

**BERRİN GÖÇER**

**090150502**

**05.06.2022**

**MAT 492**

**ÖĞR. GÖR. ÜYESİ EVREN TANRIÖVER**

Hazırlayan :

Öğrenci No :

Teslim Tarihi :

Ders :

Danışman :

TASARIM (BİTİRME) PROJESİNİN BAŞLIĞI

İçindekiler Tablosu

[Şekiller Tablosu 2](#_Toc60682892)

[1. Giriş 3](#_Toc60682893)

[2. Web Sitesinden Datanın Elde edilmesi 3](#_Toc60682894)

[3. Veri Toplanması ve Manipülasyon 5](#_Toc60682895)

[3.1 Genel Data Bilgisi 5](#_Toc60682896)

[3.2 Hacim Kolonu 6](#_Toc60682897)

[3.3 Renk Kolonu 7](#_Toc60682898)

[3.4 Üretici Kolonu 7](#_Toc60682897)

[3.5 Enerji Sınıfı Kolonu 8](#_Toc60682897)

[3.6 Fiyat Kolonu 9](#_Toc60682897)

[4. Veri ve Makine Öğrenimi İçin Gereksinimler 9](#_Toc60682899)

[5. Sonuçlar 1](#_Toc60682899)0

***6. Kaynaklar……………………………………………………………………………………………………………………10***

# Şekiller Tablosu

[Şekil 1: Siteden Çekilen HTML Tipi Data. 4](#_Toc60481172)

[Şekil 2: Siteden Çekilen Data İçeriği. 5](#_Toc60481172)

[Şekil 3: Elde Edilen DataFrame. 5](#_Toc60481172)

[Şekil 4: Kolon Bazında Manipülasyonlar. 6](#_Toc60481172)

Şekil 5: Hacim Kolonundaki İşlemler..........................................................................................6

[Şekil 6: Renk Kolonundaki İşlemler. 7](#_Toc60481172)

[Şekil 7: Üretici Kolonundaki İşlemler 7](#_Toc60481172)

[Şekil 8: Enerji Sınıfı Kolonundaki İşlemler 8](#_Toc60481172)

[Şekil 9: Fiyat Kolonundaki İşlemler. 9](#_Toc60481172)

[Şekil 10: Train - Test Validation Data 9](#_Toc60481172)

[Şekil 11: Modeller ve Score Değerleri 1](#_Toc60481172)0

# 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 buzdolap fiyatlarını etkileyen önemli kriterlerin neler olduğu incelenmiş ve bu özellikler kullanılarak en doğru tahmini yapabilmek adına modelleme yapılarak buzdolabı fiyat tahmini yapılması amaçlanmıştır.

# Proje kapsamında python yazılım dili kullanılmıştır. Dataların web sitesinden çekilmesi adına Python *BeatifulSoup* kütüphanesi kullanılarak datalar otomatize olarak web sitesinden çekilmiş ve ardından çekilen bu datalar proje ortamı olarak kullanılan Jupyter Notebook’a aktarılmıştır. Sondasında çekilen datalar Python: *Numpy-Pandas* kütüphaneleri kullanılarak dataframe haline getirilmiş üzerinde analizler gerçekleştirilmiştir. Elde edilen analizleri güçlendirmek ve doğru bir şekilde incelemek adına *seaborn* ve *matplotlib* kütüphaneleri ile bu analizler görsel grafik haline getirilmişlerdir. Datalar üzerindeki işlemeler tamamlandıktan sonra elde edilen data incelendiğinde amacımız olan buzdalabı fiyatını etkileyen faktörlerin aslında fiyata doğrusal bir şekilde etki ettiği gözlemlenmiştir ve ardından datalar: Train-test olmak üzere 2 parçaya ayrılmıştır. Sondasında test datası üzerinde alınan tahminlerin doğruluğunun daha iyi analiz edilmesi adına cross validation yapılarak test datası da kendi içinde parçalara ayrılmıştır. Son aşamada eğitim için ayrılan data *LinearRegression, Ridge* ve *Lasso makine* modelleri ile eğitilerek ve bu modellerde test datası kullanılarak tahminler yapılmış ve modelin doğruluk değeri ölçülmüştür.

