Day-10

Drug Dataset ¶

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\drug.csv")
d
```

Out[2]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

In [3]:

```
d.columns
```

Out[3]:

```
Index(['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K', 'Drug'], dtype='objec
t')
```

In [4]:

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Age          200 non-null    int64  
 1   Sex          200 non-null    object 
 2   BP           200 non-null    object 
 3   Cholesterol  200 non-null    object 
 4   Na_to_K      200 non-null    float64
 5   Drug         200 non-null    object 
dtypes: float64(1), int64(1), object(4)
memory usage: 9.5+ KB
```

In [5]:

```
x=d[['Age']]
y=d['Na_to_K']
```

In [6]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [7]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[7]:

```
LinearRegression()
```

In [8]:

```
print(lr.intercept_)
```

```
17.927457525323412
```

In [9]:

```
print(lr.score(x_test,y_test))
```

```
-0.04315949889915682
```

In [10]:

```
print(lr.score(x_train,y_train))
```

```
0.0056800551360363105
```

Ridge Regression

In [11]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [12]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
```

Out[12]:

```
-0.04315723295982066
```

Lasso Regression

In [13]:

```
la=Lasso(alpha=10)
```

In [14]:

```
la.fit(x_train,y_train)
```

Out[14]:

```
Lasso(alpha=10)
```

In [15]:

```
la.score(x_test,y_test)
```

Out[15]:

```
-0.04159637006654493
```

Elastic Regreesion

In [16]:

```
from sklearn.linear_model import ElasticNet
```

In [17]:

```
en=ElasticNet()
```

In [18]:

```
en.fit(x_train,y_train)
```

Out[18]:

```
ElasticNet()
```

In [19]:

```
predict=(en.predict(x_test))
```

In [20]:

```
print(en.score(x_test,y_test))
```

```
-0.04269752321279241
```

Evaluation Method

In [21]:

```
from sklearn import metrics
```

In [22]:

```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,predict))
```

```
Mean Absolute Error: 5.027152474371551
```

In [23]:

```
print("Mean Square Error:",metrics.mean_squared_error(y_test,predict))
```

```
Mean Square Error: 37.591506449200196
```

In [24]:

```
print("Root Mean Square Error:",np.sqrt(metrics.mean_squared_error(y_test,predict)))
```

```
Root Mean Square Error: 6.1311912748829
```

In [ ]:

```

```