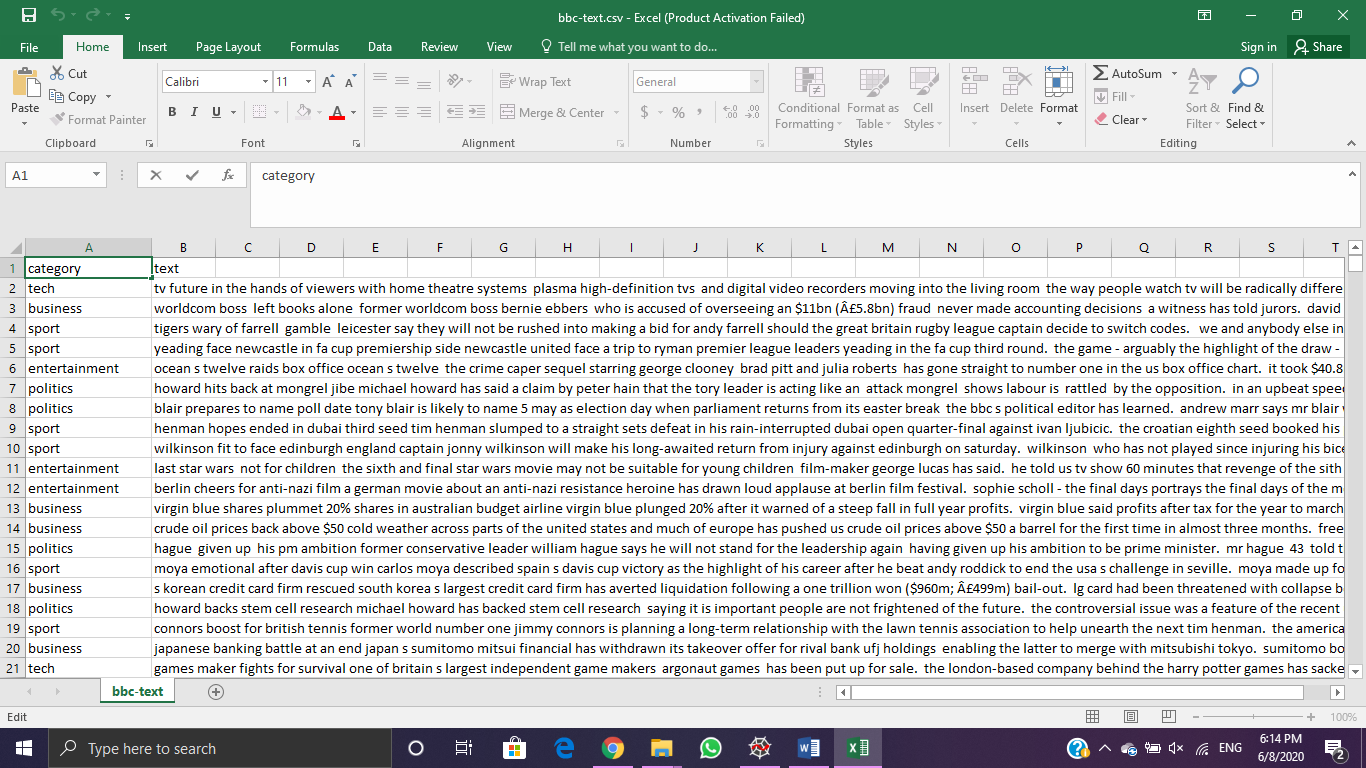
**Implementation steps:**

1. **Download Anaconda latest version from its official site. and configure it.**
2. **Open Spyder IDE**
3. **Create a project folder and copy the dataset into the folder (bbc-text.csv)**
4. **Importing dataset in python using pandas**

import pandas as pd

dataset = pd.read\_csv('bbc-text.csv')



**1)imported** **data .**

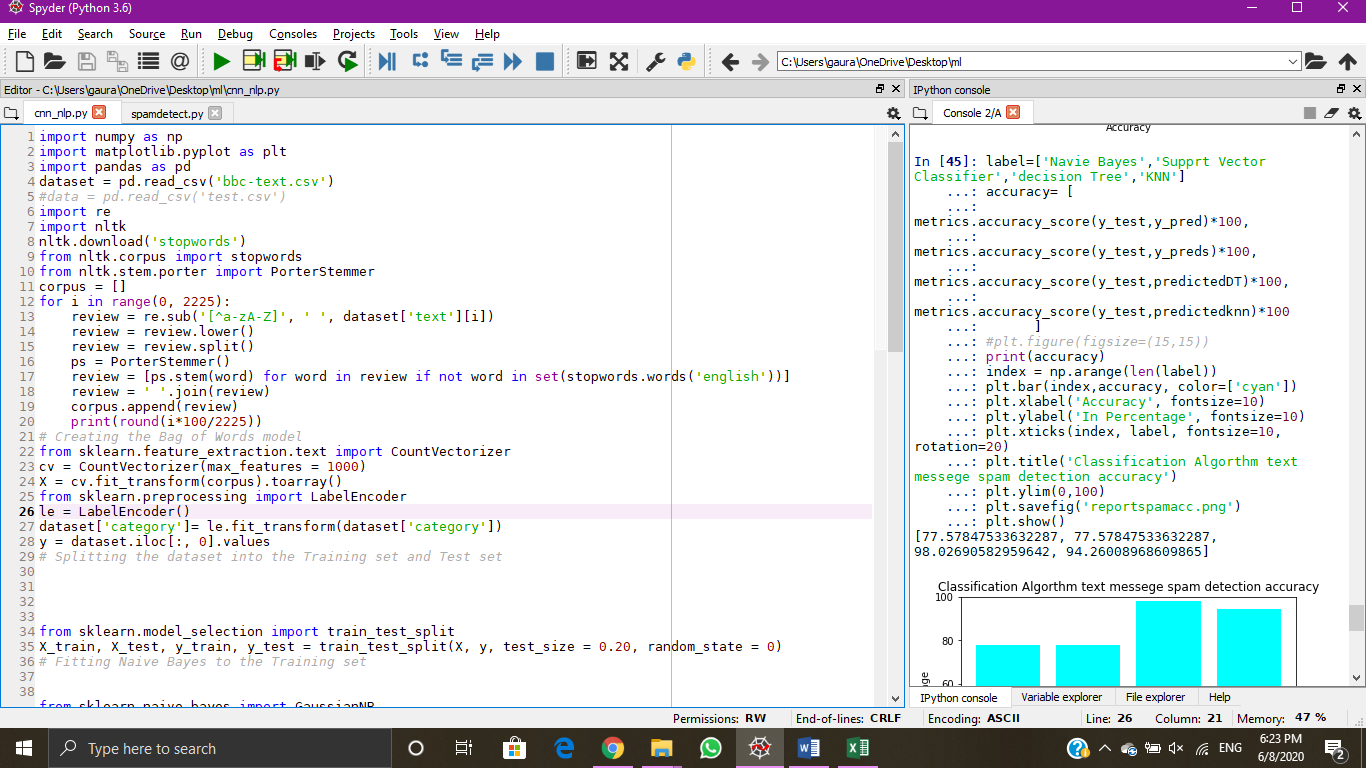
*Category:- classes(tech,business,sports,entertainment) - 4*

