**MACHINE LEARNING PROJECT**

In this project, I have used Linear Regression Machine Learning model for the Salary dataset from Kaggle website

**PROBLEM STATEMENT:**

This is a simple data where i tried to explain simple linear regression in a simplest way.For the beginner who wants to start their Machine Learning or data science can follow this simple data to understand simple linear regression.

This data consists of salary and years of experience of 35 jobholders.When i will try to show the relationship between salary and years of experience.

**ML METHODOLOGY:**

Linear Regression is the methodology used for training and testing the dataset.

Linear Regression is a method of modelling a target value based on independent predictors.This method is mostly used for forecasting and finding out cause and effect relationship between variables.Linear Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.

**DATASET DESCRIPTION:**

Some relevant columns in the dataset

* Years and experiences
* Salary

**PRE\_PROCESSING:**

Pre\_processing refers to the transformations applied to our data before feeding it to the algorithm.

1)%matplotlib inline

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

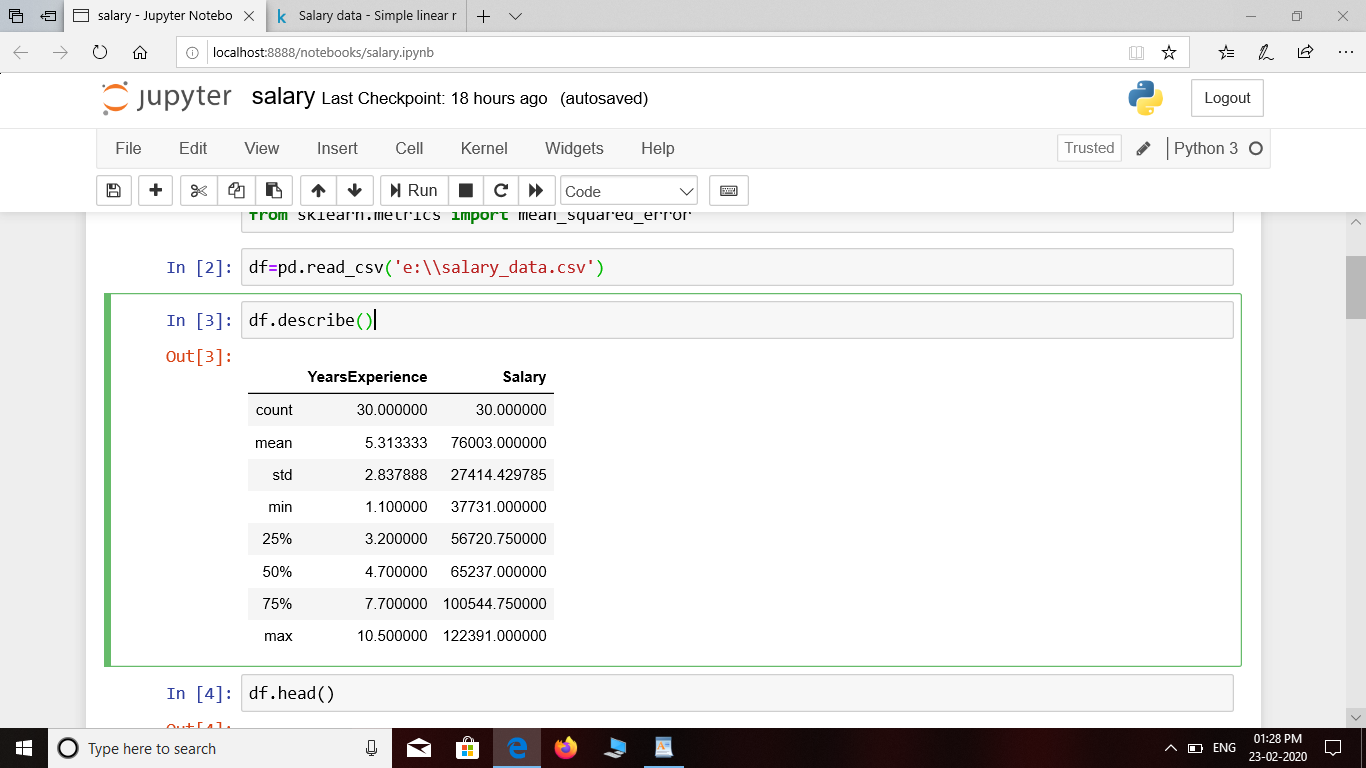
from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

