Roll No. 22DEC	**Experiment No.** 06	**Marks :**
BATCH - C		**Sign :**

Aim: Implementation of prediction algorithm (Linear regression)

Apparatus: Google Colab

Theory:

What is Linear Regression?

Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features. When the number of the independent feature is 1 then it is known as Univariate Linear regression, and in the case of more than one feature, it is known as multivariate linear regression.

Why is Linear Regression Important?

The interpretability of linear regression is a notable strength. The model's equation provides clear coefficients that elucidate the impact of each independent variable on the dependent variable, facilitating a deeper understanding of the underlying dynamics. Its simplicity is a virtue, as linear regression is transparent, easy to implement, and serves as a foundational concept for more complex algorithms.

Linear regression is not merely a predictive tool; it forms the basis for various advanced models. Techniques like regularization and support vector machines draw inspiration from linear regression, expanding its utility. Additionally, linear regression is a cornerstone in assumption testing, enabling researchers to validate key assumptions about the data.

Types of Linear Regression:

There are two main types of linear regression:

1. **Simple Linear Regression**
   This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear regression is:
   $y = \beta_0 + \beta_1 X$

where:

Y is the dependent variable
X is the independent variable
β0 is the intercept
β1 is the slope

2. **Multiple Linear Regression**

   This involves more than one independent variable and one dependent variable. The equation for multiple linear regression is:

$$y = \beta_0 + \beta_1 X + \beta_2 X + ...\beta_n X$$

   y=\beta_{0}+\beta_{1}X+\beta_{2}X+.........\beta_{n}X
   where:

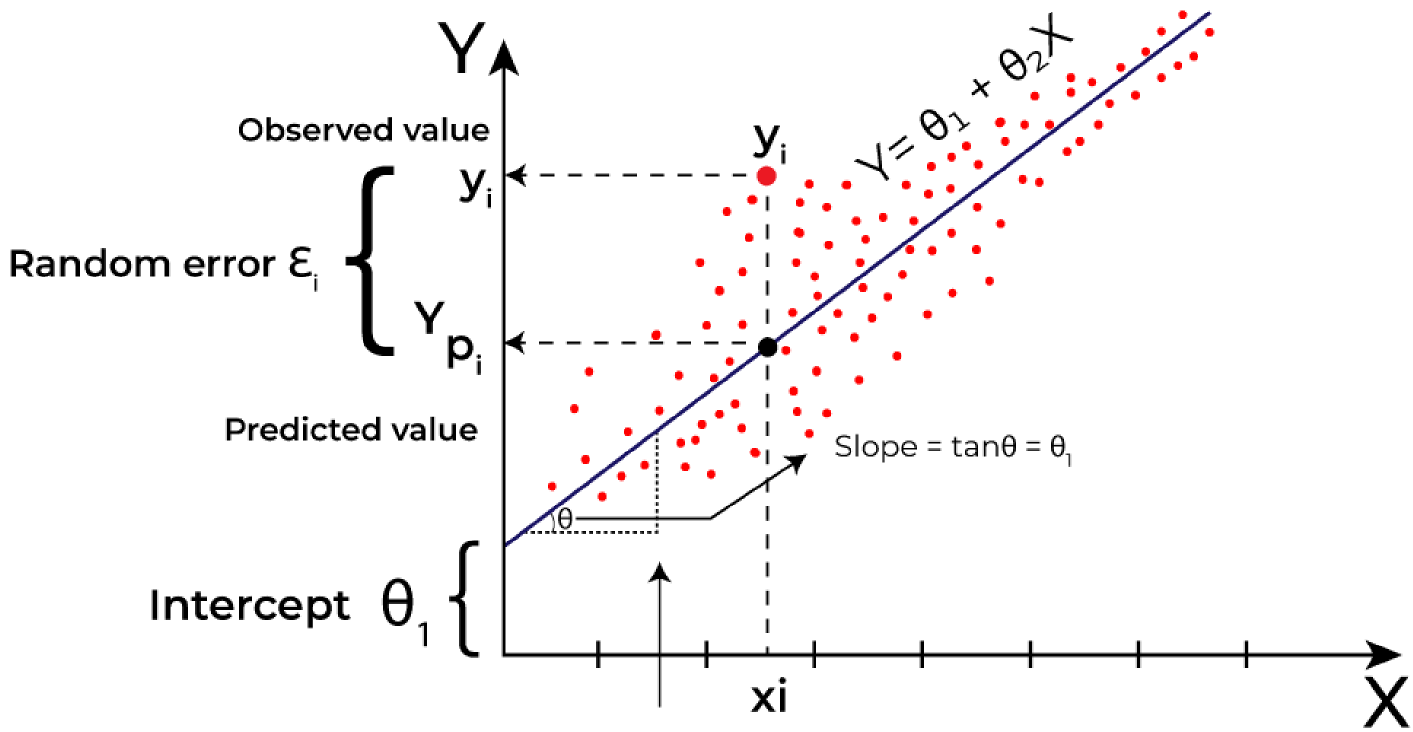
   Y is the dependent variable
   X1, X2, …, Xp are the independent variables
   β0 is the intercept
   β1, β2, …, βn are the slopes

The goal of the algorithm is to find the best Fit Line equation that can predict the values based on the independent variables.
In regression a set of records are present with X and Y values and these values are used to learn a function so if you want to predict Y from an unknown X this learned function can be used. In regression we have to find the value of Y, So, a function is required that predicts continuous Y in the case of regression given X as independent features.

What is the best Fit Line?

Our primary objective while using linear regression is to locate the best-fit line, which implies that the error between the predicted and actual values should be kept to a minimum. There will be the least error in the best-fit line.

The best Fit Line equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable(s).



Here Y is called a dependent or target variable and X is called an independent variable also known as the predictor of Y. There are many types of functions or modules that can be used for regression. A linear function is the simplest type of function. Here, X may be a single feature or multiple features representing the problem.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x)). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best-fit line for our model.

We utilize the cost function to compute the best values in order to get the best fit line since different values for weights or the coefficient of lines result in different regression lines.

Implementation:

Prediction Algorithm (Linear Regression):

```
data01 = pd.read_csv("/content/drive/MyDrive/Deep_learning_workshop/SALES_2_.csv")

data01.columns = ["price","value"]

x = data01['price'].to_numpy()
y = data01['value'].to_numpy()
x = x.reshape(-1,1)
y = y.reshape(-1,1)

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)

from sklearn.linear_model import LinearRegression
lm = LinearRegression()

lm.fit(X_train,y_train)

y_pred=lm.predict(X_test)


from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

plt.scatter(x, y, color = 'blue', label='Scatter Plot')
plt.plot(X_test, y_pred, color = 'black', linewidth=3, label = 'Regression Line')
plt.title('Relationship between Price and Value')
plt.xlabel('Price')
plt.ylabel('Value')
plt.legend(loc=4)
plt.show()
```

OUTPUT:

Conclusion:

Implementing the Apriori algorithm in Python facilitates efficient mining of association rules from transactional data. Through preprocessing, algorithm execution, and interpretation, valuable insights about item relationships are gained. These insights aid in optimizing product recommendations, market basket analysis, and understanding consumer behavior for informed business decisions.