*Text :- comments to be classified into classes.*

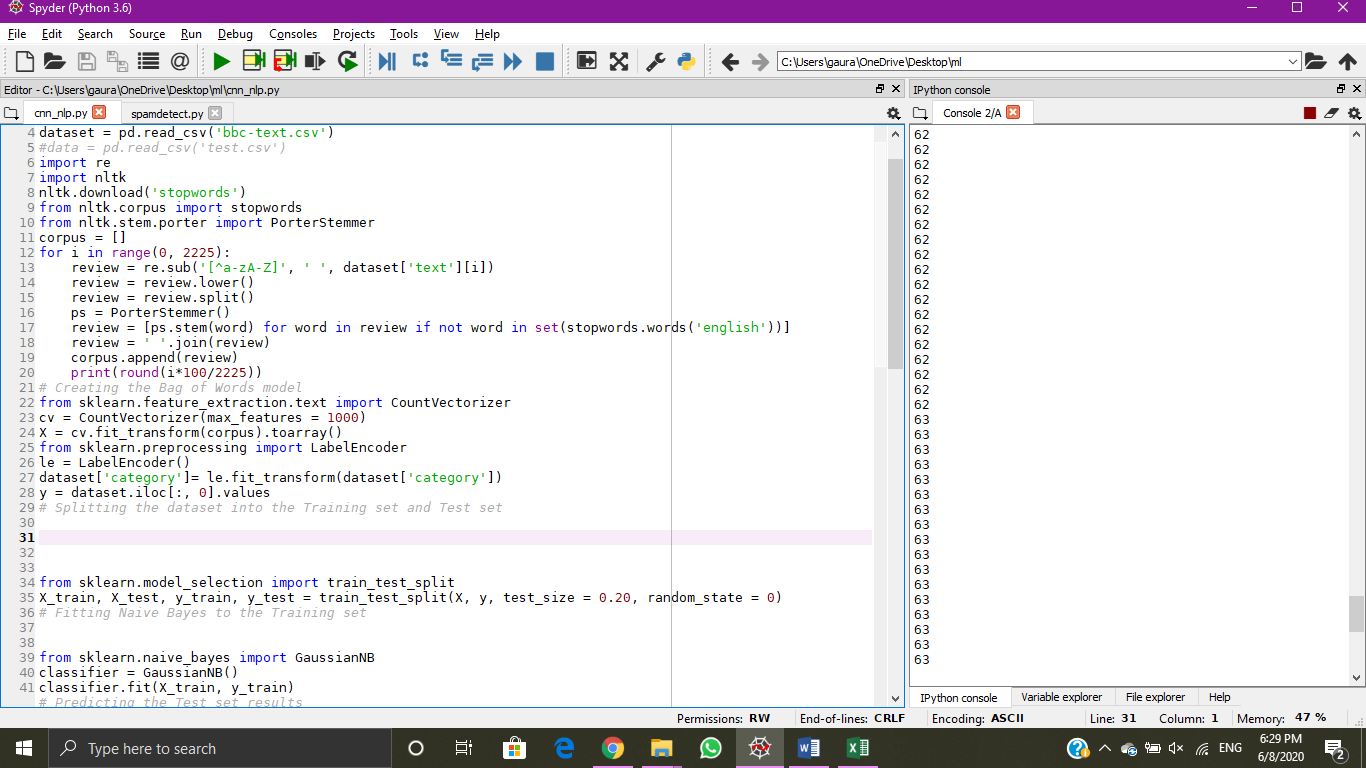
**5.Data preprocessing**

* **Removing symbols -**
* **Considering English words.**

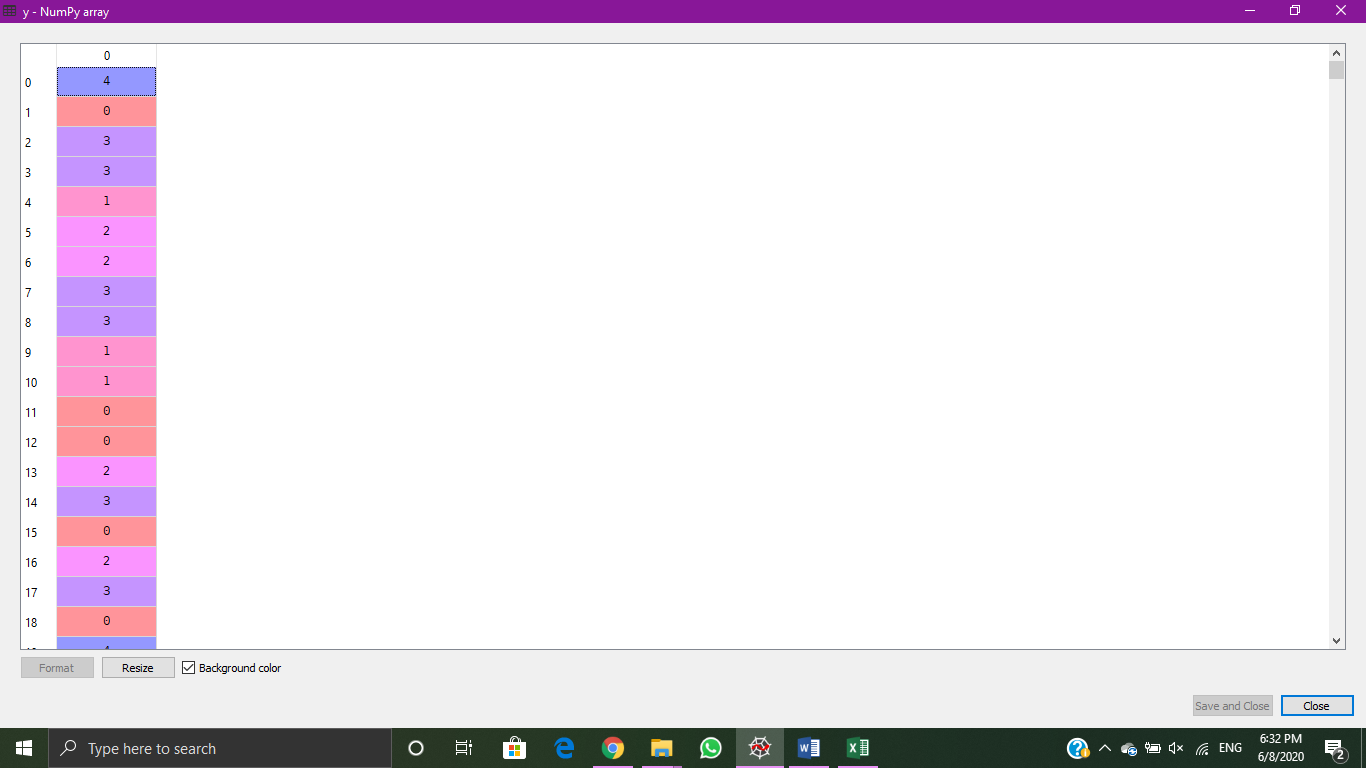
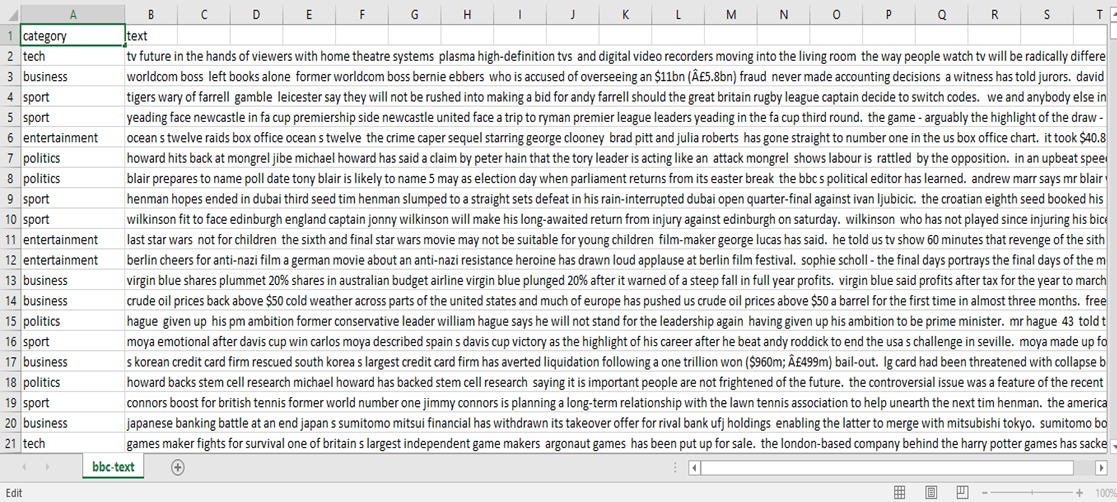
By using natural language processing toolkit .

