SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE: DJ19TEL7014 **DATE:22/09/23**

COURSE NAME: Machine Learning **CLASS:** Final Year B.Tech

EXPERIMENT NO. 2

CO Measured:

CO1 Solve real-world problems using suitable machine learning techniques.

TITLE: Model-building using regression

AIM / OBJECTIVE: To perform linear regression and find the error associated with the model.

DESCRIPTION OF EXPERIMENT:

Linear regression is one of the easiest and most popular Supervised Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc. Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.The linear regression model provides a sloped straight line representing the relationship between the variables.Cleaning Data in PythonWe will now separate the numeric columns from the categorical columns.

Mathematically, we can represent a linear regression as: $\mathbf{y= b_0+b_1x+ \varepsilon}$

Here, y= Dependent Variable (Target Variable) x= Independent Variable (predictor Variable) b_0= intercept of the line (Gives an additional degree of freedom) b_1 = Linear regression coefficient (scale factor to each input value).

ε = random error

The values for x and y variables are training datasets for Linear Regression model representation

The different values for weights or coefficient of lines (b_0, b_1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line. Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing. We can use the cost function to find the accuracy of the **mapping function**, which maps the input variable to the output variable. This mapping function is also known as **Hypothesis function**. For Linear Regression, we use the **Mean Squared Error (MSE)** cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(y_i - (b_1 x_i + b_0))^2$$

where,
N=Total number of observation y_i
= Actual value
$(b_1x_i+b_0)$= Predicted value.

Linear regression using Least Square Method

We have linear regression equation as $\mathbf{y= b_0+b_1x}$

Using least square method,

$$\boldsymbol{b_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}}$$

$$\boldsymbol{b_0 = \bar{y} - b_1\bar{x}}$$

PROCEDURE:

1. Describe the procedure that is used to perform Linear regression using Least Square Method carry out the experiment step-by-step for simple linear regression for following dataset without using scikit library. Describe every line of code with the proper interpretation of the output.

X	2	3	4	5	6	7	8	9	10
Y	1	3	6	9	11	13	15	17	20

2. Perform Regression with respect to one dataset of your choice and discuss results of all the steps.

OBSERVATIONS:

exp2

September 22, 2023

[1]:
```
#part 1
```

[2]:
```
X=[2,3,4,5,6,7,8,9,10]
Y=[1,3,6,9,11,13,15,17,20]
```

[4]:
```
def eqOfLine(m,c,x):
    return m*x+c
```

[11]:
```
def fancyPrint(x,y,xy,xsquar):
    print('\tX\tY\tXY\tX^2')
    for i in range(len(x)):
        print('\t',x[i],'\t',y[i],'\t',xy[i],'\t',xsquar[i])
    print('sum :\t',sum(x),'\t',sum(y),'\t',sum(xy),'\t',sum(xsquar))
```

[22]:
```
def leastSquareReg(X,Y):
    n=len(X)
    xsquar=[x*x for x in X]
    xy=[X[i]*Y[i] for i in range(n)]
    slope=(n*sum(xy)-sum(X)*sum(Y))/(n*sum(xsquar)-sum(X)**2)
    intercept=sum(Y)/n-slope*sum(X)/n
    fancyPrint(X,Y,xy,xsquar)
    print('\nslope     :',slope)
    print('intercept :',intercept)
    return slope,intercept
```

[23]:
```
m,c=leastSquareReg(X,Y)
```

```
        X       Y       XY      X^2
         2       1       2       4
         3       3       9       9
         4       6       24      16
         5       9       45      25
         6       11      66      36
         7       13      91      49
         8       15      120     64
         9       17      153     81
         10      20      200     100
sum :    54      95      710     384
```

```
slope     : 2.3333333333333335
intercept : -3.4444444444444464
```

[26]:
```
#prediction with custom model
y=eqOfLine(m,c,4)
print('predicted y = ',y)
print('actual y    = ',6)
```

```
predicted y =  5.8888888888888875
actual y    =  6
```

[27]:
```
#root mean squared error for custom model
rmse=(((y-6)**2)/1)**0.5
print('rmse = ',rmse)
```

```
rmse =  0.11111111111111249
```

[33]:
```
#part 2
#dataset : https://www.kaggle.com/datasets/mdrazakhan/linear-regression-dataset
```

[35]:
```
import pandas as pd
from sklearn import linear_model
import matplotlib.pyplot as plt
```

[36]:
```
df=pd.read_csv('dataset\cars.csv')
```

[38]:
```
df.head()
```

[38]:

	Car	Model	Volume	Weight	CO2
0	Toyoty	Aygo	1000	790	99
1	Mitsubishi	Space Star	1200	1160	95
2	Skoda	Citigo	1000	929	95
3	Fiat	500	900	865	90
4	Mini	Cooper	1500	1140	105

[43]:
```
x=df[['Weight']]
y=df['CO2']
```

[45]:
```
regrLine=linear_model.LinearRegression()
regrLine.fit(x, y)
```

[45]: LinearRegression()

[54]:
```
# Coefficient
print("Coefficient : ",end=' ')
print(regrLine.coef_)
```

```
Coefficient :  [0.01699973]
```

[55]:
```
predictedCO2 = regrLine.predict([[990]])
print("predicted CO2 for weight=990kg :",end=' ')
print(predictedCO2)
```

```
predicted CO2 for weight=1000kg : [96.88913578]
```

```
c:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\base.py:465: UserWarning: X does not have valid feature names,
but LinearRegression was fitted with feature names
  warnings.warn(
```

[59]:
```
print("actual CO2 for weight=990kg :")
print(df.iloc[11])
```

```
actual CO2 for weight=990kg :
Car        Suzuki
Model       Swift
Volume       1300
Weight        990
CO2           101
Name: 11, dtype: object
```

CONCLUSION:

Part 1: Simple Linear Regression using Least Squares Method

In the first part of the experiment, we performed simple linear regression using the Least Squares Method. This involved fitting a linear model without utilizing any external libraries like scikit-learn.

1. **Data Preparation:** We started with a dataset containing two variables, X and Y, with corresponding values. X represents the independent variable, and Y represents the dependent variable.
2. **Model Definition:** We defined the linear regression model, which aimed to find a linear relationship between X and Y. The model was represented as Y = m * X + c, where m is the slope and c is the intercept.
3. **Calculating Model Parameters:** Using the Least Squares Method, we calculated the values of m and c that minimize the sum of squared differences between the observed Y values and the values predicted by the model.

Part 2: Regression Analysis on a Chosen Dataset

In the second part of the experiment, we applied the regression methodology using external libraries ie scikit-learn to a cars. This part allowed us to showcase the applicability of linear regression in a real-world context and provided insights into the relationship between the chosen independent and dependent variables.

DATASET:

[1] https://www.kaggle.com/datasets/mdrazakhan/linear-regression-dataset