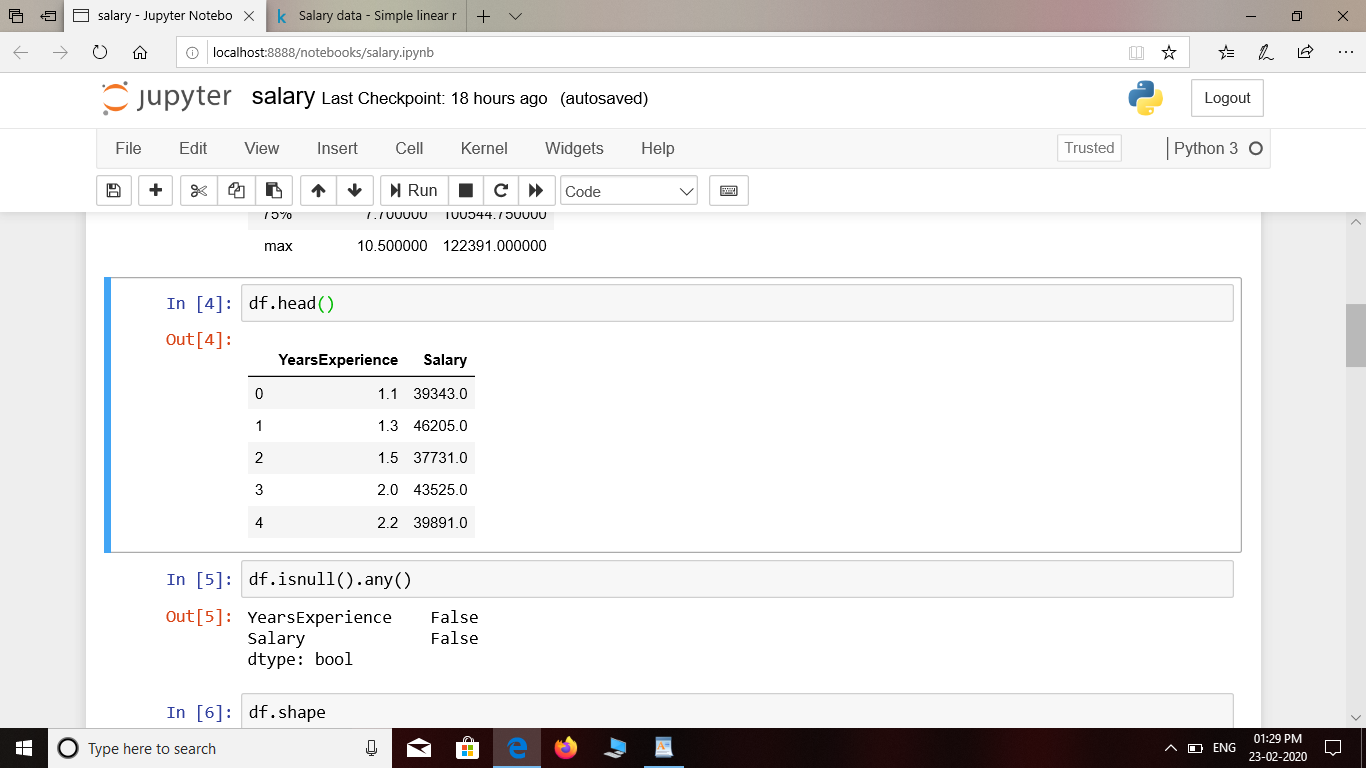
from sklearn.metrics import mean\_squared\_error

df=pd.read\_csv('e:\\salary\_data.csv')

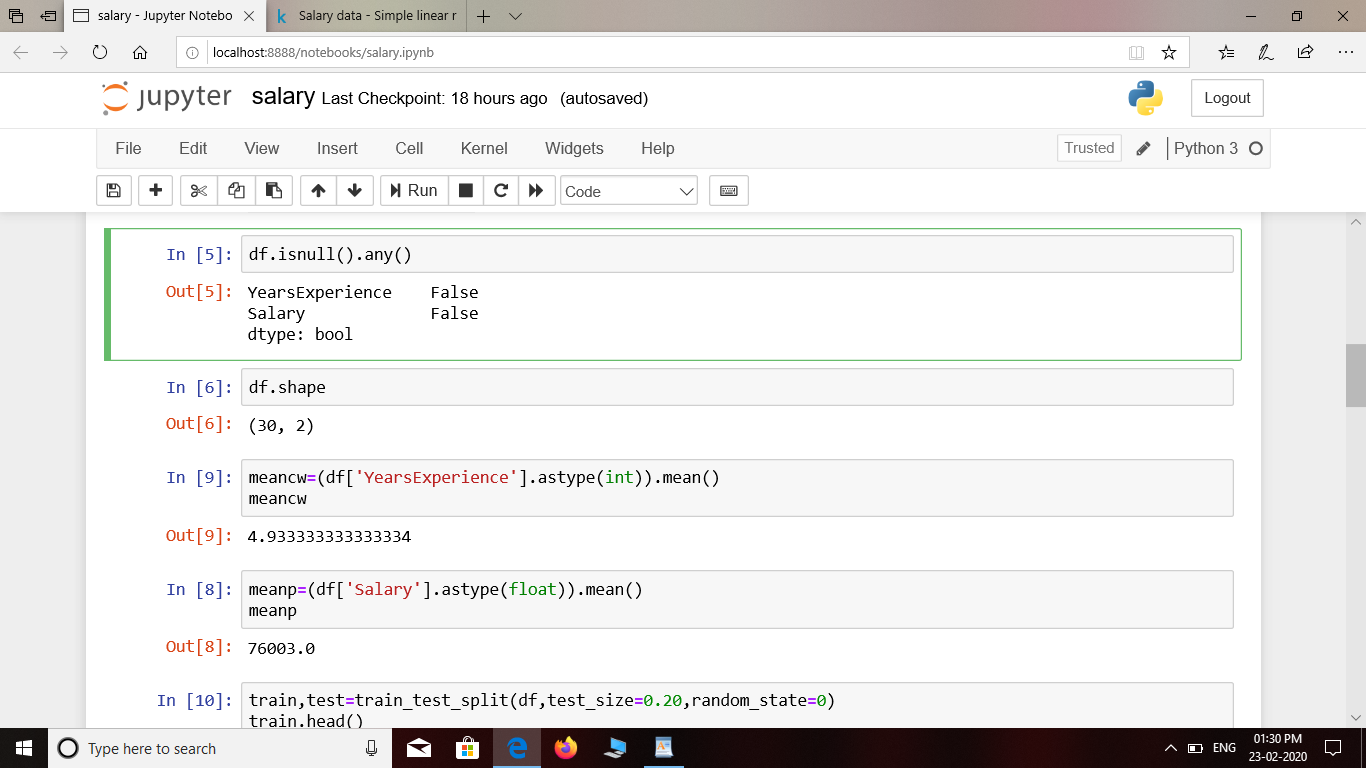
df.describe()