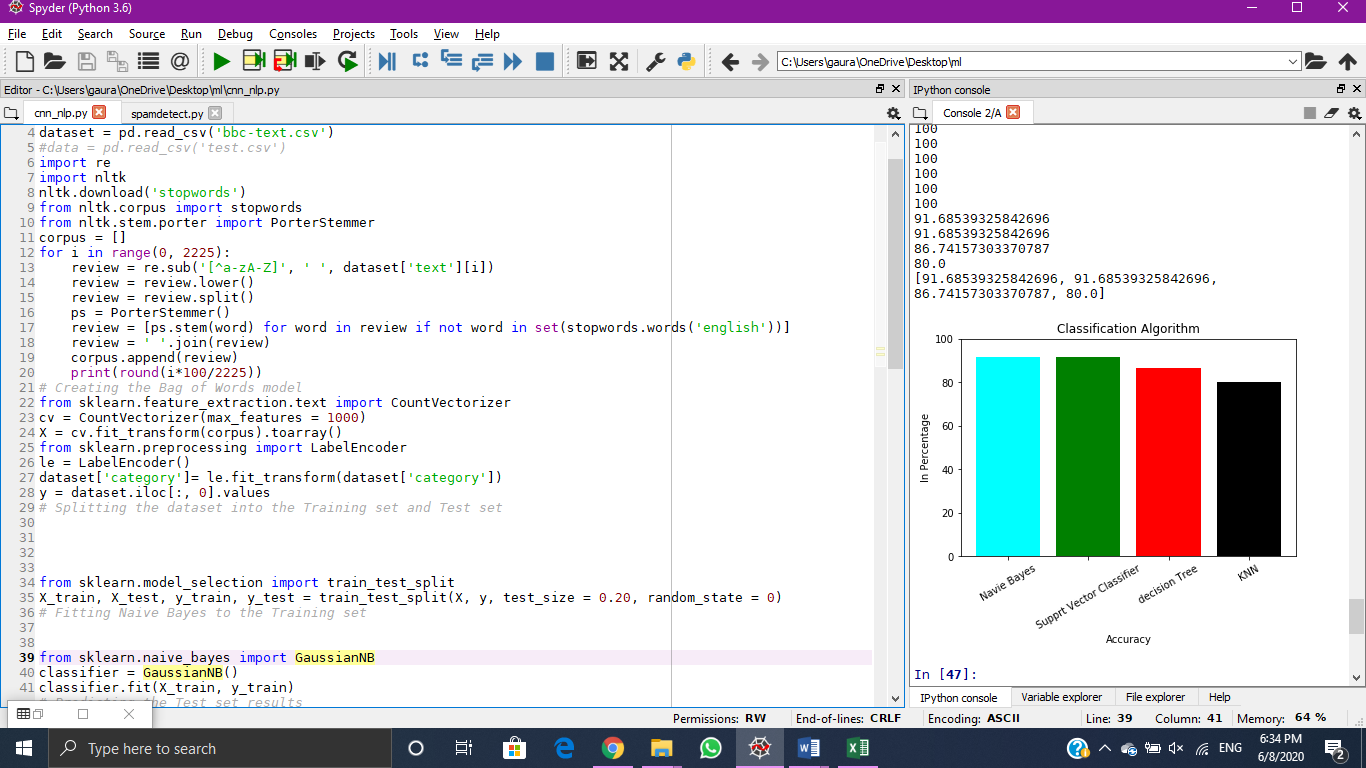


**To convert categorical Data into numeric data using Feature Extraction .**

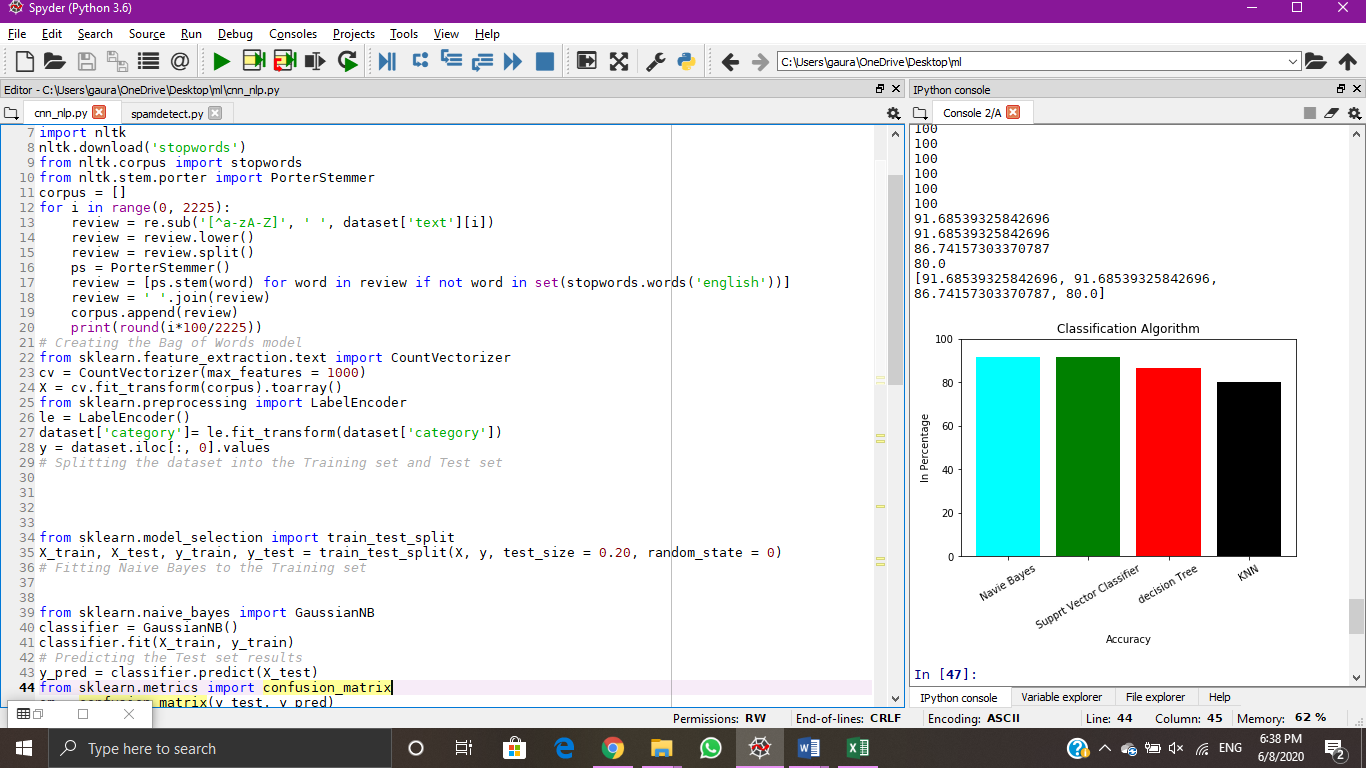


**To convert categorical Data into numeric data using Label Encoding.