CS353 ML Lab 5

Name: K V Sumanth Reddy

Roll No: 181CO225

Date: 02/03/2021

Q: Write a program in python to implement Multilinear regression for a sample data set.

Dataset Used: Breast Cancer

Importing Libraries and Dataset

```
from sklearn import linear_model
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer
import sklearn.metrics as metrics
```

Data Preprocessing

```
cancer = load_breast_cancer()
data = pd.DataFrame(cancer.data, columns=[cancer.feature_names])
data['Target'] = pd.Series(data=cancer.target, index=data.index)
featNames = cancer.feature_names
x,y = load_breast_cancer(return_X_y=True)

data.sample(10)
```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mea con poi
498	18.49	17.52	121.30	1068.0	0.10120	0.13170	0.14910	0.(
328	16.27	20.71	106.90	813.7	0.11690	0.13190	0.14780	0.(
554	12.88	28.92	82.50	514.3	0.08123	0.05824	0.06195	0.(
126	13.61	24.69	87.76	572.6	0.09258	0.07862	0.05285	0.(
420	11.57	19.04	74.20	409.7	0.08546	0.07722	0.05485	0.(
495	14.87	20.21	96.12	680.9	0.09587	0.08345	0.06824	0.(

Training the model

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

print('Training dataset size:\nx_train -', len(x_train), '\ny_train
print('Testing dataset size:\nx_test -', len(x_test), '\ny_test -',
```

```
Training dataset size:
x_train - 398
y_train - 398

Testing dataset size:
x_test - 171
y_test - 171
```

```
model = linear_model.LinearRegression()
model.fit(x_train,y_train)
y_pred = model.predict(x_test)
```

```
coeff = list(zip(model.coef_, featNames))

print('Y = %.3f'% model.intercept_, end = ' ')
for i in coeff:
  print('+', '({%.3f}'%i[0],'*{})'.format(i[1]), end = ' ')
```

```
Y = 3.026 + ({0.302} *mean radius) + ({-0.008} *mean texture) + ({-0.042} *me
```

Regression Evaluation Metrics

Here are three common evaluation metrics for regression problems:

Mean Absolute Error (MAE) is the mean of the absolute value of the errors:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Mean Squared Error (MSE) is the mean of the squared errors:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Accuracies

```
print("-------------------------------")
print('Mean Absolute Error:     %.3f'% metrics.mean_absolute_error(y
print('Mean Squared Error:      %.3f'% metrics.mean_squared_error(y_
print('Root Mean Squared Error: %.3f'% np.sqrt(metrics.mean_squared_
print("Variance score:          %.3f" % model.score(x_test, y_test))
print("-------------------------------")
```

```
-------------------------------
Mean Absolute Error:     0.188
Mean Squared Error:      0.062
Root Mean Squared Error: 0.249
Variance score:          0.743
-------------------------------
```