

**MÜHENDİSLİK FAKÜLTESİ**

**PROJE RAPORU**

|  |  |
| --- | --- |
| Öğrencinin | |
| Adı- Soyadı: | **TEOMAN YALÇİNÖZ** |
| Numarası: | **190101070** |
| Bölümü: | **Bilgisayar Mühendisliği** |

|  |  |
| --- | --- |
| Öğrencinin | |
| Adı- Soyadı: | **ALP CAN BELEN** |
| Numarası: | **180** |
| Bölümü: | **Bilgisayar Mühendisliği** |

**TANIM:**

**Elmas Nedir?**

**Elmas, bilinen en sert maddelerden biridir ve değerli bir taştır. Karbon elementinin bir modifikasyonu grafit, diğeri ise elmastır.**

**Projenin amacı nedir?**

**Elmas veri setinde bulunan elmas on farklı özelliğini yapay sinir ağları yardımı ile makineye öğreterek özellikleri verilen elmasların fiyatını tahmin ettirmektir.**

**Girişler nelerdir?**

1. **Elmasın karatı**
2. **Elmasın kesimi**
3. **Elmasın rengi**
4. **Elmasın berraklığı**
5. **Elmasın derinliği**
6. **Elmasın masası (yüzeyindeki düz fasettir)**
7. **X eksenindeki büyüklüğü**
8. **Y eksenindeki büyüklüğü**
9. **Z eksenindeki büyüklüğü**

**Çıkışlar nelerdir?**

1. **Elmasın Fiyatı**

**Veri Seti nereden alındı?**

**Goggle’ın kurmuş olduğu kaggle internet sitesinden alındı.**

[**https://www.kaggle.com/shivam2503/diamonds**](https://www.kaggle.com/shivam2503/diamonds)

**Projenin faydası nedir?**

**Birden fazla özellik yardımı ile elmasların fiyatını daha rahat tespit edebiliriz.**

**METOTLAR**

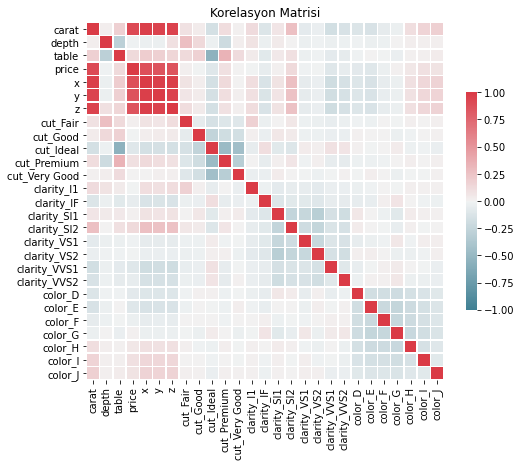
**Projenin yapım aşamasında Anaconda Jupyter Notebook programı üzerinden Python programlama dili kullanıldı.**

**Projede genel olarak yapay sinir ağları algoritmalarından aktivasyon fonksiyonları kullanılmıştır**

**Nöron değerleri fonksiyona sokarak fonksiyonun kuralına göre çıkış değeri bulan algoritmadır.**

**ReLU Fonksiyonu: Doğrultulmuş lineer birim (rectified linear unit- RELU) doğrusal olmayan bir fonksiyondur. ReLU fonksiyonu negatif girdiler için 0 değerini alırken, x pozitif girdiler için x değerini almaktadır.**