# Web Sitesinden Datanın Elde Edilmesi

Bu proje kapsamında HepsiBurada web sitesinde bulunan XX adet buzdolabı XX sayfalık web sayfasında yer almaktadır. (Web sitesinde birçok sayfa ve her sayfada da birçok ürün bulunmaktadır.) Sayfa içersindeki datalar içinde birçok ürün bulunmaktadır ve ürün bazında özgün linkler bulunmaktadır. Bu linklerin yerleri belli modele göre oluşturulmuştur 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 istediğimiz web sayfasına *request* methodunun içerisine web sayfasının linkini yerleştirerek sayfaya istek atıyoruz. Ardından *get* methodu yardımı ile data çekmek istediğimizi bildiriyoruz. Bu şekilde sayfa özelinde bütün dataları HTML dilinde *parse* olarak elde ediyoruz. Sayfa üzerindeki tek bir ürünlerin linklerine erişim adına *FindAll* ile elde edilen bu toplu data içerisinden ürünlerin linklerine erişiyoruz. Artık bir ürünün içerisine girdiğimiz noktada bir fonksiyon oluşturarak sayfalar üzerinden geçiş sağlamak adına linklerin sonundaki sayfa numaralarını kullanarak sayfalar içinde geçişler ve sayfalardaki ürünlere erişim sağlanmış olur. Son kısım olarak yine tek bir ürün bazındaki özelliklerin yerini tespit ediyoruz, *html* dilinde bu konumlar belirli başlıklar altında belirli kurallara uygun olarak yerleştirilmişlerdir. Bu dizin yerlerini göstererek çektiğimiz bu fonksiyona *product* ismini veriyoruz. (Fonksiyon ismi opsiyoneldir, değiştirilebilir.) Bu yerleri öğrenmek adına web sitesinde erişmek istediğimiz özellik(yazı) üzerinde fare ile sağ click yaparak *özellikler* yazısına tıklıyoruz ve *HTML dilinde* bu özelliklerin konumlarının nerede olduğunu gözlemleriz ve *html*.*find* methodu içerisine bu konumlar bildirilerek bu datalarda çekilir. Ardından bu işlemi tekrar bir fonksiyon içerisine yerleştirilir. Artık tek seferde toplu olarak ürün üzerinde dataları çekmiş bulunuruz.



Şekil 1: Siteden Çekilen HTML Tipi Data.



Şekil 2: Çekilen Data İçeriği.

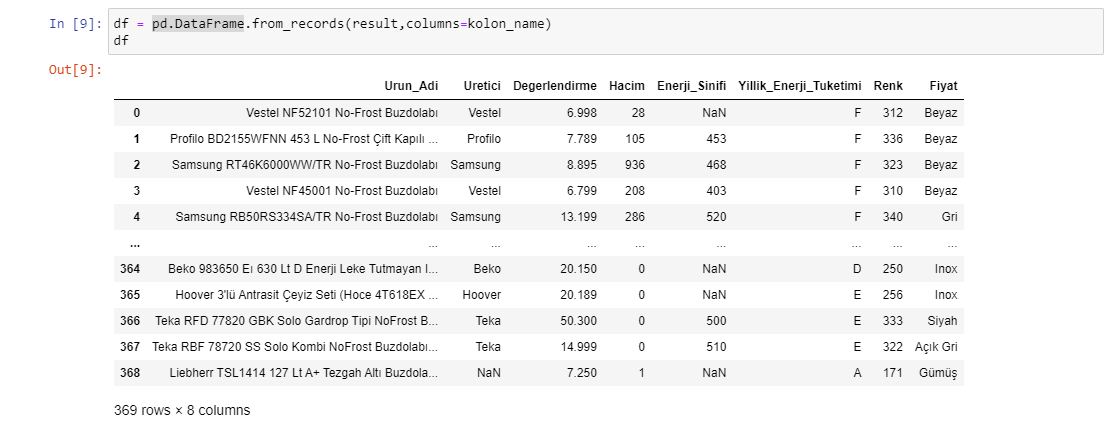
# 

# Veri Toplanması ve Manipülasyon

Şablonun bu kısmında metin boyunca kullanılacak biçim, şekil ve denklem kuralları anlatılmaktadır. Alt başlıkların sayısı istenildiği kadar arttırılabilir. Ana ve alt başlıkların tümü doğru sayfa numaralandırmalarıyla içindekiler kısmında bulunmalıdır.

