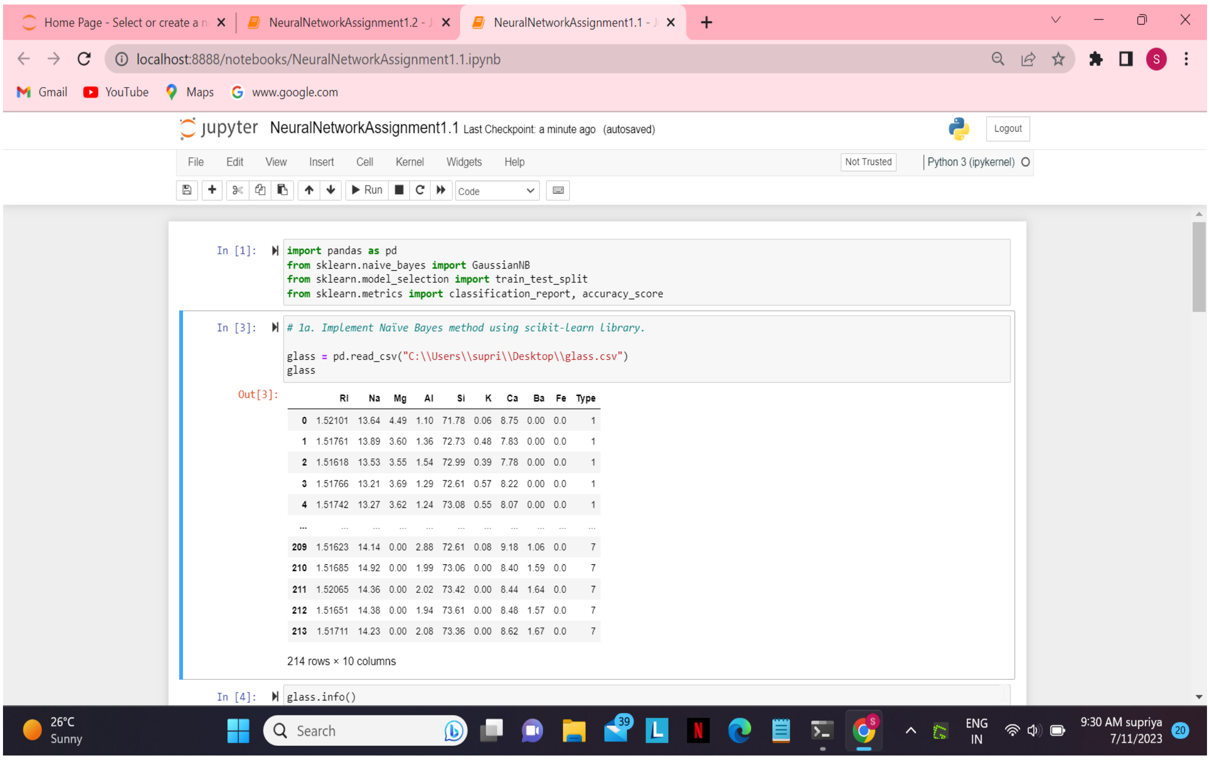
NEURAL NETWORK ASSIGNMENT 1

NAME: SUPRIYA SAMA
700744510

Video link:
https://drive.google.com/file/d/1rlyf51r3MSyEBM9fULhR1R2QffTP_4bP/view?usp=sharing

1. **Implement Naive Bayes method using scikit-learn library. Use the dataset available with name glass.**

   Using the read_csv method from the pandas module I imported a glass data set.



```
In [1]:
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
```

```
In [3]:
# 1a. Implement Naïve Bayes method using scikit-learn library.

glass = pd.read_csv("C:\\Users\\supri\\Desktop\\glass.csv")
glass
```

Out[3]:

