EXPERIMENT-2

AIM:

Designing ML model from scratch- Linear Regression.

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
salary_df = pd.read_csv('Salary_data.csv')
```

In [4]:
```
salary_df.head()
```

Out[4]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

In [5]:
```
class LinearRegression:
    def __init__(self):
        self.m = 0
        self.c  =0

    def train(self, X, Y):
        X = np.array(X);
        Y = np.array(Y);
        x_mean = np.mean(X);
        y_mean = np.mean(Y);
        xy_mean = np.mean(X * Y);
        x2_mean = np.mean(X ** 2);

        self.m = (xy_mean - (x_mean * y_mean))/ (x2_mean - x_mean ** 2)

        self.c = y_mean - self.m * x_mean

    def ptinf_weight(self):
        print(f'{self.m} &&  {self.c}');

    def predict(self, X):
        return np.array(X) * self.m + self.c

    def mse(self, X, Y):
        y_pred = np.array(X) * self.m + self.c;
        return np.sum((np.array(Y) - y_pred) ** 2)/Y.shape[0];

    def rmse(self, X, Y):
        y_pred = np.array(X) * self.m + self.c;
```

```
        return np.sqrt(np.sum((np.array(Y) - y_pred) ** 2)/Y.shape[0]);
```

In [6]:
```
model = LinearRegression()
model.train(salary_df['YearsExperience'], salary_df['Salary'])
model.ptinf_weight()
```