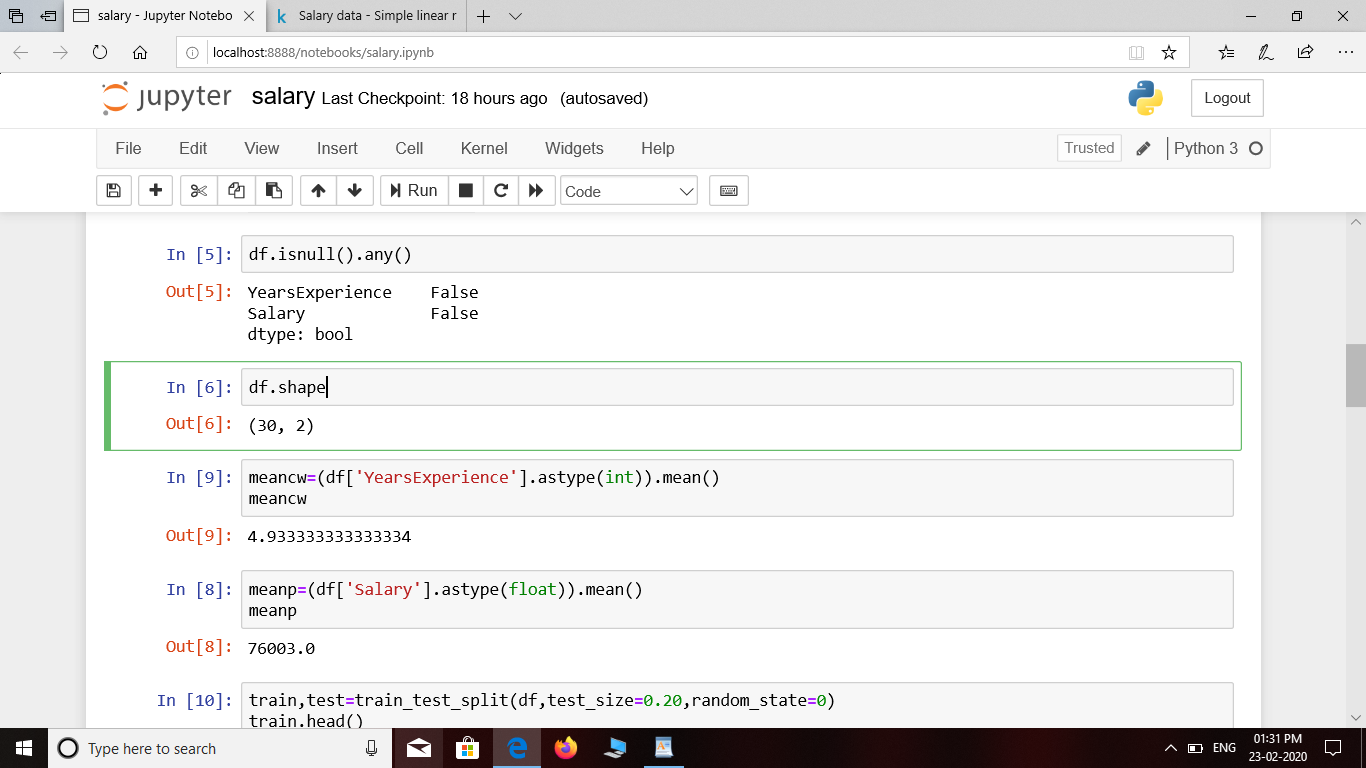
2)df.head()



3)df.isnull().any()