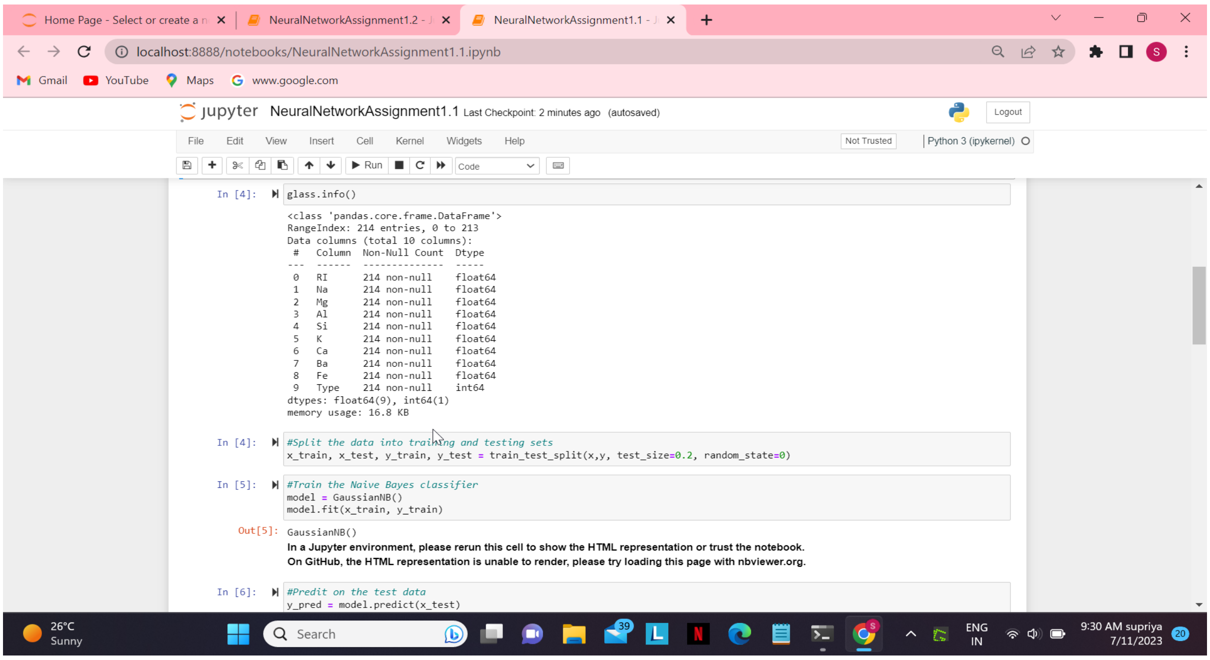
	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	Type
0	1.52101	13.64	4.49	1.10	71.78	0.06	8.75	0.00	0.0	1
1	1.51761	13.89	3.60	1.36	72.73	0.48	7.83	0.00	0.0	1
2	1.51618	13.53	3.55	1.54	72.99	0.39	7.78	0.00	0.0	1
3	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0.00	0.0	1
4	1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0.00	0.0	1
...	...	...	...	...	...	...	...	...	...	...
209	1.51623	14.14	0.00	2.88	72.61	0.08	9.18	1.06	0.0	7
210	1.51685	14.92	0.00	1.99	73.06	0.00	8.40	1.59	0.0	7
211	1.52065	14.36	0.00	2.02	73.42	0.00	8.44	1.64	0.0	7
212	1.51651	14.38	0.00	1.94	73.61	0.00	8.48	1.57	0.0	7
213	1.51711	14.23	0.00	2.08	73.36	0.00	8.62	1.67	0.0	7

214 rows × 10 columns

```
In [4]:
glass.info()
```

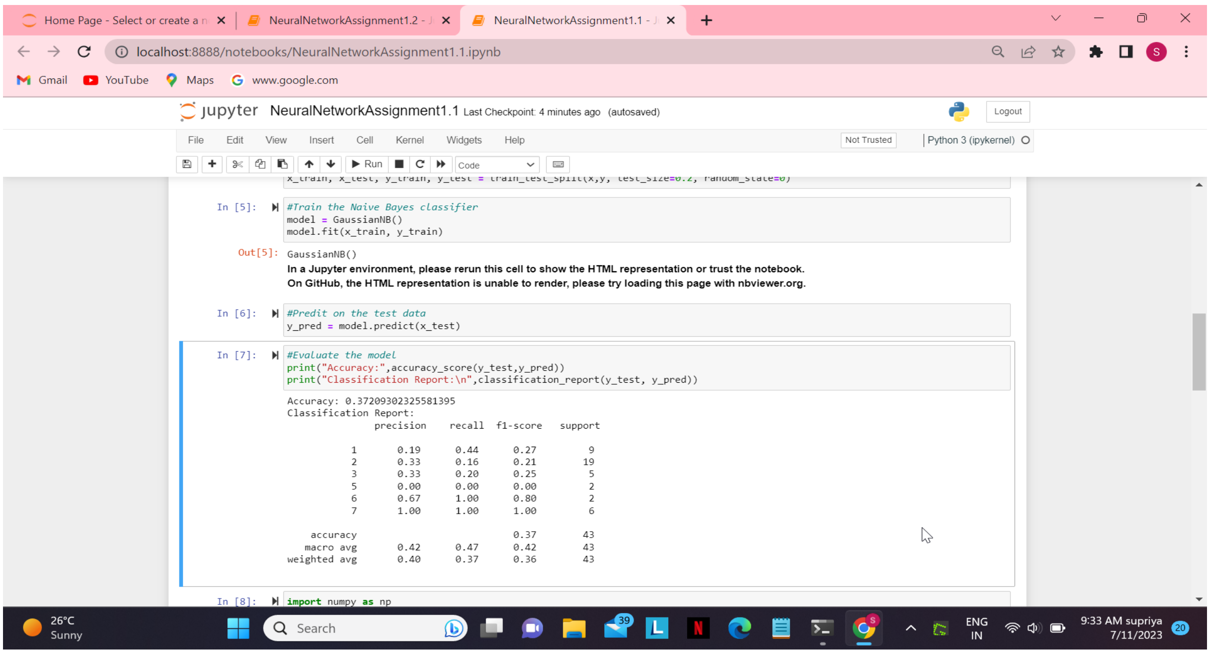
Use train_test_split to create the training and testing part.

sklearn module contains train_test_split method to split our data set into training and testing data sets. In this data set Type column can be used for labels. In this method, test_size defines how much proportion of data to be in the test data set. When we test_size value whole analysis results will change.



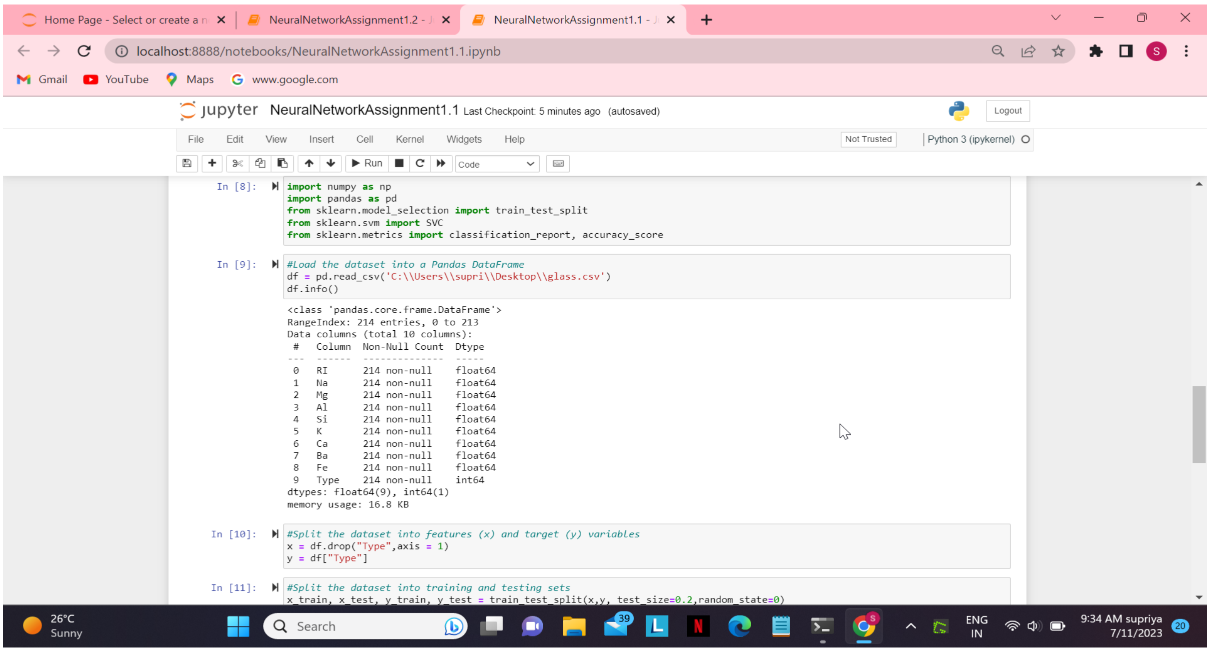
Evaluate the model on testing part using score and classification_report(y_true, y_pred)

In the given data, there are no missing values present in it. So, we can directly apply the machine learning algorithms on the data. sklearn module is imported to analyze the data using different algorithms. Classification_report and confusion_matrix methods to result the summary of the predictions made using the specific algorithm. These summaries can be used to compare with another algorithms to define which algorithm is better.

