Machine Learning Basics

In this session I am practicing the models in Machine Learning using Python

Topics Covered :-

1. Linear Regression
2. Rigid and Lasso
3. Logistic Regression

#sklearn Linear Model Regression

Linear Regression,Rigid and Lasso

In [1]:
```
#house Pricing dataset Boston House
from sklearn.datasets import load_boston
```

In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

##importing all the required files
```

In [3]:
```
## assigning to df
df=load_boston()
```

In [4]:
```
## to show the dataset
df
```

Out[4]:
```
{'data': array([[6.3200e-03, 1.8000e+01, 2.3100e+00, ..., 1.5300e+01, 3.9690e
+02,
         4.9800e+00],
        [2.7310e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9690e+02,
         9.1400e+00],
        [2.7290e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9283e+02,
         4.0300e+00],
        ...,
        [6.0760e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
         5.6400e+00],
        [1.0959e-01, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9345e+02,
         6.4800e+00],
        [4.7410e-02, 0.0000e+00, 1.1930e+01, ..., 2.1000e+01, 3.9690e+02,
         7.8800e+00]]),
 'target': array([24. , 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, 18.9,
15. ,
        18.9, 21.7, 20.4, 18.2, 19.9, 23.1, 17.5, 20.2, 18.2, 13.6, 19.6,
        15.2, 14.5, 15.6, 13.9, 16.6, 14.8, 18.4, 21. , 12.7, 14.5, 13.2,
        13.1, 13.5, 18.9, 20. , 21. , 24.7, 30.8, 34.9, 26.6, 25.3, 24.7,
```

In [5]:
```
## framed dataset
pd.DataFrame(df.data)
```

Out[5]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33
...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1.0	273.0	21.0	391.99	9.67
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1.0	273.0	21.0	396.90	9.08
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1.0	273.0	21.0	396.90	5.64
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1.0	273.0	21.0	393.45	6.48
505	0.04741	0.0	11.93	0.0	0.573	6.030	80.8	2.5050	1.0	273.0	21.0	396.90	7.88

506 rows × 13 columns

In [6]:
```
dataset=pd.DataFrame(df.data)
dataset.columns=df.feature_names
print(dataset.head())
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0

   PTRATIO       B  LSTAT
0     15.3  396.90   4.98
1     17.8  396.90   9.14
2     17.8  392.83   4.03
3     18.7  394.63   2.94
4     18.7  396.90   5.33
```

In [7]:
```
## top 5 values
dataset.head()
```

Out[7]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

In [8]:
```
## targeting the price
dataset['price']=df.target
```

In [9]:
```
dataset.head()
```

Out[9]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

In [10]:
```
## Dividing the dataset into independent and dependent features

##Indexing remving the last column
X=dataset.iloc[:,:-1] ##independent feature

y=dataset.iloc[:,-1]##dependent feature
```

In [11]:
```
X.head()
```

Out[11]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

In [12]:
```
y.head()
```

Out[12]:
```
0    24.0
1    21.6
2    34.7
3    33.4
4    36.2
Name: price, dtype: float64
```

In [13]:
```
## Linear Regression
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
lin_reg=LinearRegression()
mse=cross_val_score(lin_reg,X,y,scoring='neg_mean_squared_error',cv=5) ##cro
print(mse)
```

```
[-12.46030057 -26.04862111 -33.07413798 -80.76237112 -33.31360656]
```

In [14]:
```
mean_mse=np.mean(mse)
print(mean_mse)
```

```
-37.13180746769922
```

lin_reg.predict(4) #to predict the value

In [15]:
```
## Ridge Regression
```

In [21]:
```
from sklearn.linear_model import Ridge
from sklearn.model_selection import GridSearchCV
ridge=Ridge()
params={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-3,1e-2,1,5,10,20]}

ridge_regressor=GridSearchCV(ridge,params,scoring='neg_mean_squared_error',c
ridge_regressor.fit(X,y)


```

Out[21]:
```
GridSearchCV(cv=5, estimator=Ridge(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.001, 0.001, 0.01, 1,
                                   5, 10, 20]},
             scoring='neg_mean_squared_error')
```

In [23]:
```
print(ridge_regressor.best_params_)
print(ridge_regressor.best_score_)
```