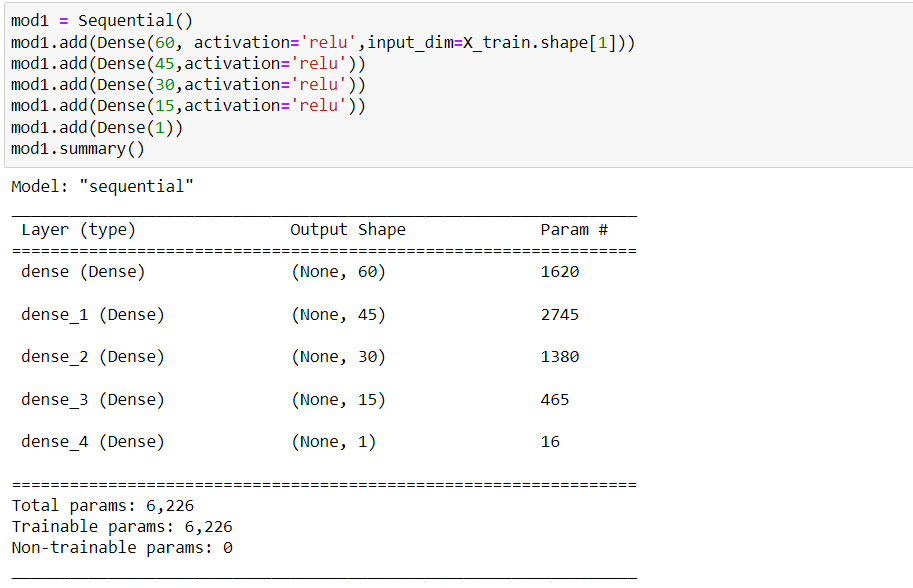
**SONUÇ**

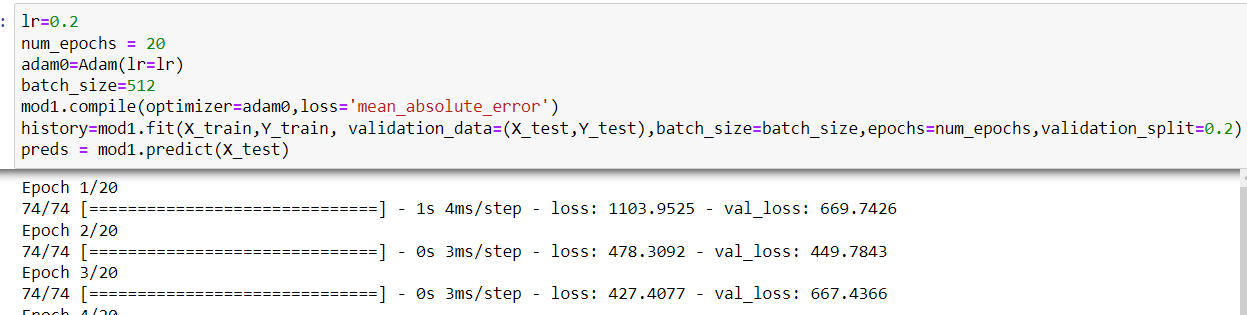
****

**Çoklu değişkenler arasındaki korelasyon katsayılarının tablosudur. Bu tabloda bir değişkenin diğer her değişken ile arasındaki korelasyon görülebilir.**

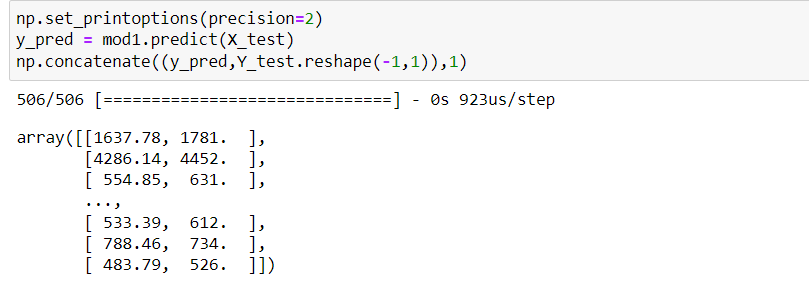
**Model Karşılaştırmaları**

**Model 1**

Bu kod, Keras kütüphanesinden boş bir yapay sinir ağı modelini tanımlar. İlk satır mod1 = Sequential() boş bir sıralı model oluşturur. Aşağıdaki satırlar daha sonra add() yöntemini kullanarak modele gizli katmanlar ekler. Her katmanın, ilk bağımsız değişken tarafından belirtilen (örneğin, ilk katmanda 60 birim) bir dizi birimi (düğümler veya nöronlar olarak da adlandırılır) vardır. Her katman için aktivasyon fonksiyonu, aktivasyon bağımsız değişkeni ile belirtilir. İlk katmandaki input\_dim bağımsız değişkeni, girdi verilerindeki özelliklerin sayısını belirtir. Son katmanın bir birimi vardır ve aktivasyon fonksiyonu yoktur, bu da doğrusal bir katman olacağı anlamına gelir. Sonunda summary() metodu, modeldeki katman sayısı ve toplam parametre sayısı (ağırlıklar ve sapmalar) dahil olmak üzere model mimarisinin bir özetini yazdırır.



