**LAB EXERCISE – 10**

**LASSO, RIDGE, ELASTIC NET**

**Aim of the Experiment**

To write python program for Regression variants like Lasso, Ridge and ElasticNet using a randomly generated dataset.

**Reference to Textbook and Explanation**

Chapter 5 and Appendix 2 for details about regression concept.

A random dataset is created as follows:

X,y = make\_regression(n\_samples = 50,n\_features=1,noise=0.1)

This command creates 50 samples with one feature and output.

The ridge model can be created as follows:

clf = RidgeCV(alphas=[1e-3, 1e-2, 1e-1, 1]).fit(X, y) and its scare can be displayed using this command,

clf.score(X, y)

Similarly, the lasso model can be created and its score can be printed as follows:

clf1 = linear\_model.Lasso(alpha=0.1).fit(X,y)

print(clf1.score(X,y))

Similarly, the ElasticNet model can be created and its score can be printed as follows:

clf2 =ElasticNet(random\_state=0).fit(X,y)

print(clf2.score(X,y))

**Program Listing**

from sklearn import linear\_model

from sklearn.linear\_model import RidgeCV

from sklearn.linear\_model import ElasticNet

from sklearn.datasets import make\_regression

# Create random dataset with 1 features. Dataset has 50 samples with noise 0.1

X,y = make\_regression(n\_samples = 50,n\_features=1,noise=0.1)

# linear regression

regr = linear\_model.LinearRegression()

regr.fit(X,y)

print('Intercept: \n', regr.intercept\_)

print('Coefficients:\n', regr.coef\_)

# find score

print('\nScore of Linear Regression:')

print(regr.score(X,y))

## RIDGE REGRESSION

clf = RidgeCV(alphas=[1e-3, 1e-2, 1e-1, 1]).fit(X, y)

clf.fit(X,y)

print('\nRidge Regression\n')

print('Ridge Intercept:', clf.intercept\_)

print('Ridge Coefficients:', clf.coef\_)

print('\nScore of Ridge Regression:')

print(clf.score(X,y))

# Compute Adjusted R squared Error

print("\nAdjusted R Squared for Ridge Regression mdel:",clf.score(X,y))

# LASSO REGRESSION

clf1 = linear\_model.Lasso(alpha=0.1).fit(X,y)

print('\nLasso Regression\n')

print('Lasso Intercept:', clf1.intercept\_)

print('Lasso Coefficients:', clf1.coef\_)

print('\nScore of Lasso Regression:')

print(clf1.score(X,y))

# Compute Adjusted R squared Error

print("\nAdjusted R Squared for Lasso Regression mdel:",clf1.score(X,y))

# ELASTICNET REGRESSION

clf2 =ElasticNet(random\_state=0).fit(X,y)

print('\nElasticNet Regression\n')

print('ElasticNet Intercept:', clf2.intercept\_)

print('ElasticNet Coefficients:', clf2.coef\_)

print('\nScore of ElasticNet Regression:')

print(clf2.score(X,y))

# Compute Adjusted R squared Error

print("\nAdjusted R Squared for ElsticNet Regression mdel:",clf2.score(X,y))

**Output**

Text

Description automatically generated

Text

Description automatically generated

It can be observed that only ElasticNet is performing worse as its score is less compared to the other methods.