Ödev 2

1- Import Boston Dataset from sklearn dataset class.

Scikit-learn, bazı harici web sitelerinden herhangi bir dosya indirmeyi gerektirmeyen birkaç küçük standart veri kümesiyle birlikte gelir.

Aşağıdaki işlevler kullanılarak yüklenebilirler:

|  |  |
| --- | --- |
| [load\_boston](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html#sklearn.datasets.load_boston)(\* [, return\_X\_y]) | Boston ev fiyatları veri kümesini (regresyon) yükleyin ve iade edin. |
| [load\_iris](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_iris.html#sklearn.datasets.load_iris)(\* [, return\_X\_y, as\_frame]) | İris veri kümesini (sınıflandırma) yükleyin ve iade edin. |
| [load\_diabetes](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html#sklearn.datasets.load_diabetes)(\* [, return\_X\_y, as\_frame]) | Diyabet veri setini yükleyin ve iade edin (regresyon). |
| [load\_digits](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html#sklearn.datasets.load_digits)(\* [, n\_class, return\_X\_y, as\_frame]) | Basamak veri kümesini (sınıflandırma) yükleyin ve iade edin. |
| [load\_linnerud](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_linnerud.html#sklearn.datasets.load_linnerud)(\* [, return\_X\_y, as\_frame]) | Fiziksel egzersiz linnerud veri kümesini yükleyin ve iade edin. |
| [load\_wine](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_wine.html#sklearn.datasets.load_wine)(\* [, return\_X\_y, as\_frame]) | Şarap veri setini yükleyin ve iade edin (sınıflandırma). |
| [load\_breast\_cancer](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html#sklearn.datasets.load_breast_cancer)(\* [, return\_X\_y, as\_frame]) | Meme kanseri wisconsin veri setini (sınıflandırma) yükleyin ve iade edin. |

Bu veri kümeleri, scikit-learn'de uygulanan çeşitli algoritmaların davranışını hızlı bir şekilde göstermek için kullanışlıdır. Bununla birlikte, gerçek dünyadaki makine öğrenimi görevlerinin temsilcisi olamayacak kadar küçüktürler.

2- Explore and analyse raw data.

Ham veri, Veri bilimi açısından kaynağından toplanıp istiflenmiş ama analize hazır hale getirilmemiş veridir.

Ham veriyi belki de [büyük veri](https://www.datasciencearth.com/big-data-nedir/) ve küçük veri içerisinde ayrı ayrı değerlendirmek daha isabetli olacaktır. Şimdi biraz bunları kısaca inceleyelim.

1-Küçük ham veri

Adından da anlaşılabileceği gibi küçük ölçekli ve kendi içerisinde bir düzen ile birlikte biriktirilmiş veri diyebiliriz buna. Örneğin ilişkisel veri tabanlarında birbirleri arasındaki ilişkiler ile bağlı olan tablolarda tutulan veri.

Örneğin siz bir markette analistsiniz ve marketin satışlarını arttırmak için bazı analizler yapmak istiyorsunuz. Aklınıza sepet analizi yapmak geldi. [İlişkisel veri tabanınızda](https://www.datasciencearth.com/sql-nedir-dml-ddl-dcl-dql-komutlari/) bir invoice ve invoiceRow tabloları var. Bu tablolardaki veri sizin için hamdır. Bu veriyi bir [birliktelik analizi](https://www.datasciencearth.com/birliktelik-kurallari-ve-uygulama-alanlari/) algoritmasına sokmak için bu ham veriden bir veriset oluşturmanız gerekiyor.

2-Büyük ham veri

İşte asıl ham verinin olduğu yer büyük veri kısmıdır. Belki burada veri göllerinden de bahsetmek gerekir.

Veri Gölü yapılandırılmamış verilerin istiflediğini toplama alanlarıdır. Genelde kurumlar işlemesi masraflı ve güncel işleme yöntemleri ile net analizler yapamayacakları verileri istiflerler. Ta ki o verileri işlemek için harcanacak para, veriyi işleyerek elde edilecek veri tabanlı ürünlerin yaratacağı gelirden daha düşük olacağı zamana yani kar edilebilir şartlar oluşuncaya kadar bu istifler olduklarıdır. Makine ya da elektronik cihazların çalışmasından ortaya çıkan veri kayıt altına alınır.