Bu kod, sinir ağı modeli için eğitim sürecini yapılandırır. İlk satır lr=0.2, öğrenme oranını 0.2 olarak ayarlar. num\_epochs değişkeni 20'ye ayarlanır, bu da modelin 20 eğitim turu olarak eğitileceği anlamına gelir (tüm eğitim veri kümesi üzerinde yinelemeler). Adam optimizasyon algoritması daha sonra belirtilen öğrenme hızıyla başlatılır ve adam0 değişkeninde depolanır. Batch\_size değişkeni 512 olarak ayarlanmıştır, bu da modelin 512 numunelik mini partiler kullanılarak eğitileceği anlamına gelir. Model daha sonra derleme() yöntemi kullanılarak, Adam optimizasyon alg. ve argüman olarak belirtilen ortalama\_absolute\_error kayıp işleviyle derlenir.



Bu kod, y\_pred'deki tahmin edilen değerleri Y\_test'teki gerçek değerlerle birleştiriyor ve her iki değer kümesiyle yeni bir NumPy dizisi döndürüyor. Ortaya çıkan dizi, ilk sütunda tahmin edilen değerler ve ikinci sütunda gerçek değerler olmak üzere iki sütuna sahip olacaktır.

|  |  |
| --- | --- |
|  |  |

MODEL 1

-------

Hidden Layer : 5

Learning Rate: 0.2

Epoch: 20

Batch size: 512

r\_square: 0.9581115959174666

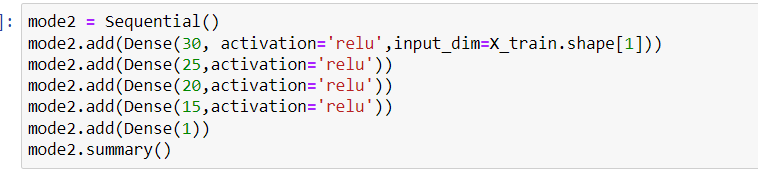
mean\_absolute\_error: 330.32771770292004

mean\_squared\_error: 673517.7413035524

explained\_variance\_score: 0.958146572407924

**Train los tahmin ettirilen veridir. Validation değeri ise tahmin edilen verilerdir. Burada öğrenme başarılı bir şekilde gerçekleşir.**

**Model 2**



**Model 1 den farklı olarak bu modelde gizli katman sayısında bir değişiklik yapmadan katmalardaki nöron sayısı azaltılmıştır.**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

|  |  |
| --- | --- |
|  |  |

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

MODEL 2

-------

Hidden Layer : 5

Learning Rate: 0.2

Epoch: 100

Batch size: 1024

r\_square: 0.9607297922693815

mean\_absolute\_error: 338.6249004944398

mean\_squared\_error: 631420.1314314632

explained\_variance\_score: 0.9608674340762136

**Model 3**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Bu modelde 2. Modeldeki katman sayıları ve nöron sayıları sabit tutulup epoch ve batch size değerleri arttırılmıştır.** Batch size’ın büyük olması, daha doğru gradyan değerinin hesaplanmasını sağlamaktadır. Bu durum da linerizasyonu azaltmaktadır. Batch size genelde 64ile 512 arasında 2'nin katı olan değerlerden belirleniyor. Derin öğrenmede problemi çözecek en uygun ağırlık değerleri adım adım hesaplandığı için ilk epoch’larda başarım düşük olacak, epoch sayısı arttıkça başarım da artacaktır.

|  |  |
| --- | --- |
|  |  |

MODEL 3

-------

Hidden Layer : 5

Learning Rate: 0.2

Epoch: 100

Batch size: 1024

r\_square: 0.9572492582735483

mean\_absolute\_error: 367.7858778487146

mean\_squared\_error: 687383.1466560334

explained\_variance\_score: 0.9599445928110879

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Model 4**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

|  |  |
| --- | --- |
|  |  |

MODEL 4

-------

Hidden Layer : 5

Learning Rate: 0.2

Epoch: 200

Batch size: 2048

r\_square: 0.9687369822625947

mean\_absolute\_error: 324.77792227517216

mean\_squared\_error: 502673.65286447125

explained\_variance\_score: 0.9691378122489991

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Burada epoch ve batch size arttırılmıştır ve öğrenmenin oranı daha da başarılı olmuştur**