**

****



**6. Spliting the data into training and testing sets**

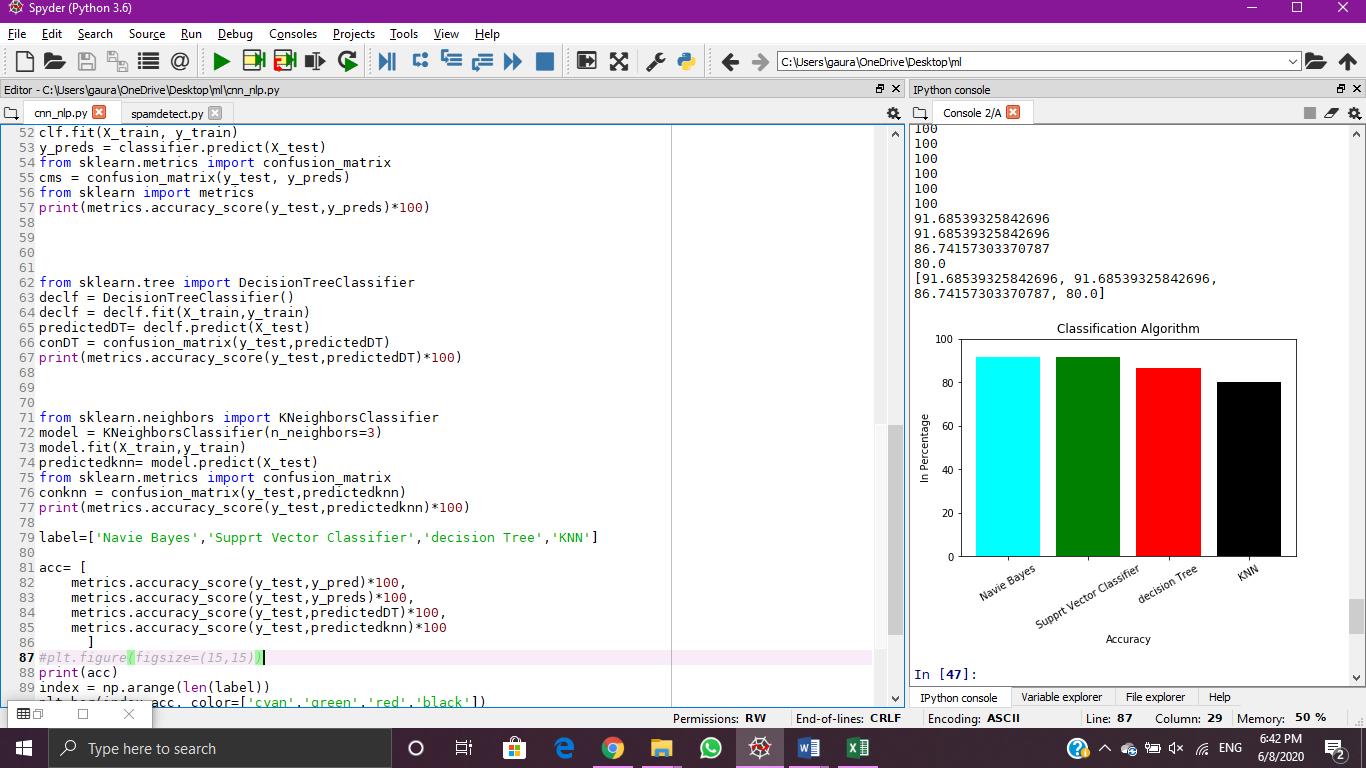


**The dataset is split in ratio of 0.2 means 80% training set and 20% testing set.**

**7. Creating classification