

# Day-9

## 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='object')
```

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 [8]:

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

In [9]:

```
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 [10]:

```
from sklearn.linear_model import LinearRegression

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

Out[10]:

```
LinearRegression()
```

In [11]:

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

```
18.628081994993995
```

In [12]:

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

```
-0.05279679876573873
```

In [13]:

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

```
0.012263571471906376
```

## Ridge Regression

In [15]:

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

In [16]:

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

Out[16]:

-0.05278461621654906

## Lasso Regression

In [18]:

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

In [19]:

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

Out[19]:

Lasso(alpha=10)

In [20]:

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

Out[20]:

-0.029104790614307596

In [ ]: