link= "C:\\Users\\DELL\\Documents\\student\_scores.csv"  
import pandas as pd  
df= pd.read\_csv(link)  
print(df)  
# EDA= Exploratory data analytics  
import matplotlib.pyplot as plt  
plt.scatter(df['Hours'],df['Scores'])  
plt.show()  
  
X= df.iloc[:,:1].values  
Y= df.iloc[:,1].values  
  
from sklearn.model\_selection import train\_test\_split  
X\_train, X\_test, Y\_train, Y\_test= train\_test\_split(X,Y,test\_size= 0.25, random\_state=100)  
  
### Run the model  
# Simple linear regression  
from sklearn.linear\_model import LinearRegression  
regressor= LinearRegression()  
regressor.fit(X\_train, Y\_train)  
# m=slope, c=cofficient  
print("Coefficient=", regressor.coef\_)  
print("Intercept=", regressor.intercept\_)  
'''  
Coefficient= [9.81094418]  
Intercept= 5.616340449521917  
y= 9.8X + 5.6C  
10 Hours= 9.8 marks  
'''  
  
  
# testing the model with test data  
y\_pred= regressor.predict(X\_test)  
result=pd.DataFrame({'Actual':Y\_test, 'Predicted':y\_pred})  
print(result)  
  
# Evaluation  
from sklearn import metrics  
mae= metrics. mean\_absolute\_error(Y\_test, y\_pred)  
print("Mean Absolute Error=", mae)  
mse= metrics.mean\_squared\_error(Y\_test, y\_pred)  
print('mean Square error=', mse)  
rmse=mse\*\*0.5  
print("Root mean squared error", rmse)  
r2= metrics.r2\_score(Y\_test, y\_pred)  
print("R squared value= ", r2)  
  
## R square- compare how well your model has performed if you had taken average, how well regression performed as per the averag  
'''  
R square= 1- Regression Error/Average Error  
R square for Best fit model= 1-0/some error=1  
R square for worse case= 1- avgerror/avgerror=0  
'''  
  
## OLS  
from statsmodels.api import OLS  
import statsmodels.api as sm  
import numpy as np  
X=np.array(X,dtype=float)  
X\_sm=sm.add\_constant (X)  
summ=OLS(Y,X\_sm).fit().summary()  
print("summary from OLS with 5 variables", summ)