models**

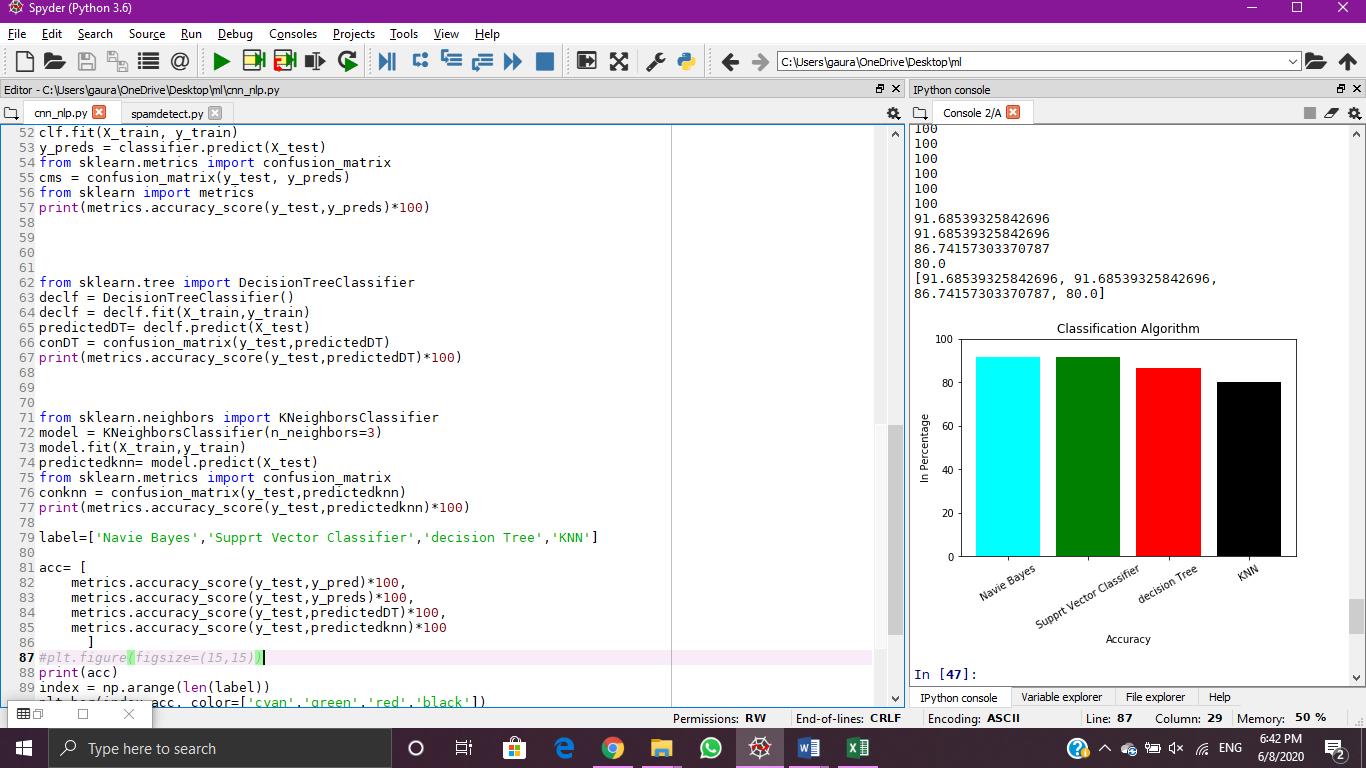
* **KNN classifier**

**Creating classifier**

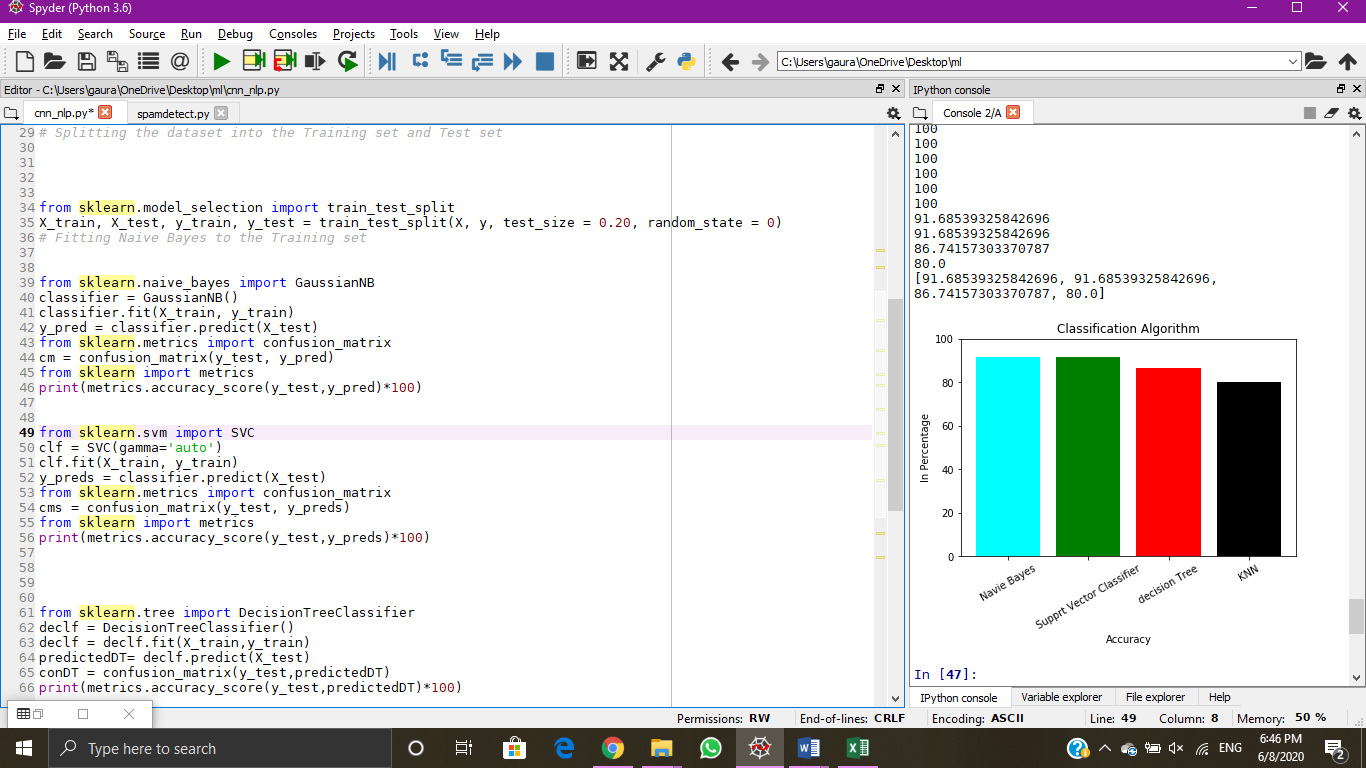


* **Decision Tree:**

**Creating classifier:**

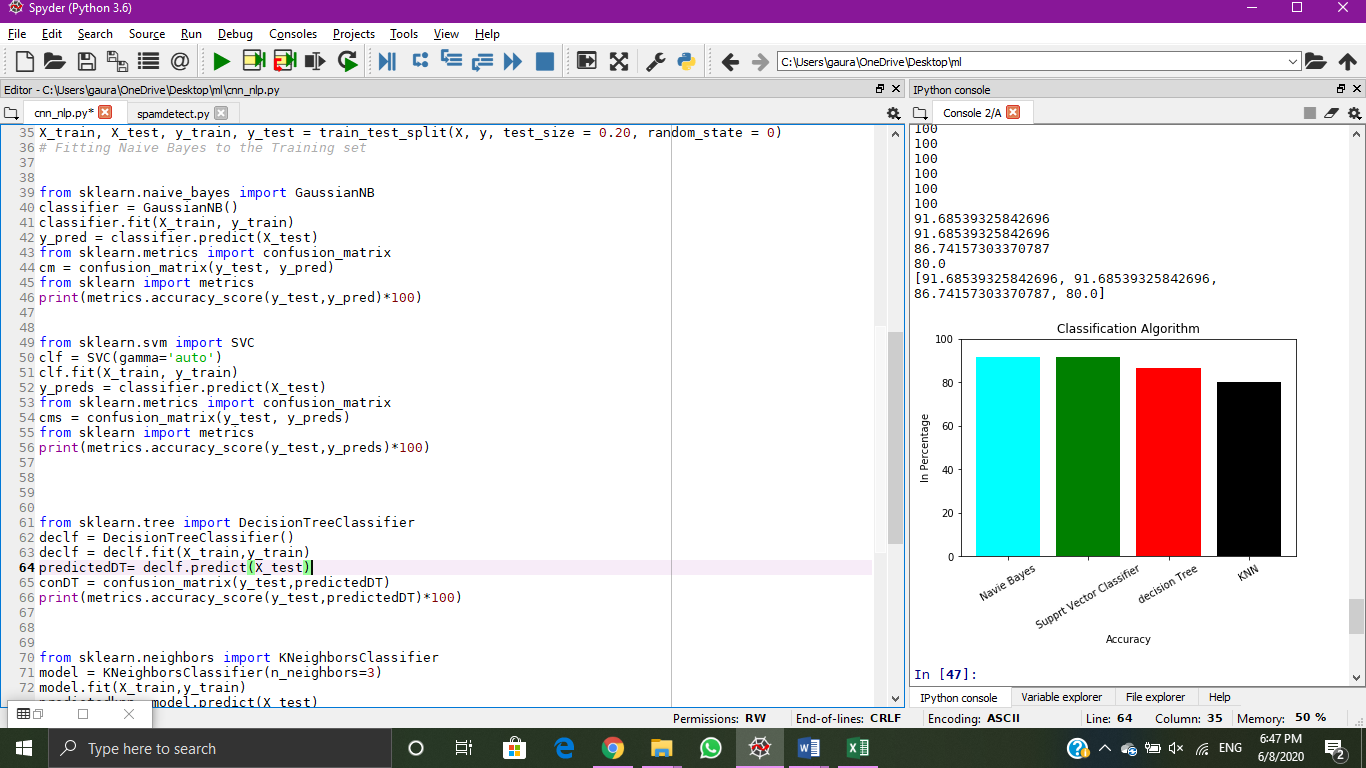


