EXPERIMENT NO 2

Aim: To write the implementation of linear regression in Python/R.

Objective: To understand the use of simple linear regression techniques by implementing user define dataset and importing dataset

Description:

Regression analysis is a very widely used statistical tool to establish a relationship model between two variables. One of these variables is called a predictor variable whose value is gathered through experiments. The other variable is called response variable whose value is derived from the predictor variable.

In Linear Regression these two variables are related through an equation, where the exponent (power) of both these variables is 1. Mathematically a linear relationship represents a straight line when plotted as a graph. A non-linear relationship where the exponent of any variable is not equal to 1 creates a curve.

The general mathematical equation for a linear regression is –

$$y = ax + b$$

Following is the description of the parameters used –

- **y** is the response variable.
- **x** is the predictor variable.
- **a** and **b** are constants which are called the coefficients.

Procedure:

1. **Understanding the Problem:** The first step in building a linear regression model is to understand the problem that needs to be solved. It is important to clearly define the dependent and independent variables and their relationship to each other.
2. **Data Collection:** The next step is to collect the data that will be used to build the model. This involves identifying the variables that will be used and gathering the necessary data from various sources. It is important to ensure that the data is accurate and complete and that any missing values are addressed.
3. **Data Preparation:** Once the data has been collected, it must be prepared for analysis. This involves cleaning the data to remove any errors or inconsistencies and transforming the data to ensure that it is in a suitable format for analysis. This may involve removing outliers, dealing with missing values, and normalizing the data.
4. **Model Selection:** The next step is to select the appropriate model for the analysis. This involves choosing the type of regression to be used, such as simple or multiple linear regression, and determining the appropriate degree of complexity for the model.

5. **Model Building:** Once the model has been selected, it is time to build the model. This involves using the data to estimate the coefficients of the model and calculating the goodness of fit of the model. The model can be refined by adding or removing variables and adjusting the degree of complexity as needed.
6. **Model Evaluation:** The final step is to evaluate the performance of the model. This involves assessing the accuracy and reliability of the model and determining whether it is suitable for the intended purpose. This may involve using techniques such as cross-validation or comparing the model to alternative models.

Code:

```
import numpy as np

class LinearRegression:

    def __init__ (self):

        self.b_0 = 0

        self.b_1 = 0


    def fit (self, X, y):

        X_mean = np.mean (X)

        y_mean = np.mean (y)

        ssxy, ssx = 0, 0

        for _ in range (len (X)):

            ssxy += (X[_]-X_mean)*(y[_]-y_mean)

            ssx += (X[_]-X_mean)**2


        self.b_1 = ssxy / ssx

        self.b_0 = y_mean - (self.b_1*X_mean)

        return self.b_0, self.b_1


    def predict (self, X):
```

```
        y_hat = self.b_0 + (X * self.b_1)

        return y_hat


if __name__ == '__main__':

    X = np.array ([173, 182, 165, 154, 170], ndmin=2)

    X = X.reshape(5, 1)

    y = np.array ([68, 79, 65, 57, 64])

    model = LinearRegression ()

    model.fit (X, y)

    y_pred = model.predict ([161])

    print (y_pred)
```

Output:

```
[60.85051546]
```

Conclusion:

1. Function used for linear regression in Python/R is Im() Function (Function_name (Parameters))

2. Explain use of numpy library.
   NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.