Experiment No - 3

Aim:- Implement Multiple Linear Regression using R/Python programming.

Objective:- To understand the use of Multiple linear regression techniques by implementing a predefined dataset of R Studio.

Description:-

Multiple linear regression is the extension of linear regression in the relationship between more than two variables. In simple linear regression, we have one predictor and one response variable. But in multiple regressions, we have more than one predictor variable and one response variable.

There is the following general mathematical equation for multiple regression -

$$y = b_0 + b_1*x_1 + b_2*x_2 + b_3*x_3 + \cdots b_n*x_n$$

Here,

- y is a response variable.

- **b0, b1, b2...bn** are the coefficients.

- **x1, x2, ...xn** are the predictor variables.

Program:-

```
import numpy as np

class MultipleRegression:
    def __init__ (self):
        self.params = np.zeros(int(np.random.random()), float)[:,np.newaxis]

    def fit (self, X, y):
        bias = np.ones (len (X))
        X_bias = np.c_[bias, X]
        inner_part = np.transpose (X_bias) @ X_bias
        inverse_part = np.linalg.inv (inner_part)
        outer_part = inverse_part @ np.transpose (X_bias)
        lse = outer_part @ y
        self.params = lse
        return self.params
```

```python
    def predict (self, X):
        bias_testing = np.ones (len (X))
        X_test = np.c_[bias_testing, X]
        y_hat = X_test @ self.params
        return y_hat

if __name__ == '__main__':

    X = np.array ([
        [1, 4],
        [2, 5],
        [3, 8],
        [4, 2]
    ])

    y = np.array ([1, 6, 8, 12])

    model = MultipleRegression ()
    parameters = model.fit (X, y)
    print (f'The parameters are {parameters}')
    y_pred = model.predict ([[5, 3]])
    print (f'The predicted value is {y_pred}')
```

Output:-

```
The parameters are [-1.69945355  3.48360656 -0.05464481]
The predicted value is [15.55464481]
```

Conclusion:-

1. Equation for multiple linear regression is $y = b_0 + b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + \cdots b_n * x_n$
2. When there is only one dependent variable and multiple independent variable then this types of regression is known as Ordinary least-squares(OLS) Regression
3. How to check inbuilt dataset in R/Python programming?
   -In R, you can list available datasets using the **'data()'** function, and then access individual datasets with their names, like **'data("mtcars")'**. In Python, many datasets are available through libraries like scikit-learn or seaborn, and you can typically find information on accessing them in the respective library documentation.