* **Naïve Bayes :  
   Creating classifier :**

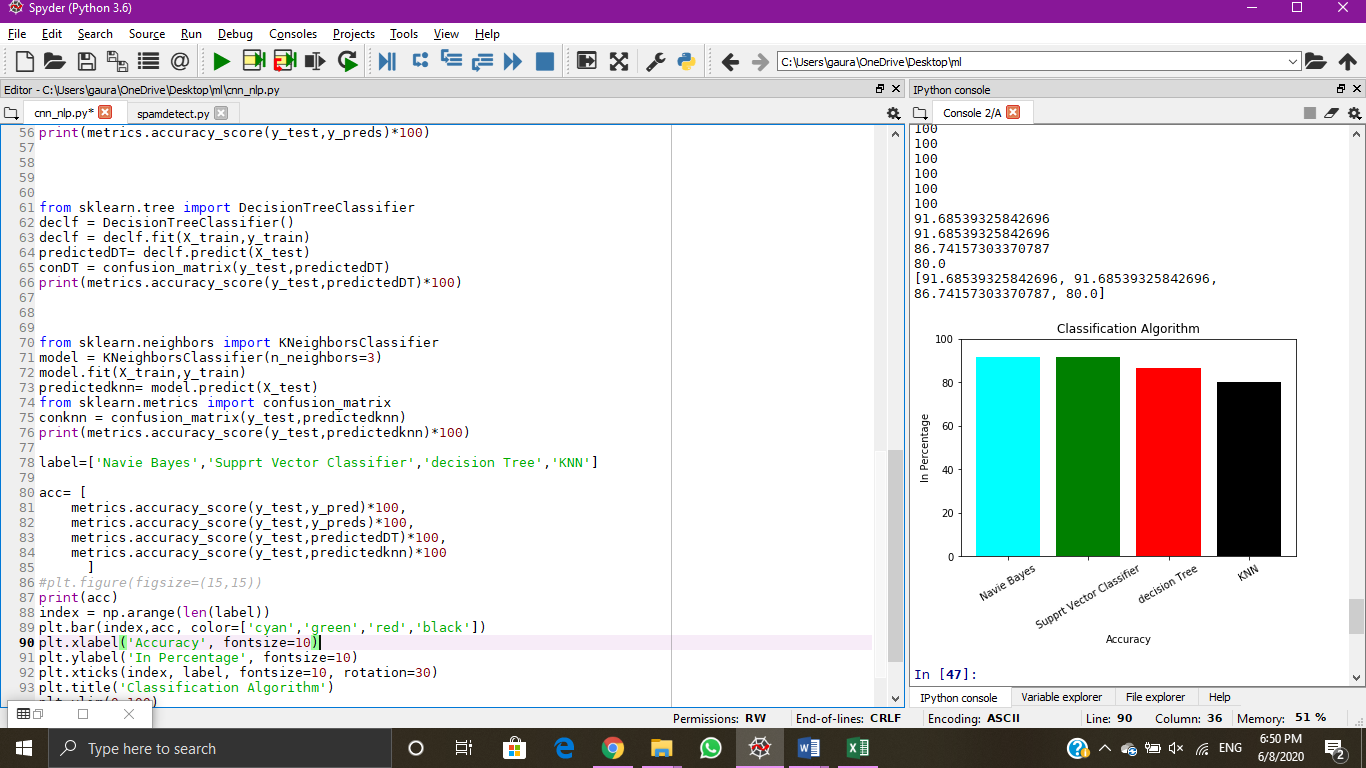


* **Support Vector Machine :**

**Creating Classifier model**

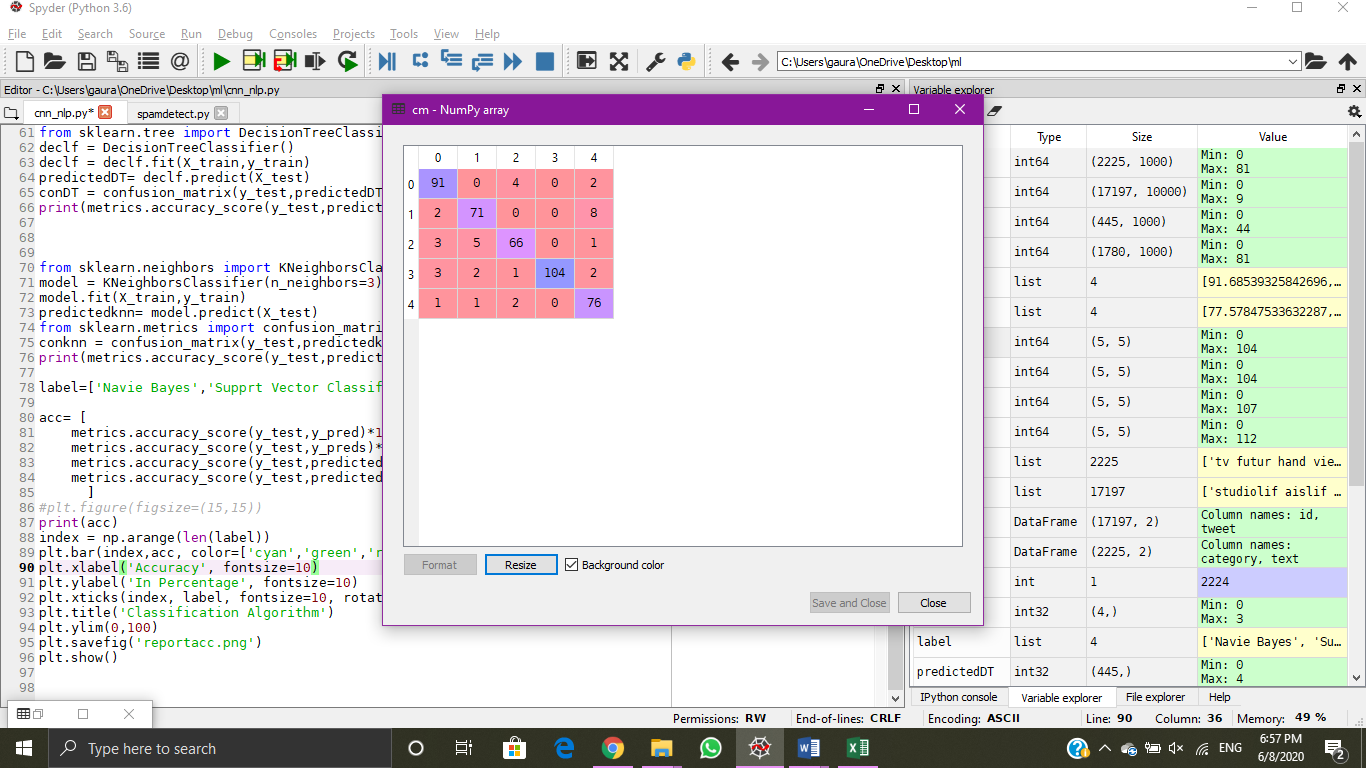
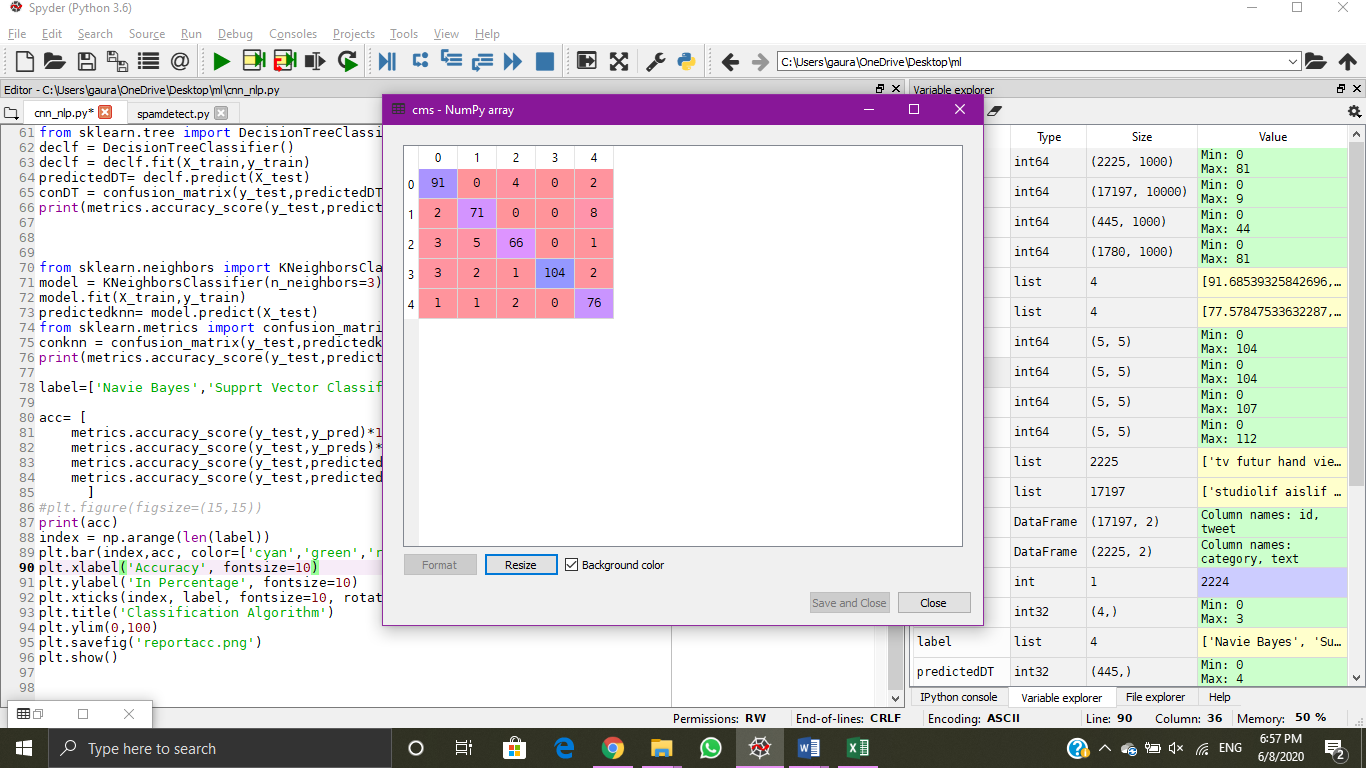