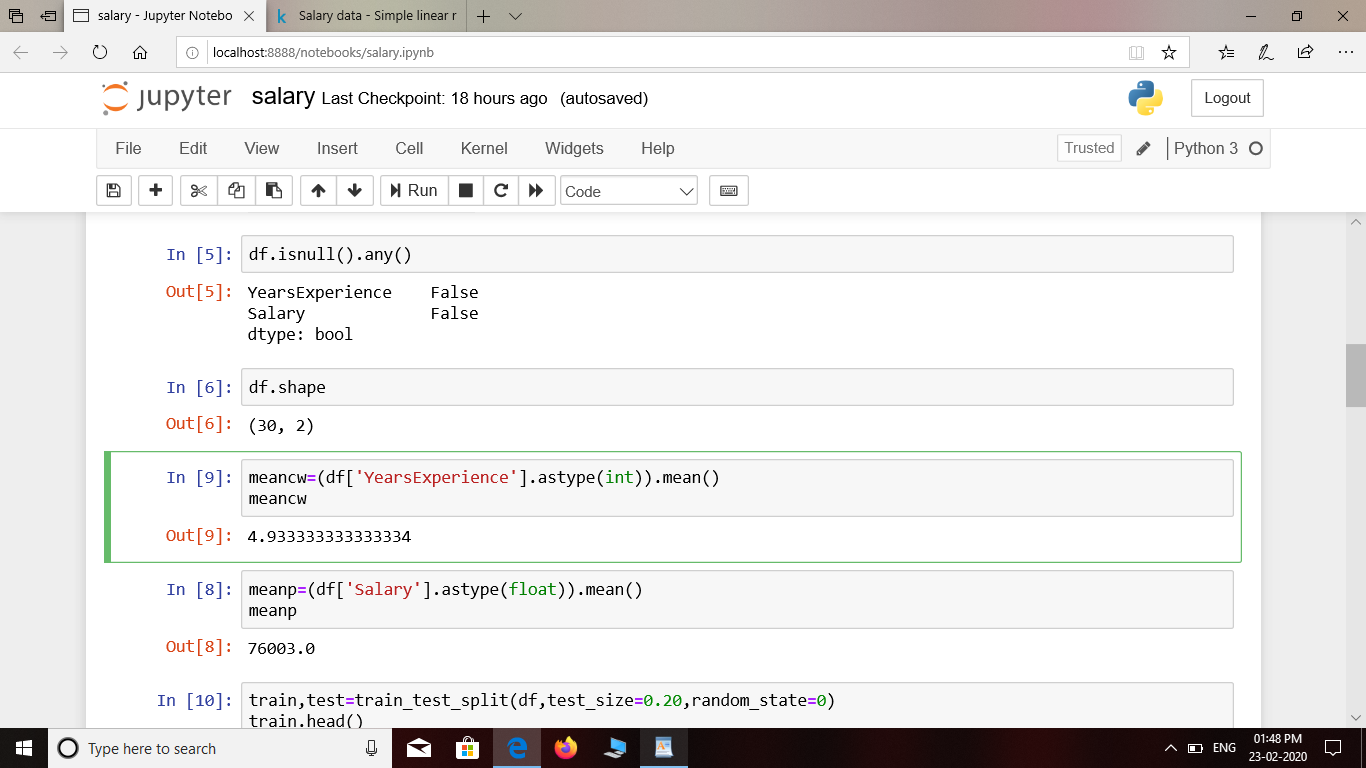


4)df.shape



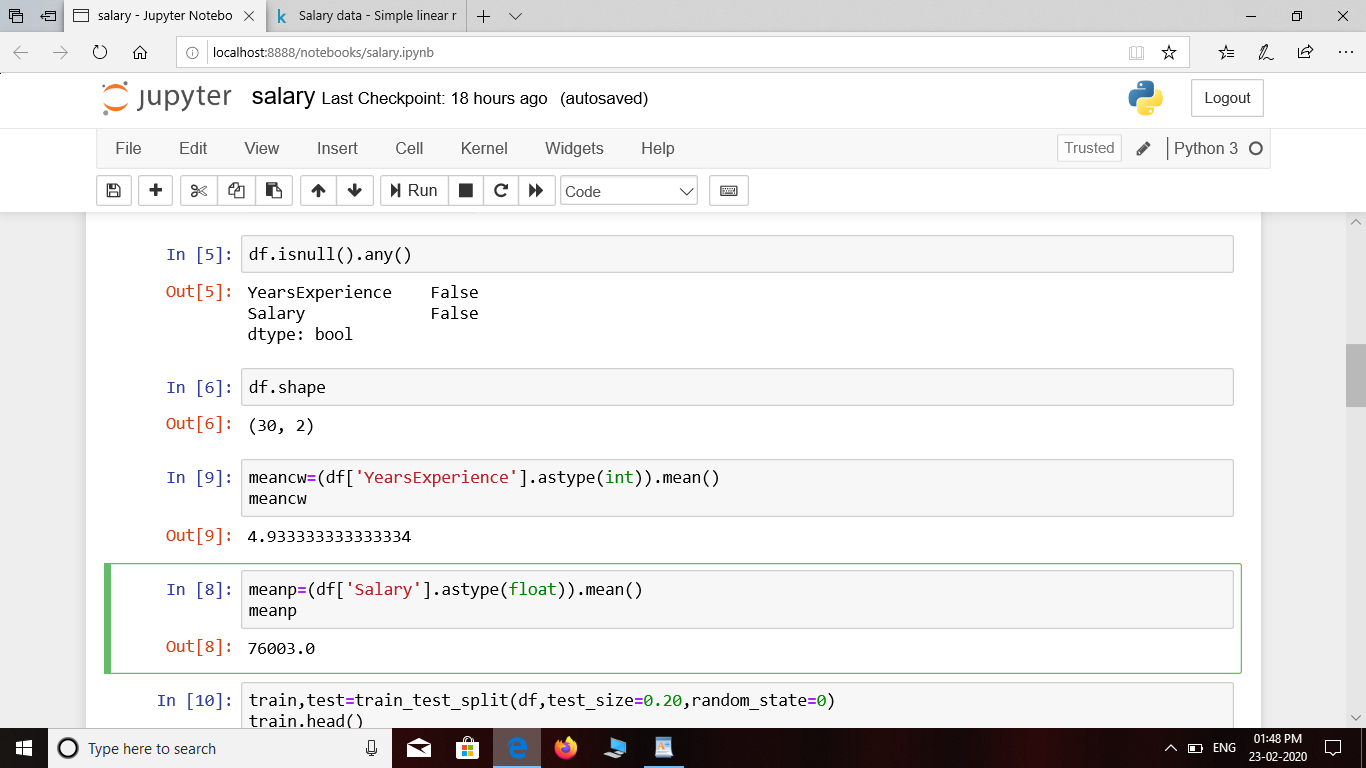
5)meancw=(df['YearsExperience'].astype(int)).mean()

meancw



6)meanp=(df['Salary'].astype(float)).mean()

