Implementation-of-Linear-Regression-Using-Gradient-Descent

AIM:

To write a program to predict the profit of a city using the linear regression model with gradient descent.

Equipments Required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Jupyter notebook

Algorithm

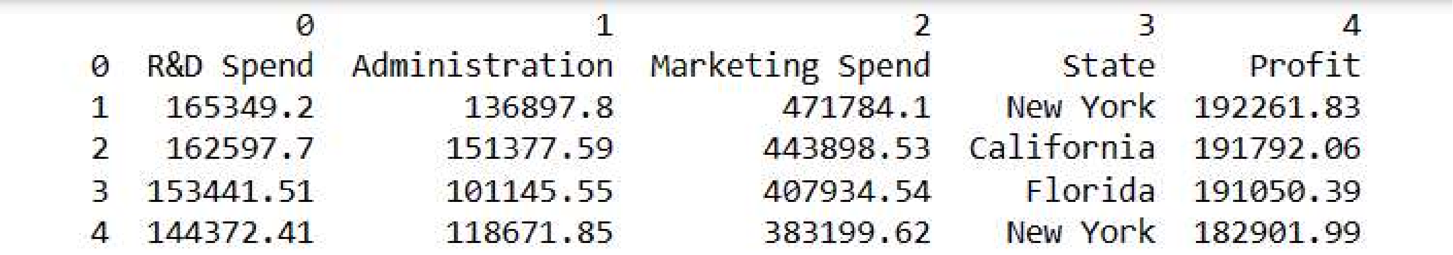
1. Import the required library and read the dataframe.
2. Write a function computeCost to generate the cost function.
3. Perform iterations og gradient steps with learning rate.
4. Plot the Cost function using Gradient Descent and generate the required graph.

Program:

```
/*
Program to implement the linear regression using gradient descent.
Developed by: Dharshni V M
RegisterNumber: 212223240029

import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
def linear_regression(X1, y, learning_rate=0.01, num_iters=1000):
  #Add a column of ones to X for the intercept term
 X = np.c_[np.ones(len(X1)),X1]

 # Initialize theta with zeros
 theta = np.zeros(X.shape[1]).reshape(-1,1)

 # Perform gradiant descent
 for _ in range(num_iters):
  #Calculate predictions
  predictions = (X).dot(theta).reshape(-1,1)

  #Calculate errors
  errors = (predictions - y).reshape(-1,1)

  #update theta using gradiant descent
  theta -= learning_rate * (1/len(X1)) * X.T.dot(errors)

  return theta
data = pd.read_csv('50_Startups.csv',header=None)
print(data.head())

#Assuming the last column is your target variable 'y' and the preceding column
X = (data.iloc[1:, :-2].values)
print(X)
X1=X.astype(float)
scaler = StandardScaler()
y = (data.iloc[1:,-1].values).reshape(-1,1)
print(y)
X1_Scaled = scaler.fit_transform(X1)
Y1_Scaled = scaler.fit_transform(y)
print(X1_Scaled)
print(Y1_Scaled)

# Learn model parameters
theta = linear_regression(X1_Scaled, Y1_Scaled)

# Predict target value for a new data point
new_data = np.array([165349.2,136897.8,471784.1]).reshape(-1,1)
new_Scaled = scaler.fit_transform(new_data)
```

```
prediction = np.dot(np.append(1, new_Scaled), theta)
prediction = prediction.reshape(-1,1)
pre=scaler.inverse_transform(prediction)
print(f"Predicted value: {pre}")

*/
```

Output:



```
           0               1               2           3          4
0  R&D Spend  Administration  Marketing Spend       State     Profit
1   165349.2        136897.8        471784.1    New York  192261.83
2   162597.7       151377.59       443898.53  California  191792.06
3  153441.51       101145.55       407934.54     Florida  191050.39
4  144372.41       118671.85       383199.62    New York  182901.99
```

Result:

Thus the program to implement the linear regression using gradient descent is written and verified using python programming.