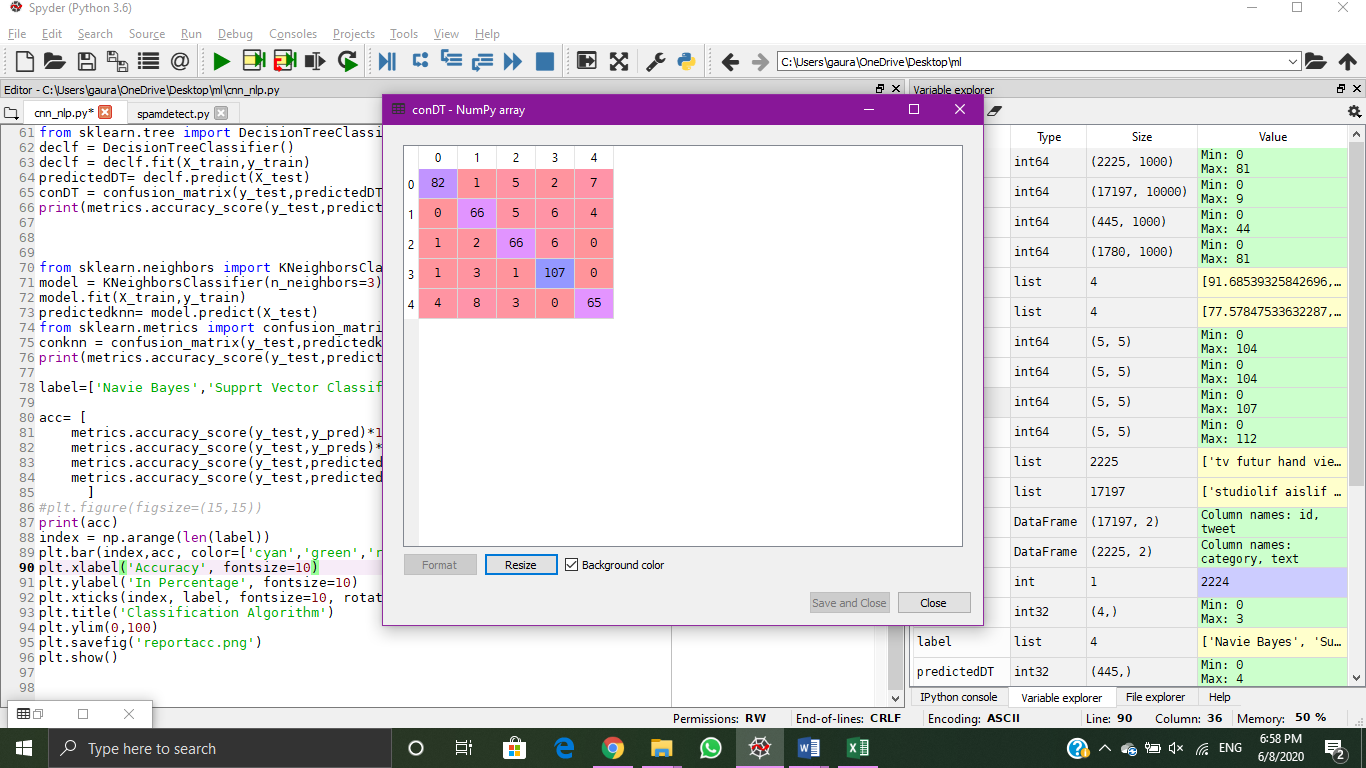
**8.Calculating Accuracy .**

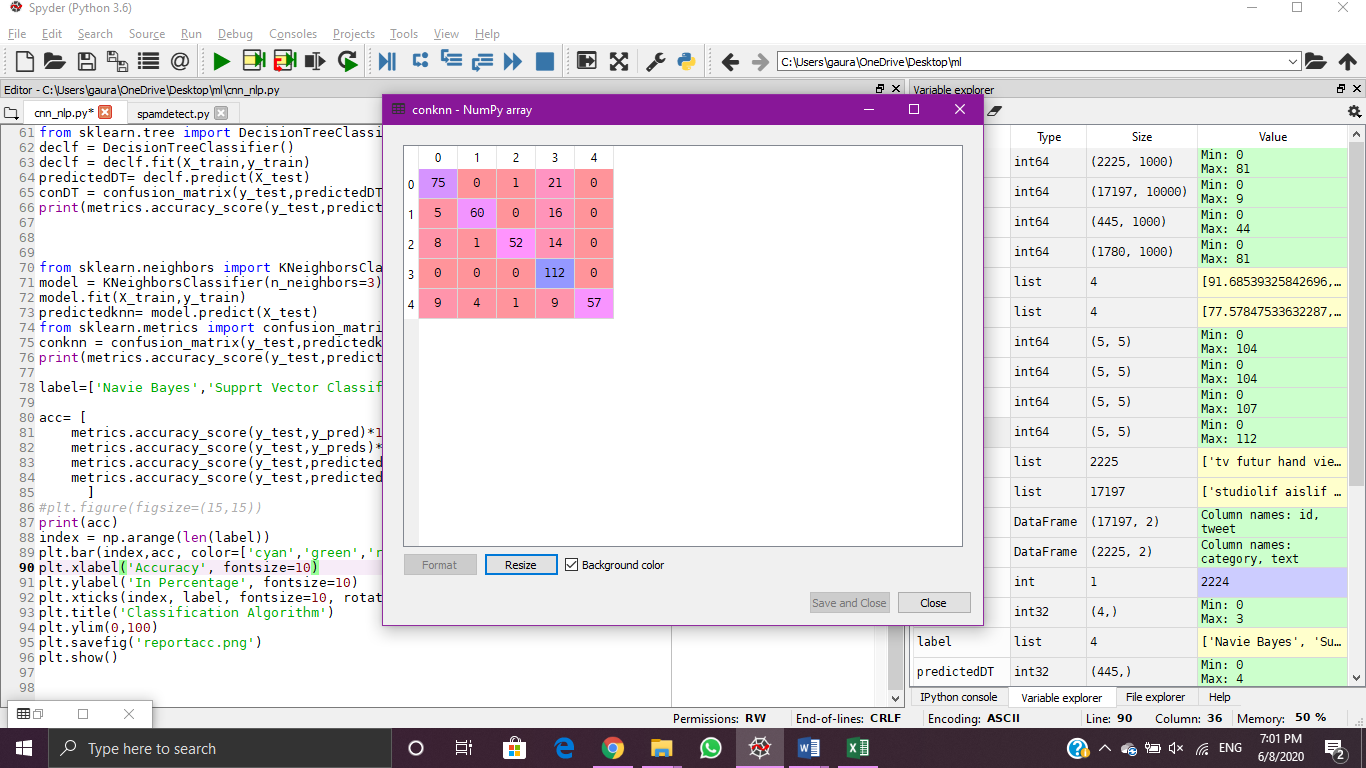


|  |  |
| --- | --- |
| **Method** | **Accuracy** |
| KNN | 80.00% |
| Decision Tree | 86.74% |
| Naïve bayes | 91.68% |
| SVM | 91.68% |

**9 Data Visualization and Results**

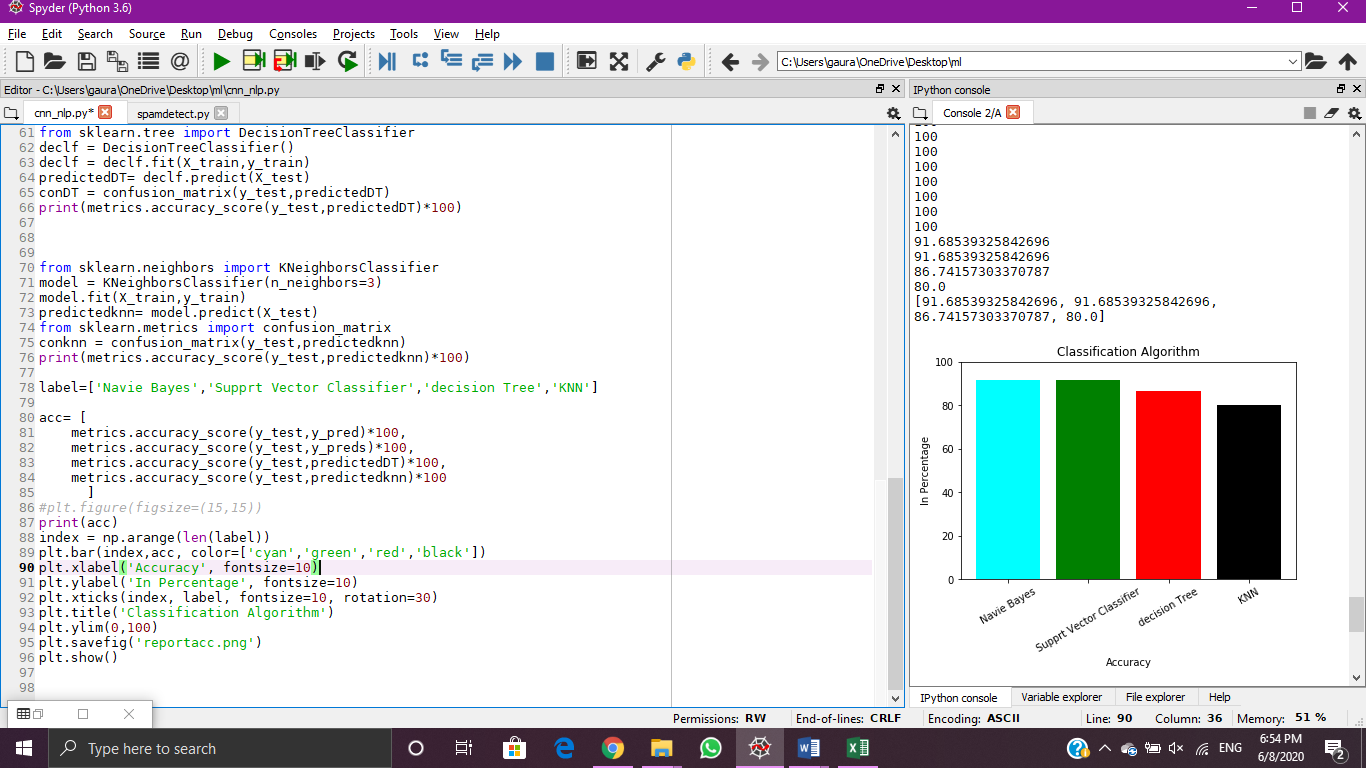
* **Data visualization of Confusion matrix**

** Naïve Bayes** **Support vector Machine Classifier**

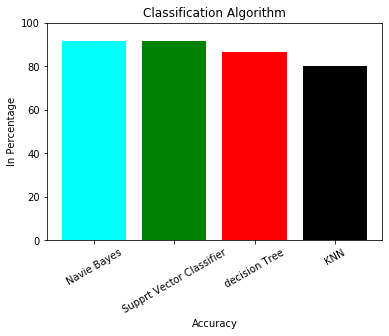


**Decision Tree** **K – Nearest Neighbors**

* **Data visualization for comparing classification algorithm**



**Output of the above code :**



**Conclusion:** In this way using python we have performed maching Learning on a CNN text

dataset and analyzed confusion matrix for different classifier models.