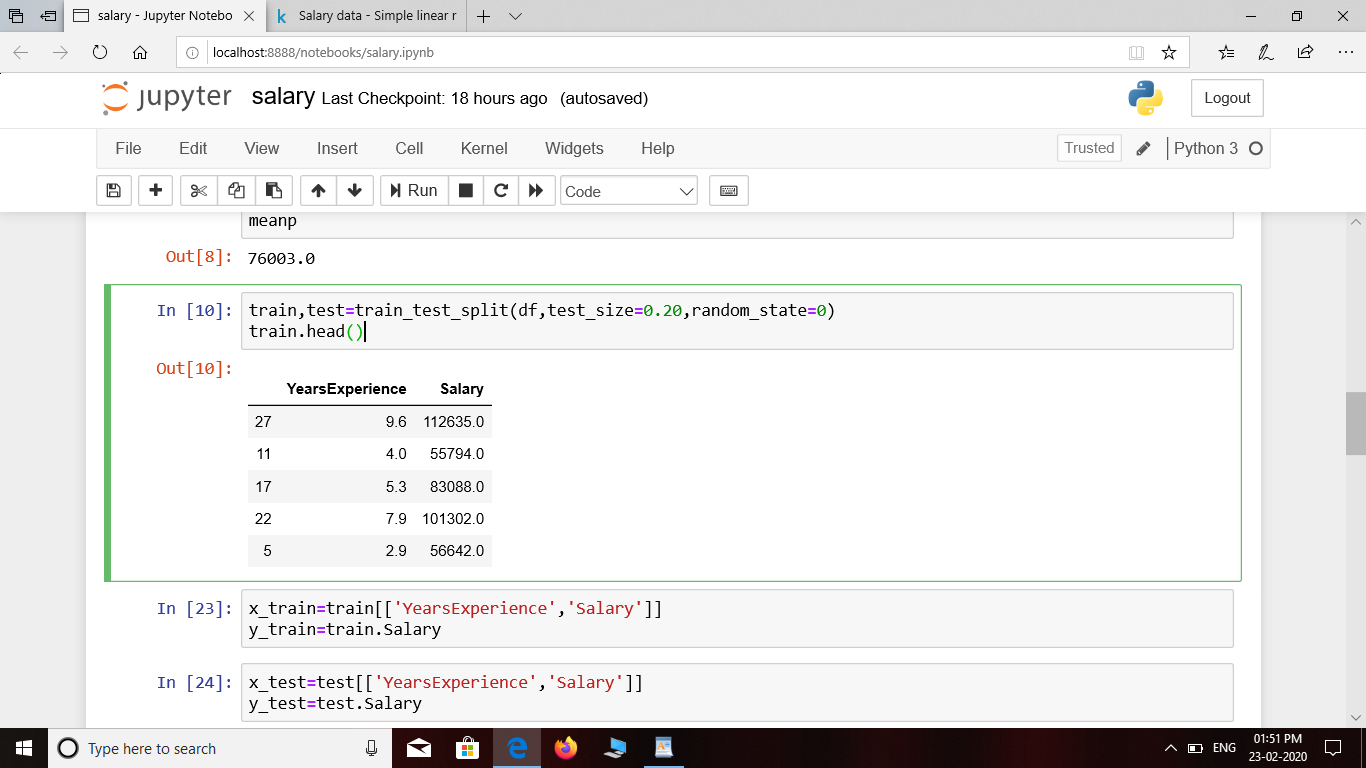
Meanp



**Buliding,training and evaluation of all ML models**

train,test=train\_test\_split(df,test\_size=0.20,random\_state=0)

train.head()



x\_train=train[['YearsExperience','Salary']]

y\_train=train.Salary

x\_test=test[['YearsExperience','Salary']]

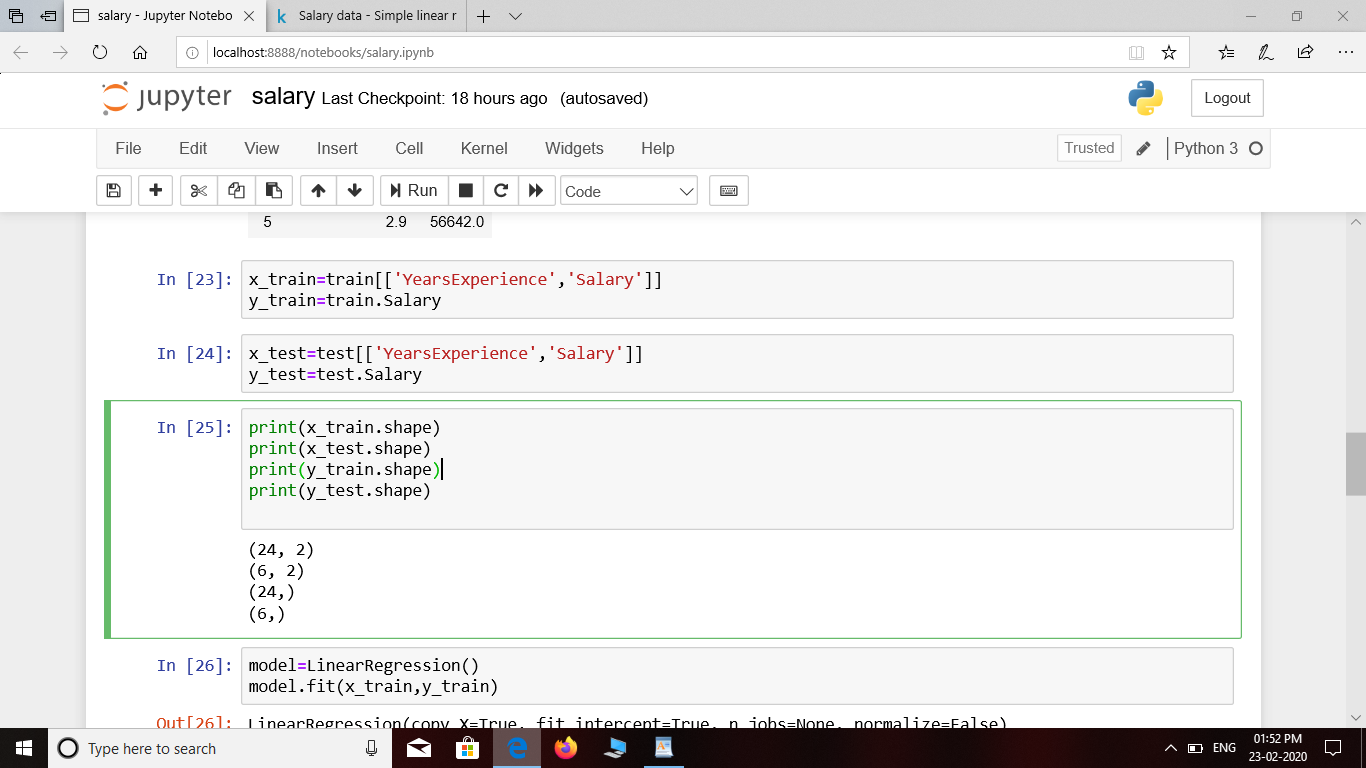
y\_test=test.Salary

print(x\_train.shape)

print(x\_test.shape)