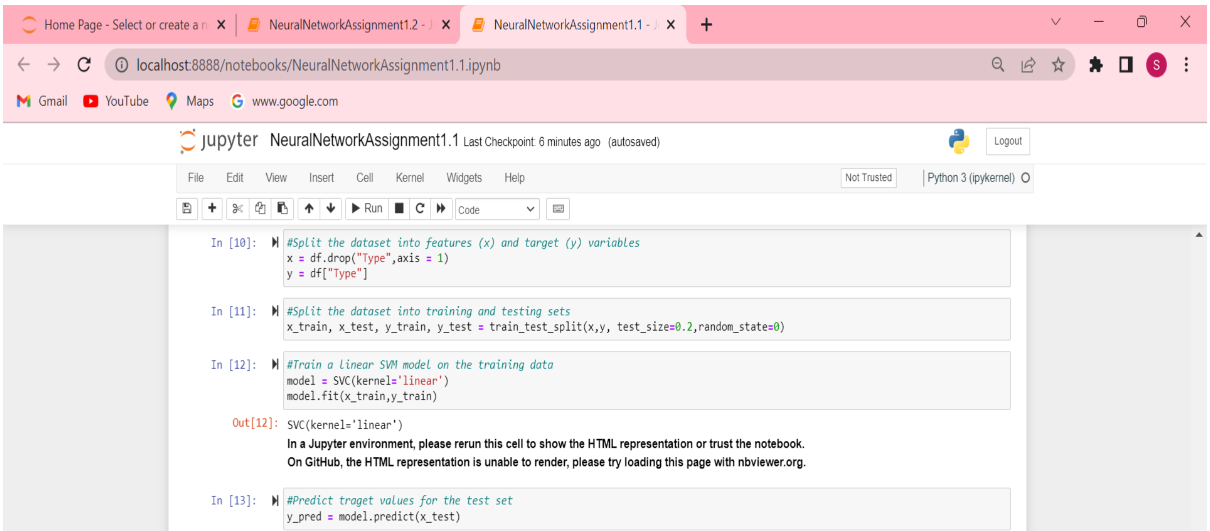


2. Implement linear SVM method using scikit library.

Using read_csv method from pandas module I imported glass data set.