Örneğin X fabrikasının üretim bandını izleyen 1.000 sensör olabilir ve bu her bir sensör saniyede bir veri üretir.Saniye de 1.000 veri, dakikada 60.000 veri, saatte 3.600.000 veri … Ve tabi bir şirketin 10 fabrikası olduğu durumlarda var. Dünya da ise sayısız fabrika var.

3- Do preprocessing for regression

Veri ön işleme, ham verilerin anlaşılabilir bir biçime dönüştürülmesini içeren bir veri madenciliği tekniğidir. Gerçek dünya verileri genellikle eksiktir: öznitelik değerleri eksiktir, belirli ilgi özniteliklerinden yoksundur veya yalnızca toplu verileri içerir, Gürültülü: Hatalar veya aykırı değerler içerir. Tutarsız: Kodlarda veya adlarda tutarsızlıklar içeren. Veri ön işleme, bu tür sorunları çözmek için kanıtlanmış bir yöntemdir.

Gerçek dünyada veriler genellikle eksiktir: öznitelik değerlerinin olmaması, ilgilenilen belirli özniteliklerin olmaması veya yalnızca toplu verileri içermesi.

Gürültülü: Hatalar veya aykırı değerler içerir.

Tutarsız: Kodlarda veya adlarda tutarsızlıklar içeren.

**1.1) Veri Ön İşleme Adımları**

Adım 1: Kitaplıkları içe aktarın  
Adım 2: Veri kümesini içe aktarın  
Adım 3: Eksik değerleri kontrol edin  
Adım 4: Kategorik verileri kodlayın  
Adım 5: Veri kümesini Eğitim ve Test kümesine bölme  
Adım 6: Özellik ölçekleme

Tüm bu adımları ayrıntılı olarak tartışalım.

**1. Adım: Kitaplıkları içe aktarın**

Bir  kütüphane, aynı zamanda, davranışın çağrıldığı iyi tanımlanmış bir arayüze sahip, bir dil açısından yazılmış bir davranış uygulamaları koleksiyonudur. Örneğin, daha üst düzey bir program yazmak isteyen kişiler, bu sistem çağrılarını tekrar tekrar uygulamak yerine sistem çağrıları yapmak için bir kitaplık kullanabilir. 3 temel python kütüphanesini içe aktarmamız gerekiyor.

1.  **[Numpy](https://www.numpy.org/?source=post_page---------------------------" \t "_blank)**  , Python ile bilimsel hesaplama için temel pakettir.

2.  [**Pandalar**](https://pandas.pydata.org/?source=post_page---------------------------) *için* veri işleme ve analizi*.*

3.  **[Matplotlib](https://matplotlib.org/?source=post_page---------------------------" \t "_blank)**  , çeşitli basılı kopya formatlarında ve platformlar arasında etkileşimli ortamlarda yayın kalitesinde rakamlar üreten bir Python 2D çizim kitaplığıdır.

içe aktarma numpy

içe aktar matplotlib.pyplot plt olarak

içe aktar pandaları pd olarak içe aktar

**2. Adım: Veri kümesini içe aktarın**

Veriler, pandas kitaplığı kullanılarak içe aktarılır.

data = pd.read\_csv ('/ path\_of\_your-dataset / Data.csv')

X = data.iloc [:,: - 1] .values

y = data.iloc [:, 3] .values

Burada,  **X**, bağımsız değişkenlerin bir matrisini ve  **y**, bağımlı değişkenin bir vektörünü temsil eder.

**3. Adım: Eksik değerleri kontrol edin**

Veri setimizde eksik değerleri ele almanın iki yolu vardır. Boş değerleri işlemek için yaygın olarak kullanılan ilk yöntem. Burada, belirli bir özellik için boş bir değere sahipse belirli bir satırı ve eksik değerlerin% 75'inden fazlasına sahipse belirli bir sütunu sileriz. Bu yöntem, yalnızca veri setinde yeterli örnek olduğunda önerilir. Verileri sildikten sonra hiçbir önyargı eklenmediğinden emin olunmalıdır.

