

Experiment no:2

Aim: To predict housing prices in Boston using [Linear Regression](#) and Ridge regression taken the Boston Housing Dataset.

Program:

```
import numpy as np
import pandas as pd
import requests
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression , Ridge , Lasso
from io import StringIO

file_path = r"C:/Users/LENOVO/Downloads/housing.csv"
boston_pd = pd.read_csv(file_path, delim_whitespace=True, header=None)
boston_pd.columns = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO',
'B', 'LSTAT', 'House Price']

#boston_pd = pd.DataFrame(data)

#boston_pd.columns = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO',
'B', 'LSTAT']

#boston_pd['House Price'] = pd.Series(target)

X = boston_pd.iloc[:, :-1]
Y = boston_pd.iloc[:, -1]

print(boston_pd)

x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.25)
print("Train data shape of X=%s and Y=%s :"%(x_train.shape,y_train.shape))
```

```
print("Test data shape of X=%s and Y=%s :"%(x_test.shape,y_test.shape))
```

```
#apply multiple linear regression model
```

```
lreg = LinearRegression()
```

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

```
lreg_y_pred = lreg.predict(x_test)
```

```
mean_squared_error = np.mean((lreg_y_pred - y_test) ** 2)
```

```
print("Mean squared Error on test set:", mean_squared_error)
```

```
lreg_coefficient = pd.DataFrame()
```

```
lreg_coefficient["Columns"] = x_train.columns
```

```
lreg_coefficient["Coefficient Estimate"] = pd.Series(lreg.coef_)
```

```
print(lreg_coefficient)
```

```
fig, ax = plt.subplots(figsize=(20, 10))
```

```
color=['tab:gray', 'tab:blue', 'tab:orange',
```

```
'tab:green', 'tab:red', 'tab:purple', 'tab:brown',
```

```
'tab:pink', 'tab:gray', 'tab:olive', 'tab:cyan',
```

```
'tab:orange', 'tab:green', 'tab:blue', 'tab:olive']
```

```
ax.bar(lreg_coefficient["Columns"], lreg_coefficient['Coefficient Estimate'],color=color)
```

```
ax.spines["bottom"].set_position('zero')
```

```
plt.style.use('ggplot')
```

```
plt.xticks(rotation=45)
```

```
plt.show()
```

```
# Train the model
```

```
ridgeR=Ridge(alpha = 1)
```

```

ridgeR.fit(x_train, y_train)
y_pred=ridgeR.predict(x_test)
# calculate mean square error
mean_squared_error_ridge=np.mean((y_pred-y_test)**2)
print("Mean squared error on test set",mean_squared_error_ridge)

```

```

# get ridge coefficient and print them
ridge_coefficient=pd.DataFrame()
ridge_coefficient["Columns"]= x_train.columns
ridge_coefficient['Coefficient Estimate']=pd.Series(ridgeR.coef_)
print(ridge_coefficient)

```

```

fig, ax = plt.subplots(figsize=(20, 10))
color=['tab:gray', 'tab:blue', 'tab:orange',
'tab:green', 'tab:red', 'tab:purple', 'tab:brown',
'tab:pink', 'tab:gray', 'tab:olive', 'tab:cyan',
'tab:orange', 'tab:green', 'tab:blue', 'tab:olive']
ax.bar(ridge_coefficient["Columns"], ridge_coefficient['Coefficient Estimate'],color=color)
ax.spines["bottom"].set_position('zero')
plt.style.use('ggplot')
#plt.xticks(rotation=45)
plt.show()

```

```

#import Lasso regression from sklearn Library

```

```

lasso = Lasso(alpha = 1)
lasso.fit(x_train, y_train)
y_pred1 = lasso.predict(x_test)
# Calculate Mean Squared Error
mean_squared_error = np.mean((y_pred1 - y_test)**2)
print("Mean squared error on test set", mean_squared_error)

```

```
lasso_coeff=pd.DataFrame()  
lasso_coeff["Columns"]=x_train.columns  
lasso_coeff["Coefficient Estimate"] = pd.Series(lasso.coef_)  
print (lasso_coeff)
```