```
{'alpha': 20}
-32.38025025182513
```

In [24]:
```
from sklearn.linear_model import Lasso
from sklearn.model_selection import GridSearchCV
lasso=Lasso()
params={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-3,1e-2,1,5,10,20]}

lasso_regressor=GridSearchCV(lasso,params,scoring='neg_mean_squared_error',c
lasso_regressor.fit(X,y)


```

```
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 4430.746729651311, tolera
nce: 3.9191485420792076
  model = cd_fast.enet_coordinate_descent(
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 4397.459304778431, tolera
nce: 3.3071316790123455
  model = cd_fast.enet_coordinate_descent(
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 3796.653037433508, tolera
nce: 2.813643886419753
  model = cd_fast.enet_coordinate_descent(
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 2564.292735790545, tolera
nce: 3.3071762123456794
  model = cd_fast.enet_coordinate_descent(
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 4294.252997826028, tolera
nce: 3.4809104444444445
  model = cd_fast.enet_coordinate_descent(
```

Out[24]:
```
GridSearchCV(cv=5, estimator=Lasso(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.001, 0.001, 0.01, 1,
                                   5, 10, 20]},
             scoring='neg_mean_squared_error')
```

In [25]:
```
print(lasso_regressor.best_params_)
print(lasso_regressor.best_score_)
```

```
{'alpha': 1}
-35.531580220694856
```

In [26]:
```
from sklearn.linear_model import Ridge
from sklearn.model_selection import GridSearchCV
ridge=Ridge()
params={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-3,1e-2,1,5,10,20,30,40,45,50,55,10

ridge_regressor=GridSearchCV(ridge,params,scoring='neg_mean_squared_error',c
ridge_regressor.fit(X,y)


```

Out[26]:
```
GridSearchCV(cv=5, estimator=Ridge(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.001, 0.001, 0.01, 1,
                                   5, 10, 20, 30, 40, 45, 50, 55, 100]},
             scoring='neg_mean_squared_error')
```

In [27]:
```
from sklearn.linear_model import Lasso
from sklearn.model_selection import GridSearchCV
lasso=Lasso()
params={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-3,1e-2,1,5,10,20,30,40,45,50,55,10

lasso_regressor=GridSearchCV(lasso,params,scoring='neg_mean_squared_error',c
lasso_regressor.fit(X,y)


```

```
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 4430.746729651311, tolera
nce: 3.9191485420792076
  model = cd_fast.enet_coordinate_descent(
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 4397.459304778431, tolera
nce: 3.3071316790123455
  model = cd_fast.enet_coordinate_descent(
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 3796.653037433508, tolera
nce: 2.813643886419753
  model = cd_fast.enet_coordinate_descent(
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 2564.292735790545, tolera
nce: 3.3071762123456794
  model = cd_fast.enet_coordinate_descent(
C:\Users\arya shriva\anaconda3\lib\site-packages\sklearn\linear_model\_coordina
te_descent.py:530: ConvergenceWarning: Objective did not converge. You might wa
nt to increase the number of iterations. Duality gap: 4294.252997826028, tolera
nce: 3.4809104444444445
  model = cd_fast.enet_coordinate_descent(
```

Out[27]:
```
GridSearchCV(cv=5, estimator=Lasso(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.001, 0.001, 0.01, 1,
                                   5, 10, 20, 30, 40, 45, 50, 55, 100]},
             scoring='neg_mean_squared_error')
```

In [28]:
```
print(lasso_regressor.best_params_)
print(lasso_regressor.best_score_)
```

```
{'alpha': 1}
-35.531580220694856
```

In [30]:
```
from sklearn.model_selection import train_test_split
X_train, X_test,y_train,y_test = train_test_split(
X,y,test_size=0.33,random_state=42)
```

In [31]:
```
#ridge_regressor
from sklearn.linear_model import Ridge
from sklearn.model_selection import GridSearchCV
ridge=Ridge()
params={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-3,1e-2,1,5,10,20,30,40,45,50,55,10

ridge_regressor=GridSearchCV(ridge,params,scoring='neg_mean_squared_error',c
ridge_regressor.fit(X,y)
```

Out[31]:
```
GridSearchCV(cv=5, estimator=Ridge(),
             param_grid={'alpha': [1e-15, 1e-10, 1e-08, 0.001, 0.001, 0.01, 1,
                                   5, 10, 20, 30, 40, 45, 50, 55, 100]},
             scoring='neg_mean_squared_error')
```