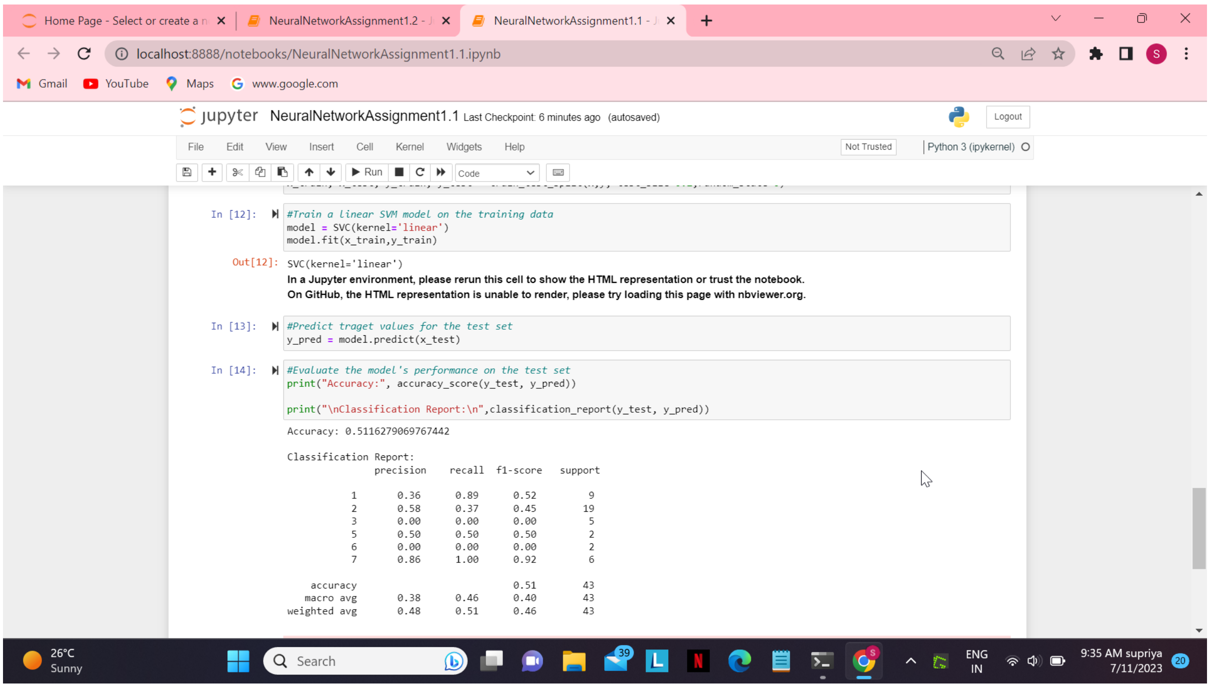


Use train_test_split to create the training and testing part. sklearn module contains train_test_split method to split our data set into training and testing data sets. In this data set Type column can be used for labels. In this method, test_size defines how much proportion of data to be in the test data set. When we test_size value whole analysis results will change.



Evaluate the model on testing part using score and classification_report(y_true, y_pred)

Support vector machine algorithm is applied to this data set using sklearn module.



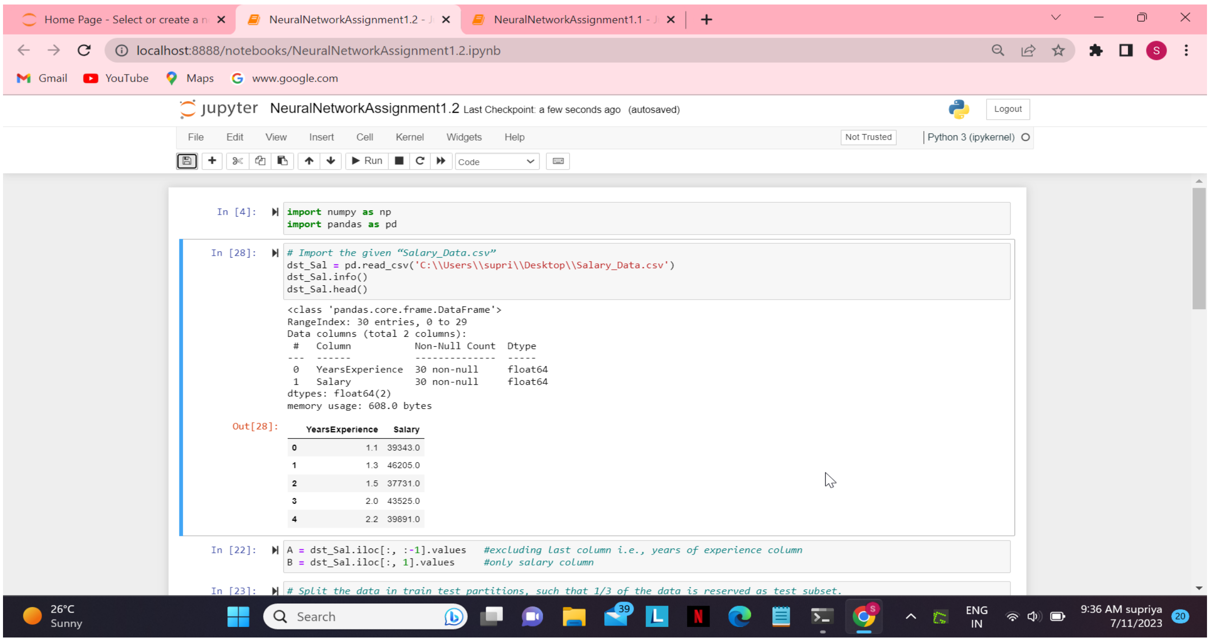
Which algorithm got better accuracy? Can you justify why?

We got better accuracy for the SVM method. We get better results using SVM than naïve bayes when we work with a glass data set. In this glass data set, SVM can handle more data than the naive bayes algorithm. Naive Bayes assumes independence between features, which can limit its ability to model complex relationships in the data.