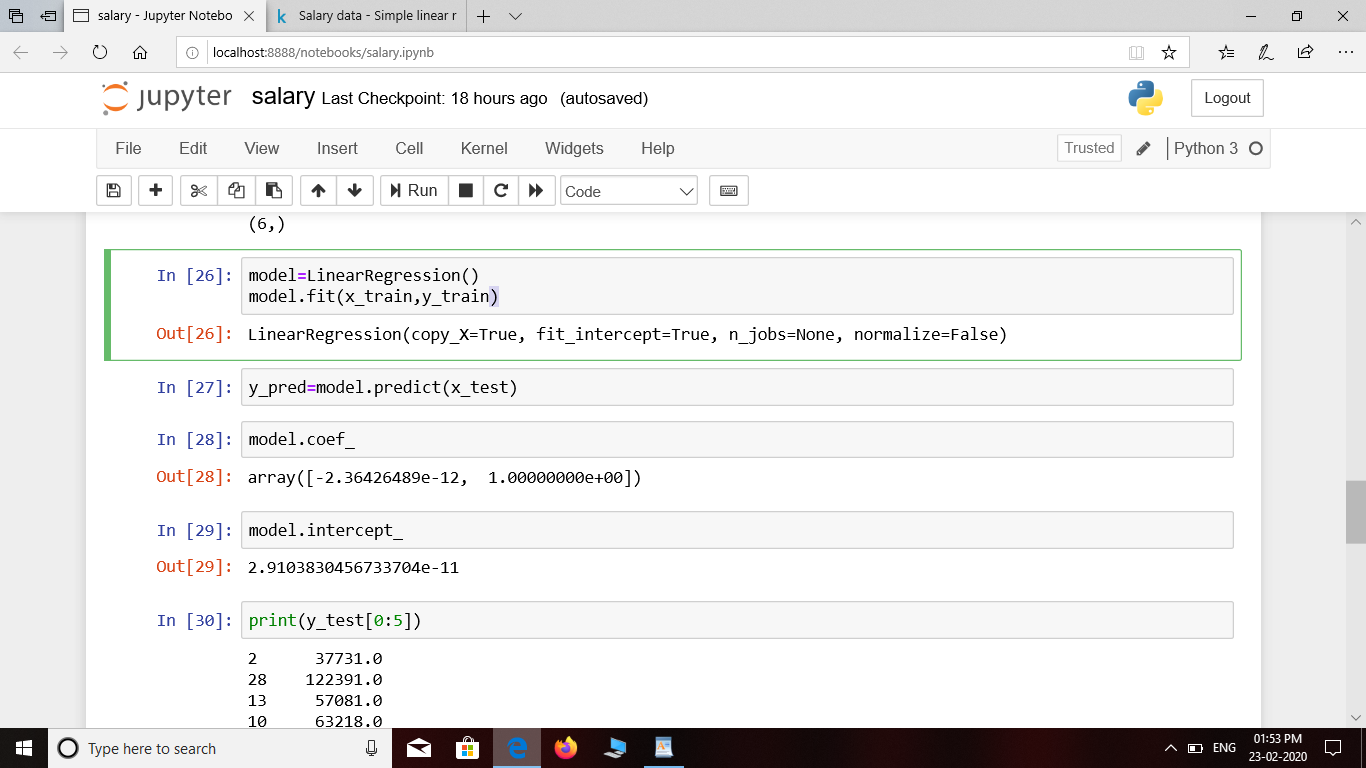
print(y\_train.shape)

print(y\_test.shape)



model=LinearRegression()

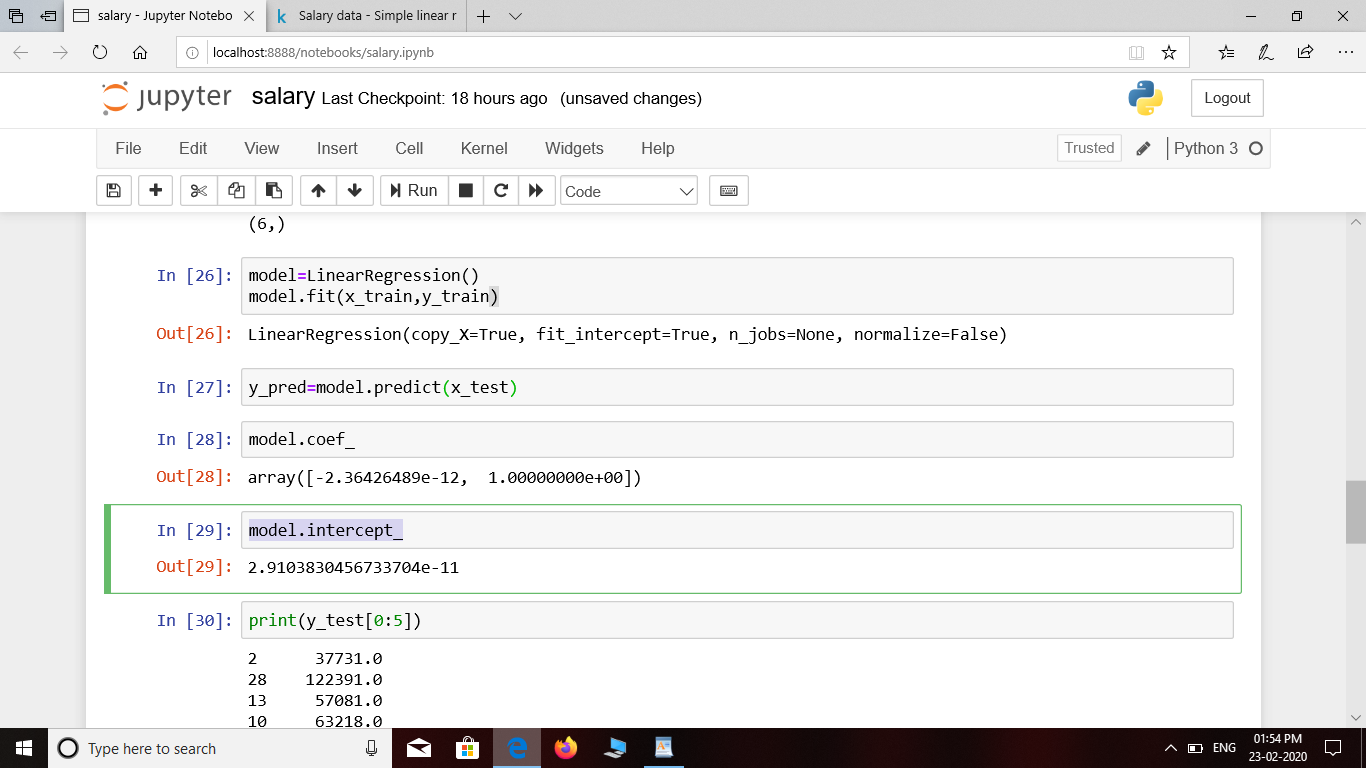
model.fit(x\_train,y\_train)