3.1 Genel Data Bilgisi

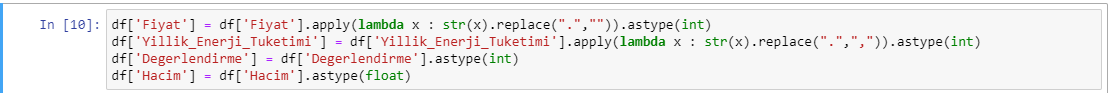
İlk aşamada web sitesinden çekilen datalar üzerinde analizler gerçekleştirebilmek adına data *pd.DataFrame methodu* ile data *DataFrame* haline getirilir.(Çekilen datalar *string* formundadırlar).



Şekil 3: Elde Edilen DataFrame.

Şekil X den de gördüğümüz gibi ilk olarak çektiğimiz data buzdolaplarına dair:  
*Ürün Adı - Üreti Firma - Web sitedeki Değerlendirme Sayısı - Eneji Sınıfı - Yıllık Enerji Tüketimi - Renk - Fiyat* özellikleri çekilmiştir. Data 8 kolon ve satır(ürün sayısını temsil eder)’dan oluşmaktadır.

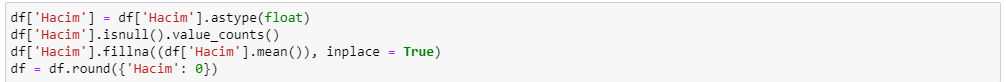
Ardından *Fiyat özelliği* içerisindeki tüm datalar üzerinde işlem yapmak addına *apply methodu* ile *float* cinsindeki datadan nokta kaldırılarak *integer* tipine çevrilmiştir. Bu data özelinde işlem sonrasında *Yıllık Enerji Tüketimi* ve *Değerlendirme* özellikleri üzerinde de apply methodu ile aynı işlem bu kolonlara da uygulanmıştır. Diğer bir özelliğimiz olan *Hacim* kolonu da *Float tipine* çevrilmiştir.



Şekil 4: Kolon Bazında Manipülasyonlar.