```
9449.962321455081 &&  25792.200198668666
```

In [7]:
```
model.predict(salary_df['YearsExperience'])
```

Out[7]:
```
array([ 36187.15875227,  38077.15121656,  39967.14368085,  44692.12484158,
        46582.11730587,  53197.09093089,  54142.08716303,  56032.07962732,
        56032.07962732,  60757.06078805,  62647.05325234,  63592.04948449,
        63592.04948449,  64537.04571663,  68317.03064522,  72097.0155738 ,
        73987.00803809,  75877.00050238,  81546.97789525,  82491.9741274 ,
        90051.94398456,  92886.932681  , 100446.90253816, 103281.8912346 ,
       108006.87239533, 110841.86109176, 115566.84225249, 116511.83848464,
       123126.81210966, 125016.80457395])
```

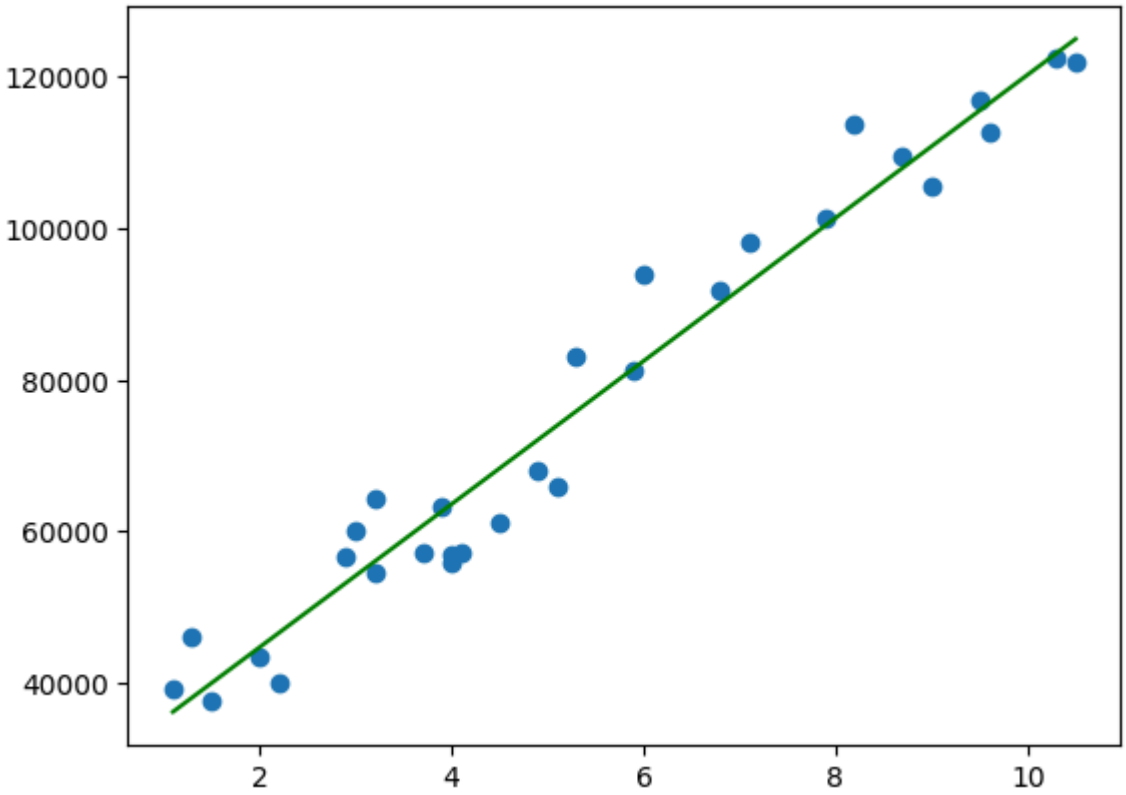
In [8]:
```
salary_df.head()
```

Out[8]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

In [9]:
```
plt.scatter(salary_df['YearsExperience'], salary_df['Salary'])
plt.plot(salary_df['YearsExperience'],model.predict(salary_df['YearsExperience']
```

Out[9]:
```
[<matplotlib.lines.Line2D at 0x2053af7acf0>]
```



In [10]:
```
model.mse(salary_df['YearsExperience'], salary_df['Salary'])
```

Out[10]:
```
np.float64(31270951.722280957)
```

In [11]:
```
model.rmse(salary_df['YearsExperience'], salary_df['Salary'])
```

Out[11]:
```
np.float64(5592.043608760661)
```

Gradient Decent

In [12]:
```
def grad_descent(x,y,learning_rate=0.0001,epochs=100):
    m = 0
    c = 0
    n = len(x)
    for i in range(100):
        y_pred = m*x + c
        dm = (-2/n) * np.sum(x*(y-y_pred))
        dc = (-2/n) * np.sum(y-y_pred)
        m = m - learning_rate*dm
        c = c - learning_rate*dc
    return m,c
```

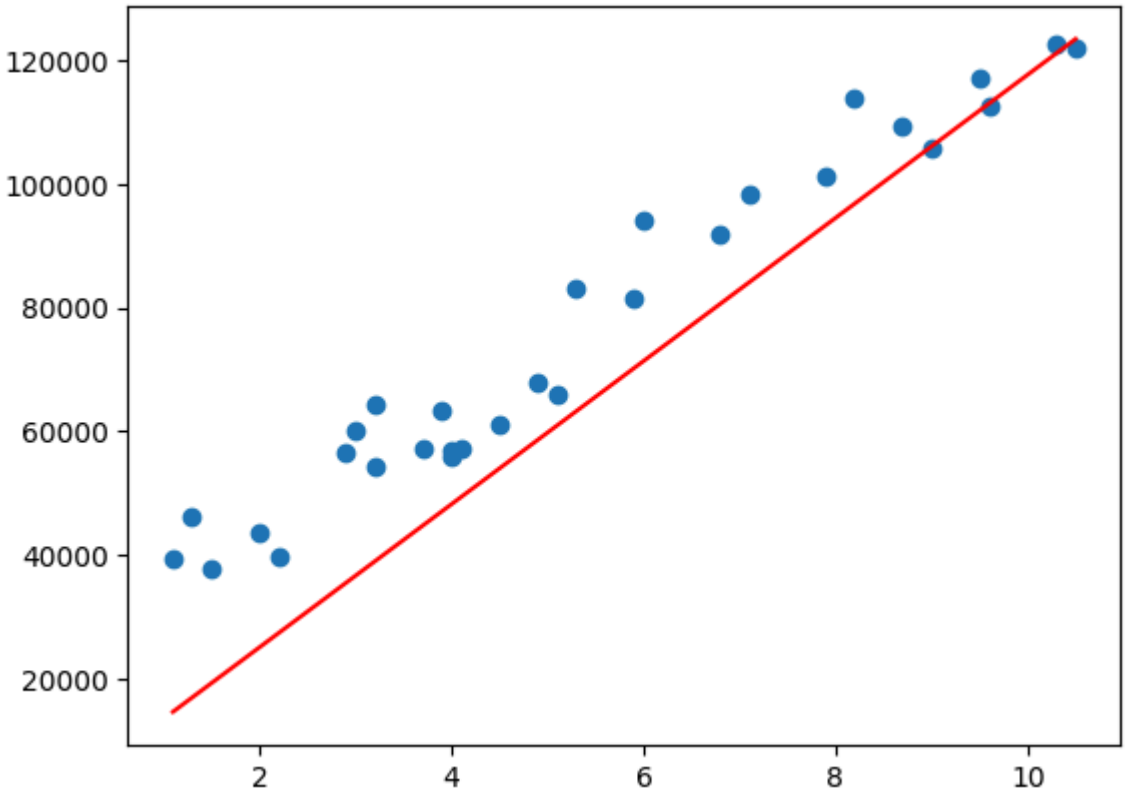
In [13]:
```
m, c = grad_descent(salary_df['YearsExperience'], salary_df['Salary'], learning_
```

In [ ]:
```

```

In [14]:
```
plt.scatter(salary_df['YearsExperience'], salary_df['Salary'])
plt.plot(salary_df['YearsExperience'], m*salary_df['YearsExperience']+c, 'r-')
```

Out[14]:
```
[<matplotlib.lines.Line2D at 0x2053d24d580>]
```



In [ ]: