```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

import sklearn.metrics as sm

from sklearn.linear_model import LinearRegression

from sklearn.model_selection import train_test_split

df = pd.read_csv("archive.zip")

print(df)

df.describe()

X = df['YearsExperience'].values.reshape(-1,1)

Y = df['Salary'].values

print(X)
print(Y)

X_train, X_test, Y_train, Y_test =  train_test_split(X, Y, test_size = 0.1, random_state = 1)

model =  LinearRegression()

model.fit(X, Y)

Y_pred = model.predict(X_test)

Y_predl = model.predict(X.reshape (-1,1))

print(Y_pred)

print("Mean absolute error =", round(sm.mean_absolute_error(Y_test, Y_pred), 2))
print("Mean squared error =", round(sm.mean_squared_error(Y_test, Y_pred), 2))
print("Median absolute error =", round(sm.median_absolute_error(Y_test, Y_pred), 2))
print("Explain variance acore=", round(sm.explained_variance_score(Y_test, Y_pred), 2))
print("R2 score", round(sm.r2_score (Y_test, Y_pred), 2))

plt.scatter (X_train, Y_train,color='g')

plt.plot(X, Y_predl, color='k')

plt.show()
```

```
    Unnamed: 0  YearsExperience    Salary
0            0              1.2   39344.0
1            1              1.4   46206.0
2            2              1.6   37732.0
3            3              2.1   43526.0
4            4              2.3   39892.0
5            5              3.0   56643.0
6            6              3.1   60151.0
7            7              3.3   54446.0
8            8              3.3   64446.0
9            9              3.8   57190.0
10          10              4.0   63219.0
11          11              4.1   55795.0
12          12              4.1   56958.0
13          13              4.2   57082.0
14          14              4.6   61112.0
15          15              5.0   67939.0
16          16              5.2   66030.0
17          17              5.4   83089.0
18          18              6.0   81364.0
19          19              6.1   93941.0
20          20              6.9   91739.0
21          21              7.2   98274.0
22          22              8.0  101303.0
23          23              8.3  113813.0
24          24              8.8  109432.0
25          25              9.1  105583.0
26          26              9.6  116970.0
27          27              9.7  112636.0
28          28             10.4  122392.0
29          29             10.6  121873.0
[[ 1.2]
 [ 1.4]
 [ 1.6]
 [ 2.1]
 [ 2.3]
 [ 3. ]
 [ 3.1]
 [ 3.3]
 [ 3.3]
 [ 3.8]
 [ 4. ]
 [ 4.1]
 [ 4.1]
 [ 4.2]
 [ 4.6]
 [ 5. ]
 [ 5.2]
 [ 5.4]
 [ 6. ]
 [ 6.1]
 [ 6.9]
 [ 7.2]
 [ 8. ]
 [ 8.3]
 [ 8.8]
 [ 9.1]
 [ 9.6]
 [ 9.7]
 [10.4]
 [10.6]]
[ 39344.  46206.  37732.  43526.  39892.  56643.  60151.  54446.  64446.
  57190.  63219.  55795.  56958.  57082.  61112.  67939.  66030.  83089.
```