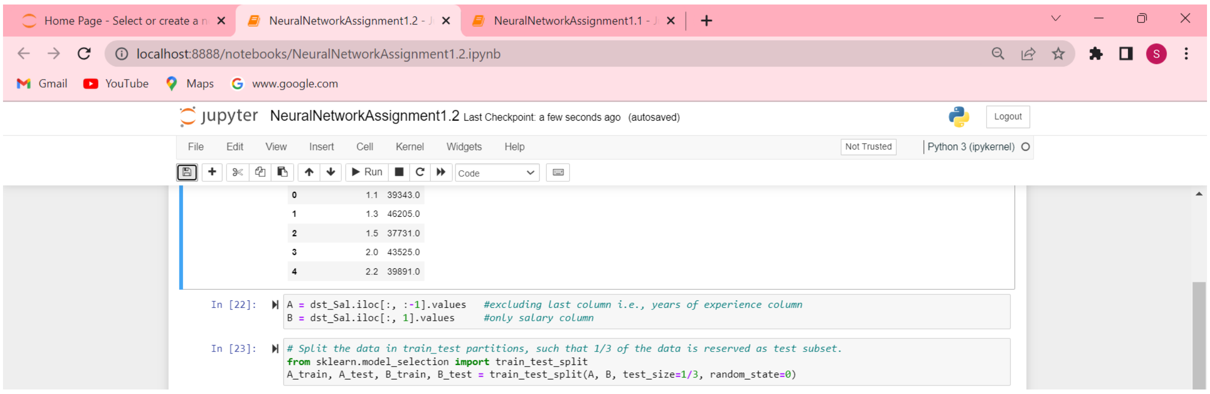
3. Implement Linear Regression using scikit-learn.

A) Import the given “Salary_Data.csv”

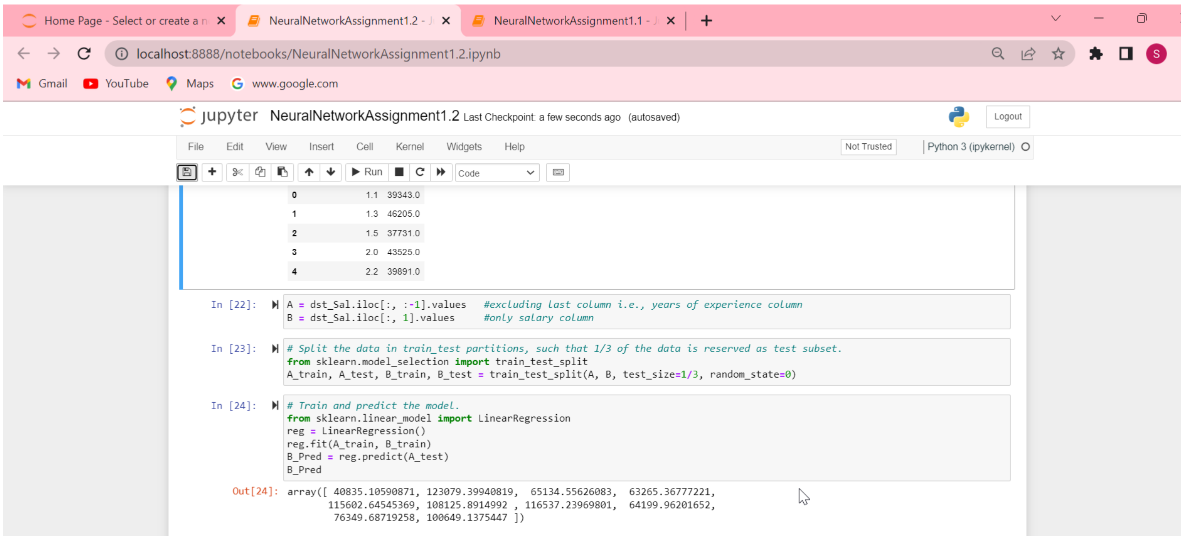
I have imported the Salary_Data.csv by using read_csv.