İkinci yöntemde, tüm NaN değerlerini ortalama, medyan veya en sık değerle değiştiririz. Bu, veri setine varyans ekleyebilecek bir yaklaşımdır. Ancak veri kaybı, satırların ve sütunların kaldırılmasına kıyasla daha iyi sonuçlar veren bu yöntemle reddedilebilir. Yukarıdaki üç yaklaşımla değiştirmek, eksik değerleri ele almak için istatistiksel bir yaklaşımdır. Bu yöntem aynı zamanda  eğitim sırasında verilerinsızdırılması olarak da adlandırılır.

Eksik verilerle başa çımak için ‘’sklearn.processing’’ paketindeki Imputer kütüphanesini kullanırız. Ortalama sağlamak yerine, strateji parametresinde medyan veya en sık kullanılan değeri de sağlayabiliriz.

4- Split your dataset into train and test test (0.7 for train and 0.3 for test).

Bu sorunun cevabı koda yazılmıştır.

5- Try Ridge and Lasso Regression models with at least 5 different alpha value for each.

Bu sorunun cevabı koda yazılmıştır.

6- Evaluate the results of all models and choose the best performing model.

**1- Anahtar Farkı**

**Ridge:** Modeldeki özelliklerin tümünü (veya hiçbirini) içerir. Bu nedenle, sırt regresyonunun en büyük avantajı, katsayı büzülmesi ve model karmaşıklığının azaltılmasıdır.

**Lasso:** Küçülme katsayılarının yanı sıra, kement, özellik seçimi de gerçekleştirir. ( Tam kement formundaki ' *seçimi* ' hatırlıyor musunuz?) Daha önce gözlemlediğimiz gibi, bazı katsayılar tam olarak sıfır olur, bu da belirli özelliğin modelden çıkarılmasına eşdeğerdir.

Geleneksel olarak, özellik seçimi yapmak ve cimri modeller yapmak için aşamalı regresyon gibi teknikler kullanıldı. Ancak Makine Öğrenimindeki gelişmelerle birlikte, ridge ve lasso regresyon, çok daha iyiçıktılar sağladıkları,daha azayarlamaparametresi gerektirdikleri ve büyük ölçüde otomatikleştirilebildikleri için çok iyi alternatifler sağlar .

2. Tipik Kullanım Durumları

**Ridge:** Büyük ölçüde *aşırı oturmayı önlemek için* kullanılır. Tüm özellikleri içerdiğinden, aşırı yüksek # özelliklerde, örneğin milyonlarca, hesaplama zorlukları oluşturacağından çok kullanışlı değildir.

**Lasso:**  o sağladığından *seyrek çözümler*, genellikle #features milyonlarca ya da daha fazla olduğu durumları modelleme için seçim modeli (veya bu kavramın bazı varyant) 'dir. Böyle bir durumda, seyrek bir çözüm elde etmek büyük bir hesaplama avantajı sağlar çünkü sıfır katsayılı özellikler basitçe göz ardı edilebilir.

Aşamalı seçim tekniklerinin yüksek boyutluluk durumlarında uygulanmasının neden pratik olarak çok külfetli hale geldiğini anlamak zor değil. Böylece kement önemli bir avantaj sağlar.

3. Son Derece İlişkili Özelliklerin Varlığı

**Ridge:** Modele hepsini dahil edeceği için yüksek korelasyonlu özelliklerin varlığında bile genellikle iyi çalışır, ancak katsayılar korelasyona bağlı olarak aralarında dağıtılacaktır.

**Lasso:** Son derece ilişkili olanlar arasından herhangi bir özelliği keyfi olarak seçer ve geri kalanların katsayılarını sıfıra indirir. Ayrıca, seçilen değişken, model parametrelerindeki değişiklikle rastgele değişir. Bu genellikle sırt regresyonuna kıyasla o kadar iyi çalışmaz.

Kementin bu dezavantajı, yukarıda tartıştığımız örnekte görülebilir. Bir polinom regresyon kullandığımız için, değişkenler oldukça korelasyonluydu. (Neden emin değil misiniz? Data.corr () çıktısını kontrol edin). Böylece, küçük alfa değerlerinin bile önemli seyreklik verdiğini gördük (yani sıfır olarak yüksek katsayılar).

Ridge ve Lasso ile birlikte Elastic Net, hem L1 hem de L2 düzenliliğini birleştiren başka bir kullanışlı tekniktir. Sırt ve kement regresyonunun artılarını ve eksilerini dengelemek için kullanılabilir. Daha fazlasını keşfetmenizi tavsiye ederim.