace=True)
new_df3["Enerji_Sinifi"].replace({"E": "E"}, inplace=True)
new_df3["Enerji_Sinifi"].replace({"A": "A"}, inplace=True)
new_df3["Enerji_Sinifi"].replace({"D": "D"}, inplace=True)
new_df3["Enerji_Sinifi"].replace({"G": "D"}, inplace=True)
```

```
new_df3['Enerji_Sinifi'].value_counts()
```

```
F    256
E     81
A     16
D      5
Name: Enerji_Sinifi, dtype: int64
```

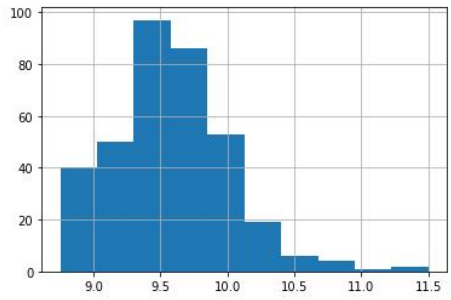
```
scale_mapper = {"A":1, "D": 2,"E":3 ,'F':4}
new_df3["Scale_Energy"] = new_df3["Enerji_Sinifi"].replace(scale_mapper)
```

Şekil 8: Enerji Sınıfı Kolonundaki İşlemler

3.5 Fiyat Kolonu

Bu kolon kendi içinde çok değişkenli(5 benzemez kuralı) olduğu için kendi içinde kümeleme, loglama işlemi ile datayı kendi içinde gruplaştırarak modelin daha iyi öğrenmesi sağlanır.

```
new_df4['log_price']=np.log(new_df4.Fiyat)
new_df4.log_price.hist();
```



Şekil 9: Fiyat Kolonundaki İşlemler

Son aşamada oluşturulan bu kolonlar liste haline getirilerek bu listeye subset ismi verilir.
Bu liste *DataFrame* içine yerleştirilerek model ile eğitilecek data hazır hale getirilir.

4. Veri ve Makine Öğrenimi İçin Gereksinimler

Bu noktada ilk olarak modele sokulacak datamız artık hazır noktada bulunmaktadır.

İlk etapta data train ve test olmak üzere data 2 parçaya ayrılır, ardından train data içinden cross validation yapmak adına data tekrar bölünür. Böylece: eğitim için ve modelin ne kadar başarılı olduğunu gözlemlemek adına , test ve validation datası mevcut hale gelmiş olur.

```
X, y = new_df7.drop('log_price',axis=1), new_df7['log_price']

train = new_df7
test = new_df7

x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.10, random_state=20)
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size=0.20, random_state=20)
```

Şekil 10: Train - Test - Validation Data

Ardından çalışmak istenilen modeller tanıtılır.

Bu proje kapsamında 3 farklı makine modeli kullanılmaktadır:

- *LinearRegression Modeli*
- *Ridge Modeli*
- *Lasso Modeli*

Bu 3 model data üzerinde çalıştırılır ve farklı oranlarda başarılar elde edilir. 3 model arasında en başarılı sonuç veren model ***LinearRegression Modeli*** ‘dir.

Ardından modeli geliştirmek adına basic feature engineering: adding polynomial terms ile hacim kolonu yenilenir, sonrasında model üzerinde önemli etkiye sahip olan 3 özellik kullanılarak yeni kolonlar oluşturulur ve model başarısının arttığı gözlemlenir.

```
#Yillik_Enerji_Tuketimi
#Uretici_Franke
#Hacim

new_df9 = new_df7.copy()

# multiplicative interaction
new_df9['Y_E_Tuketimi_Hacim'] = new_df9['Yillik_Enerji_Tuketimi'] * new_df9['Hacim']
new_df9['Hacim'] = new_df9['Hacim'] ** 2
new_df9['New_Feature_3'] = new_df9['Yillik_Enerji_Tuketimi'] * new_df9['Scale_Energy'] * new_df9['Hacim']
```

Şekil 11: Basic feature engineering: adding polynomial terms ana interaction terms

Data ile çalıştırılan modeller ve başarıları şekildeki gibidir.

```
print('\n','lr_model_test:', score_lr_model_test, '\n',
      '\n','lr2_model_test', score_lr2_model_test, '\n'
      '\n','ridgeReg_test', score_ridgeReg_test, '\n'
      '\n','lassoReg_test',score_lassoReg_test, '\n')
```

```
lr_model_test: 47.66

lr2_model_test 50.36

ridgeReg_test 49.34

lassoReg_test -9.46
```

Şekil 12: Modeller ve Başarı Değerleri

5. Sonuçlar

Bu çalışma buzdolabı fiyatını tahmin etmek üzere Hepsi Burada web sitesinden datalar çekilerek, bu datalar ile fiyat tahmin etmek üzere makine öğrenmesi projesidir. Bu kapsamda farklı kaynaklardan elde edilen buzdolabı dataları düzenlenerek modelde kullanılarak model başarısı arttırılabilir. Şuan model başarısı en yüksek olan model LinearRegression Modelidir ve bu kapsamda datalar bu model üzerinde beslenerek tahmin başarı skoru %50,36'dan %80'lere ve %90'lara çıkartılabilir. Bu noktada fiyatı etkileyen özellik kapsamı arttırılması gerekmektedir. Mevcut dataların verdiği başarı bununla sınırlı kalmaktadır bunun sebebi ürün özelliklerinin eksikliği ve datanın az olmasından kaynaklıdır. Bu noktalar beslenerek skor arttırılır.

6. Kaynaklar

[1] HepsiBurada, "no-frost-buzdolabı" https://www.hepsiburada.com/ara?q=no-frost%20buzdolab%C4%B1&kategori=2147483637_235604&filtreler=fiyat:2500-max&sayfa=1 ", Son erişim tarihi : 05.06.2022

[2] Python-BeatifulSoup kütüphanesi, data çekme, "https://www.dataquest.io/blog/web-scraping-python-using-beautiful-soup/ ", Son erişim tarihi : 05.06.2022

[3] Python-BeatifulSoup kütüphanesi, data çekme, "https://realpython.com/beautiful-soup-web-scraper-python/ ", Son erişim tarihi : 05.06.2022

[4] Python-Matplotlib Kütüphanesi, "Python-Matplotlib kütüphanesi, grafik oluşturma "https://matplotlib.org/ ", Son erişim tarihi : 05.06.2022

[5] Python-Heatmap-Seaborn kütüphanesi, data çekme, "https://seaborn.pydata.org/generated/seaborn.heatmap.html", Son erişim tarihi : 05.06.2022

[6] Python-Linear Regression, model ile data eğitimi, "https://realpython.com/linear-regression-in-python/ ", Son erişim tarihi : 05.06.2022

[7] Python-Ridge/Lasso, model ile data eğitimi, "https://www.analyticsvidhya.com/blog/2016/01/ridge-lasso-regression-python-complete-tutorial/ ", Son erişim tarihi : 05.06.2022