### 3.2 Hacim Kolonu

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

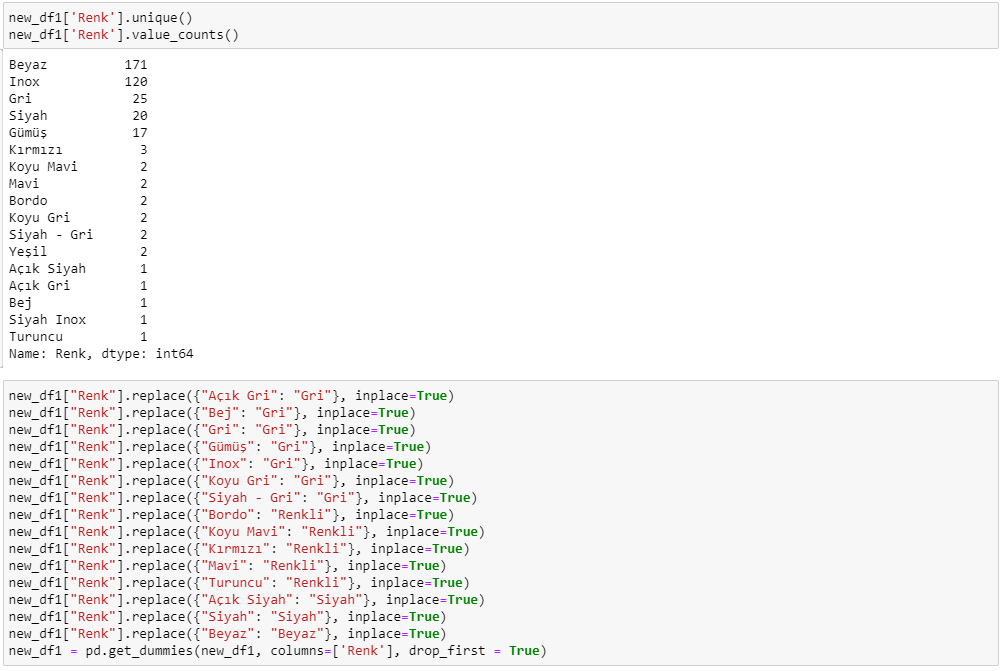


Şekil 5: Hacim Kolonundaki İşlemler.

### 3.3 Renk Kolonu

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

Ardından bu renk seçenekler *DataFrame*’de kolon olarak eklenmesi adına *Dummies methodu* uygulanarak yeni kolonlar eklenmiştir. Yeni kolonların oluşturulma nedeni modelin daha iyi öğrenmesi adına özellik sayısının artırılması ve datanın detaylandırılmasıdır.



Şekil 6: Renk Kolonundaki İşlemler.

### 3.4 Üretici Kolonu

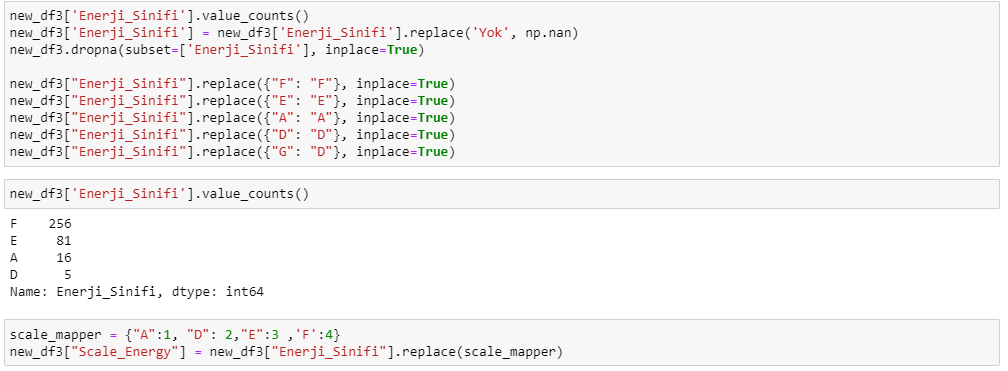
Üretici: Bu kolon bazında da yine özgün değerlere bakılmış ardından bu özgün değerlerin hangisinden kaç tane olduğuna bakılmış ve bu çeşitliliklerin fiyatı nasıl etkilediği analiz edilmiştir. İlk aşama ürün sayısı fazla olan üretici firmalar kendi içinde gruplanarak kolonlar oluşturulmuş ancak markadan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



Şekil 7: Üretici Kolonundaki İşlemler.

### 3.5 Enerji Sınıfı Kolonu

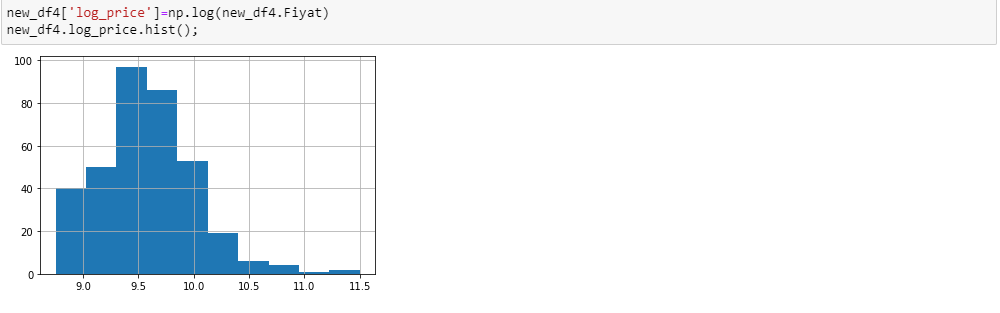
Enerji Sınıfı: Bu kolonda da genel incemeler yapılmış ve neticesinde enerji sınıfının kendi içinde hiyerarşik olduğu gözlemlenmiştir. Ve bunu modele anlatabilmek adına *dummies* işlemi yapılarak oluşturulan kolonlara *scale\_mapper mothodu* uygulanarak modele bu hiyerarşik durum anlatılmıştır.