y\_pred=model.predict(x\_test)

model.coef\_

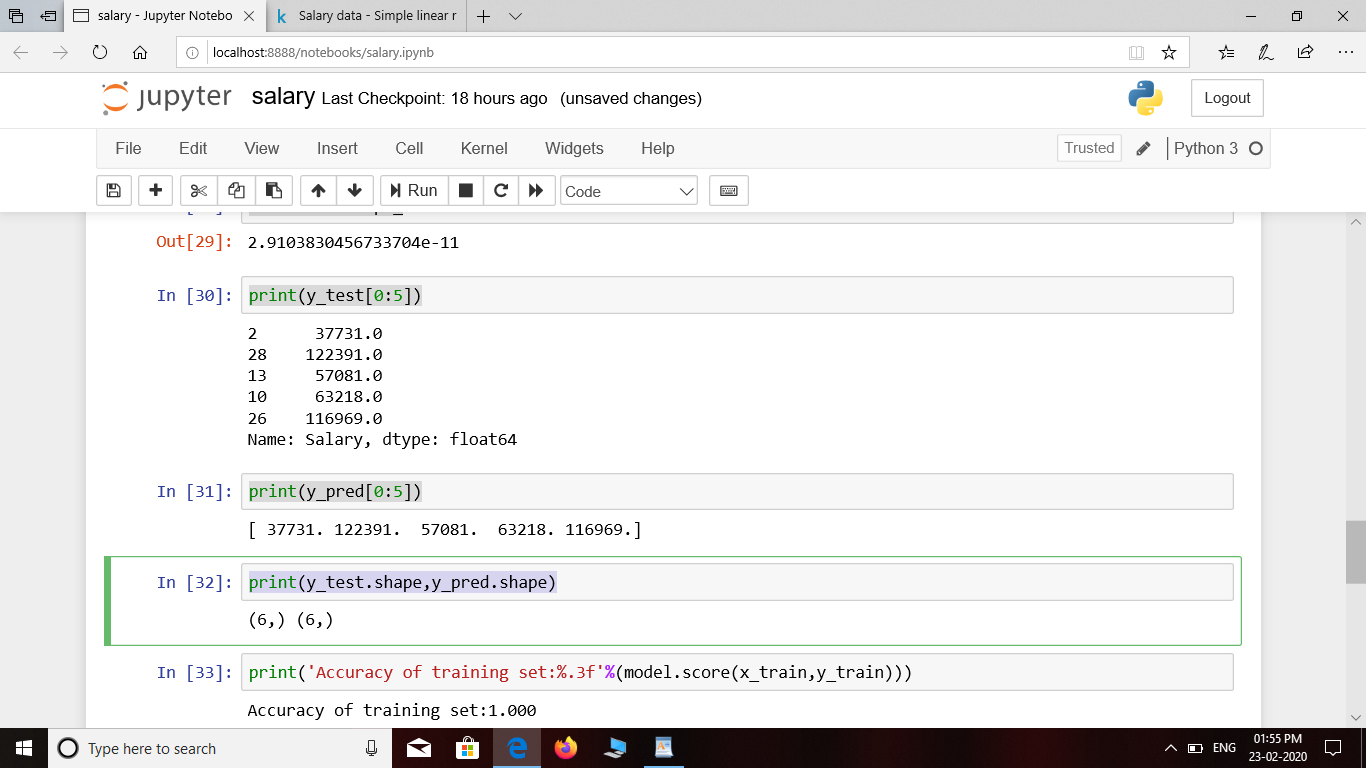
model.intercept\_



print(y\_test[0:5])

print(y\_pred[0:5])

print(y\_test.shape,y\_pred.shape)



print('Accuracy of training set:%.3f'%(model.score(x\_train,y\_train)))