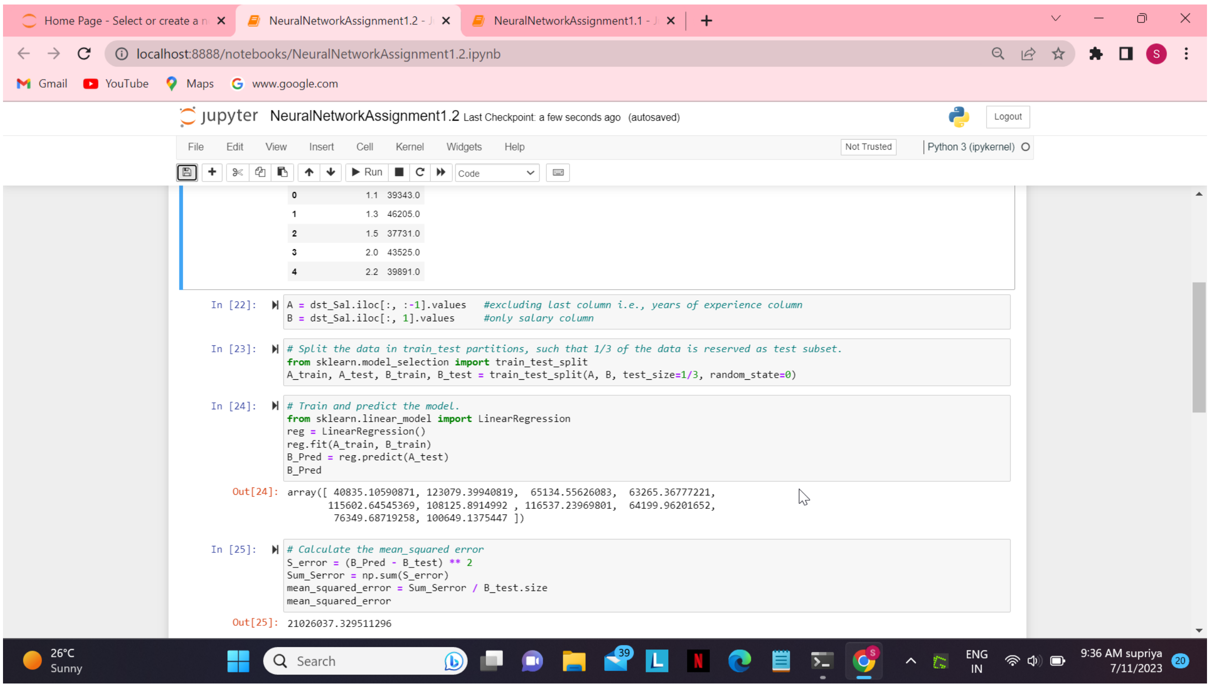
B. Split the data in train_test partitions, such that ⅓ of the data is reserved as test subset.



C. Train and predict the model



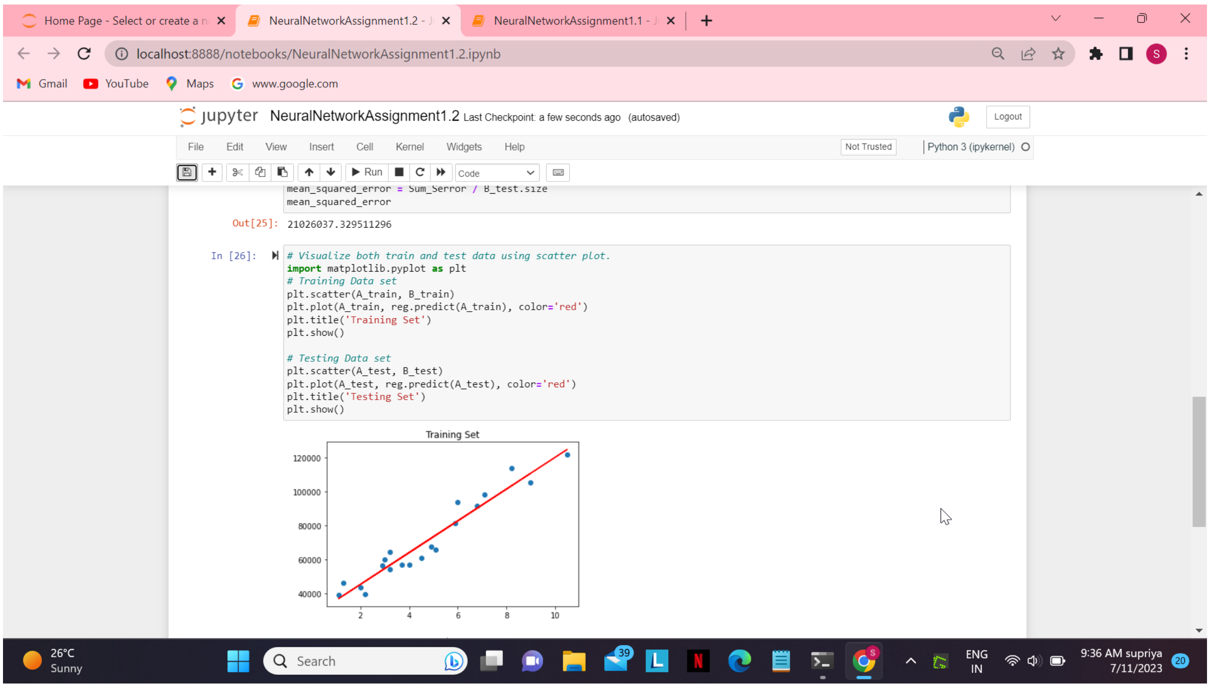
D. Calculate the mean_squared error.