Şekil 8: Enerji Sınıfı Kolonundaki İşlemler.

### 3.5 Fiyat Kolonu

Bu kolon kendi içinde çok değişkenli olduğu için kendi içinde kümeleme, loglama işlemi ile datayı kendi içinde gruplaştırarak modelin daha iyi öğrenmesini sağlıyoruz.



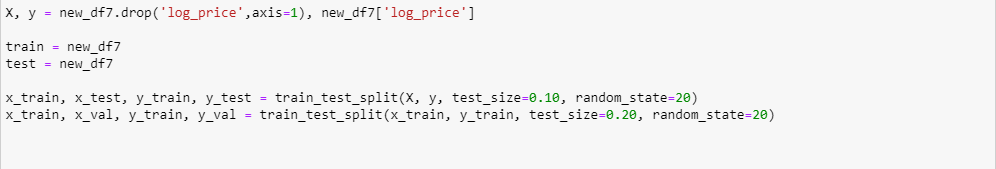
Şekil 9: Fiyat Kolonundaki İşlemler.

Son aşamada oluşturulan bu kolonlar liste haline gelirilerek bu listeye subset ismi verilmiştir.

Bu liste *DataFrame* içine yedirilerek modele sokacağımız datamız hazır hale getirilmiştir.

# 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 elimizde: eğitim için ve modelin ne kadar başarılı olduğunu gözlemlemek adına elimizde, test ve validation datamız mevcut hale gelmiş oluyor.



Şekil 10: Train - Test - Validation Data.

Ardından çalışmak istediğimiz modelleri tanıtıyoruz.  
Bu proje kapsamında 3 farklı makine modeli kullanılmıştır:

*LinearRegression Modeli*

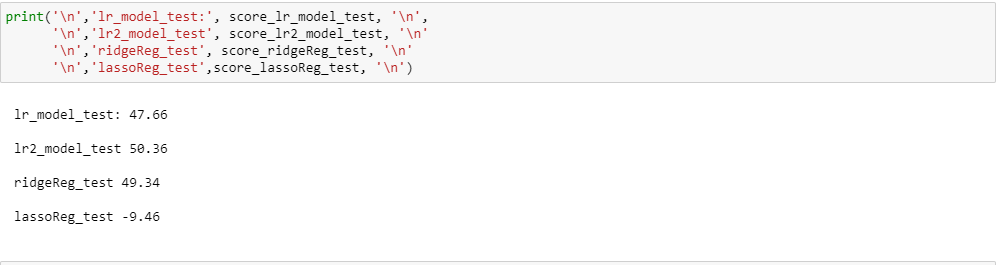
*Ridge Modeli*

*Lasso Modeli*  
Bu 3 model data üzerinde çalıştırılmış ve farklı oranlarda başarılar elde edilmiştir.

Bu 3 model arasında en başarılı sonuç veren model *LinearRegression Modeli* olmuştur.  
Ardından modeli geliştirmek adına basic feature engineering: adding polynomial terms ile hacim kolonu yenilenmiş, sonrasında model üzerinde önemli etkiye sahip olan 3 özellik kullanılarak yeni kolonlar oluşturulmuş ve model başarısının arttığı gözlemlenmiştir.



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

*Datamız ile çalıştırdığımız modeller ve başarıları şekildeki gibidir.*

Şekil 11: Modeller ve Score Değerleri

# Sonuçlar

Bu çalışma buzdolap fiyatını tahmin etmek üzere HepsiBurada web sitesinden datalar çekilmiş ve bu datalar ile fiyat tahminleme üzere makine öğrenmesi projesidir. Bu kapsamda farklı kaynaklardan elde edilen buzdolap 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, özellik eksikliği ve datanın az olmasından dolayı. Bu noktalar beslenmelidir.

# 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 : 16.06.2022