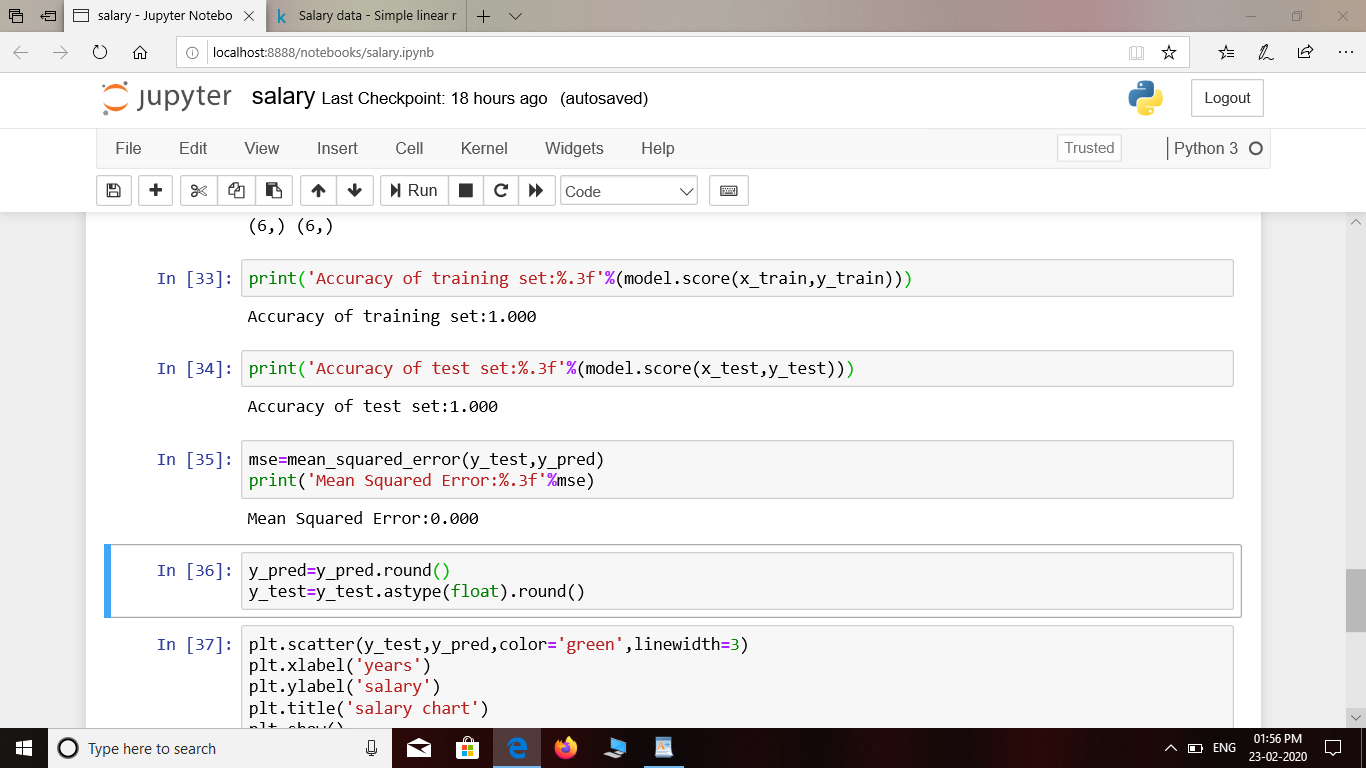
print('Accuracy of test set:%.3f'%(model.score(x\_test,y\_test)))

mse=mean\_squared\_error(y\_test,y\_pred)

print('Mean Squared Error:%.3f'%mse)

y\_pred=y\_pred.round()

y\_test=y\_test.astype(float).round()



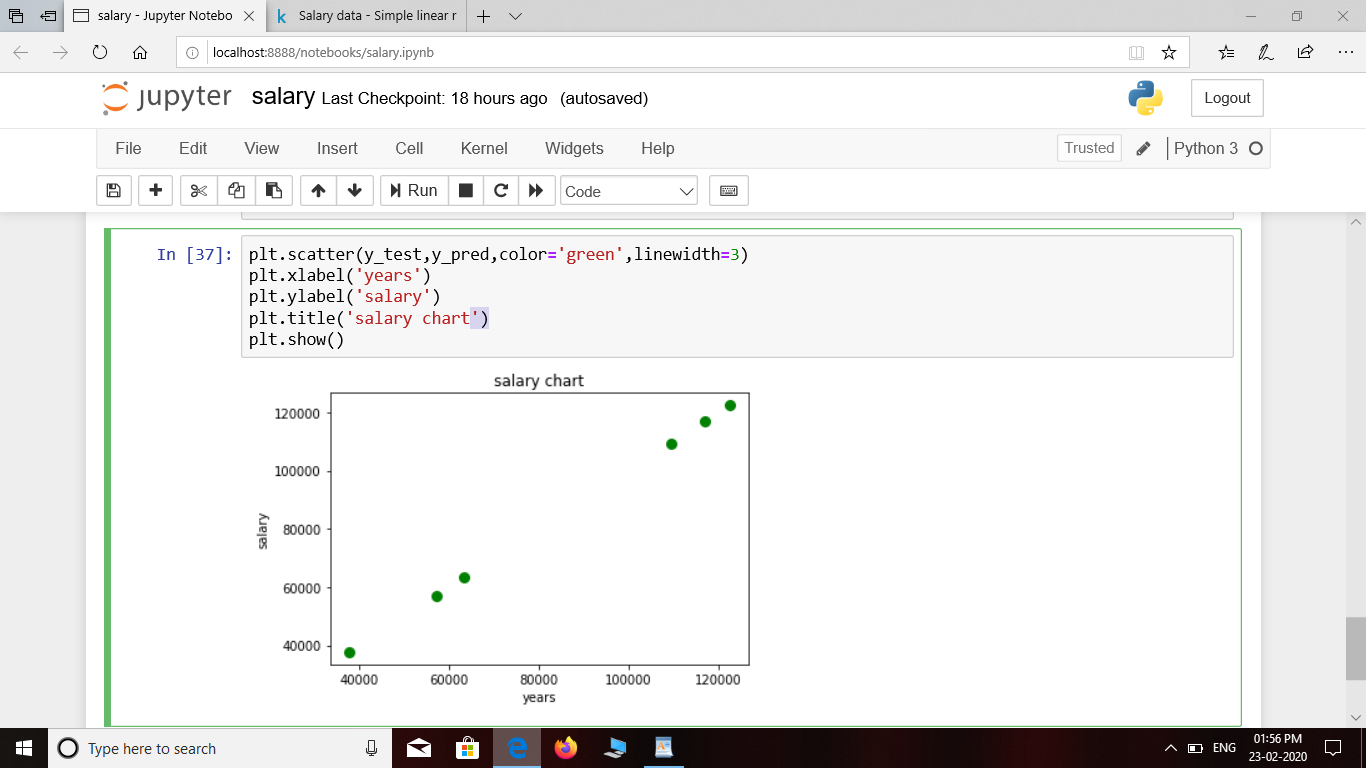
plt.scatter(y\_test,y\_pred,color='green',linewidth=3)

plt.xlabel('years')

plt.ylabel('salary')

plt.title('salary chart')

plt.show()



**CONCLUSION:**

Here salary is predicted with linear regression and relationship between salary and years of experiences is established.