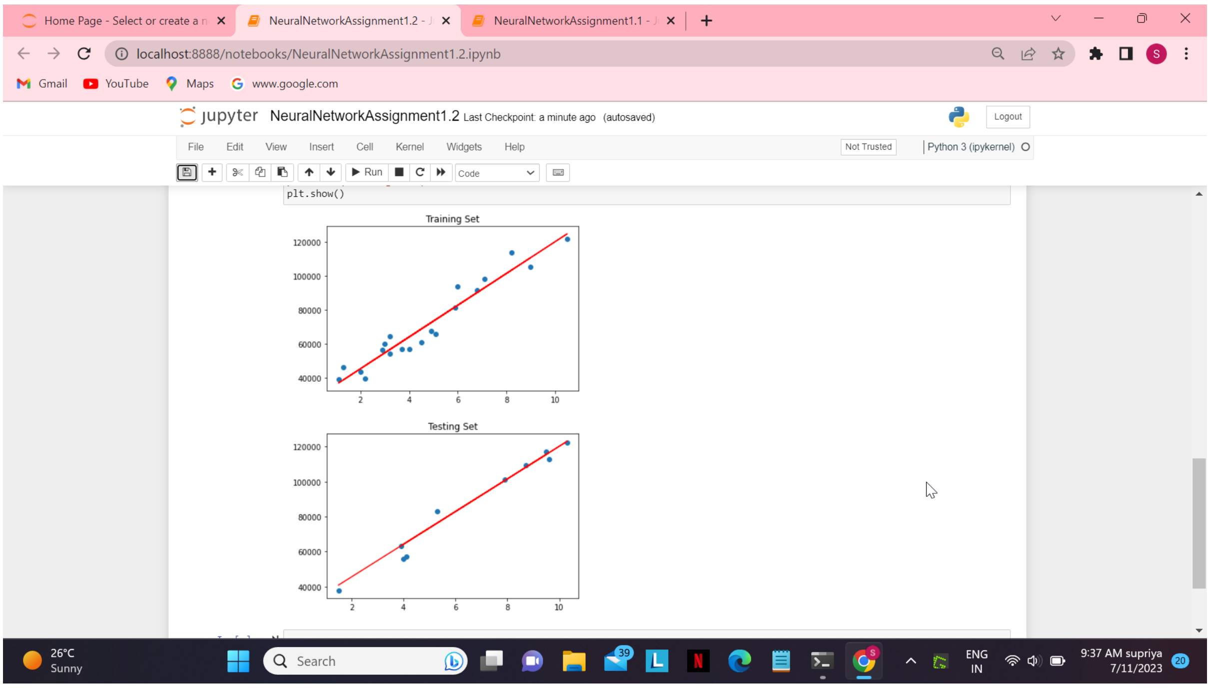


We can see that we have got mean square error.

E. Visualise both train and test data using scatter plot.

We can see the scatter plots of both training and testing data.


localhost:8888/notebooks/NeuralNetworkAssignment1.2.ipynb
jupyter NeuralNetworkAssignment1.2 Last Checkpoint: a few seconds ago (autosaved)
mean_squared_error = Sum_Serror / B_test.size
mean_squared_error
Out[25]: 21026037.329511296
In [26]:
# Visualize both train and test data using scatter plot.
import matplotlib.pyplot as plt
# Training Data set
plt.scatter(A_train, B_train)
plt.plot(A_train, reg.predict(A_train), color='red')
plt.title('Training Set')
plt.show()
# Testing Data set
plt.scatter(A_test, B_test)
plt.plot(A_test, reg.predict(A_test), color='red')
plt.title('Testing Set')
plt.show()
Training Set
9:36 AM supriya
7/11/2023


localhost:8888/notebooks/NeuralNetworkAssignment1.2.ipynb
jupyter NeuralNetworkAssignment1.2 Last Checkpoint: a minute ago (autosaved)
plt.show()
Training Set
Testing Set
9:37 AM supriya
7/11/2023