***Network Severity detection at a specific location based on log data available***

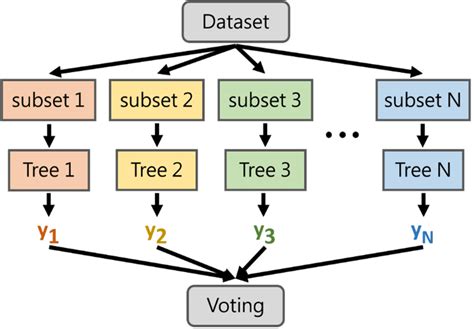
The goal of the problem is to predict the severity of a network's faults at a given time and location based on the log data available. Each row in the main dataset (train.csv, test.csv) represents a location and a time point. The "id" column, which is the key "id" used in other data files, identifies them. Using “ID” as a primary key we merge other files to make up a full dataset which then was used for data cleaning, Analysis and modelling. A lot of emphasis has been put on data analysis rather than data modelling since, identifying the issue and consequences of fault severity was very important.

***Base Paper***

1. <https://www.researchgate.net/publication/234074950_A_Review_of_Current_Practice_in_Network_Disruption_Analysis_and_an_Assessment_of_the_Ability_to_Account_for_Isolating_Links_in_Transportation_Networks>
2. <https://globalnetworkinitiative.org/policy-issues/network-disruptions/>

***Algorithm Description***

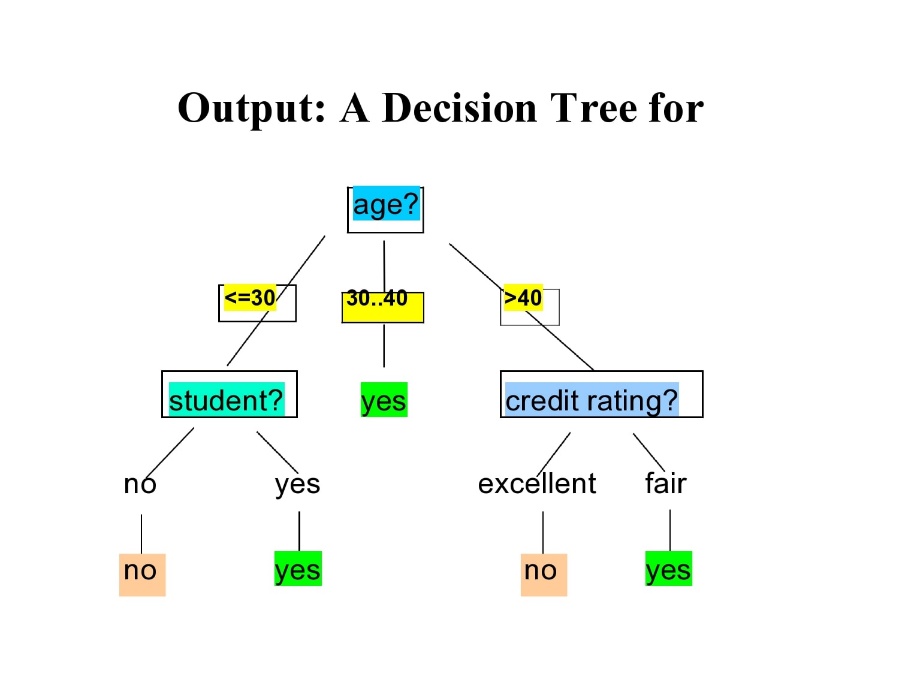
***Random Forest Classifier:*** Random Forest Classifier is an ensemble algorithm which works with multiple algorithms parallelly. This is a supervised algorithm and it can be used with both classification and regression problems. The output of the new data is estimated either by using majority voting or average voting technique. Since the algorithm works with bagging technique, multiple decision trees are used to provide the output for the specific input. This is a key difference between decision trees and random forests. While decision trees consider all the possible feature splits, random forests only select a subset of those features. Random forest works best with large datasets and high dimensional.



***References:***

1. <https://www.geeksforgeeks.org/random-forest-regression-in-python/>
2. <https://www.geeksforgeeks.org/random-forest-regression-in-python/>

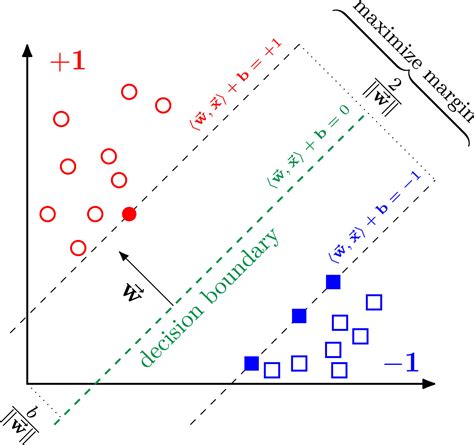
***Decision Tree Classifier:*** A decision tree is a tool for making decisions and the process for making decisions is in a tree like structure, decision tree is a supervised machine learning algorithm mainly used for predicting the outcome after computing all the attributes. The process flow of Decision tree goes from Root node to leave node i.e., the decision node.



***References:***

1. <https://www.ibm.com/topics/decision-trees>
2. <https://www.mastersindatascience.org/learning/introduction-to-machine-learning-algorithms/decision-tree/>

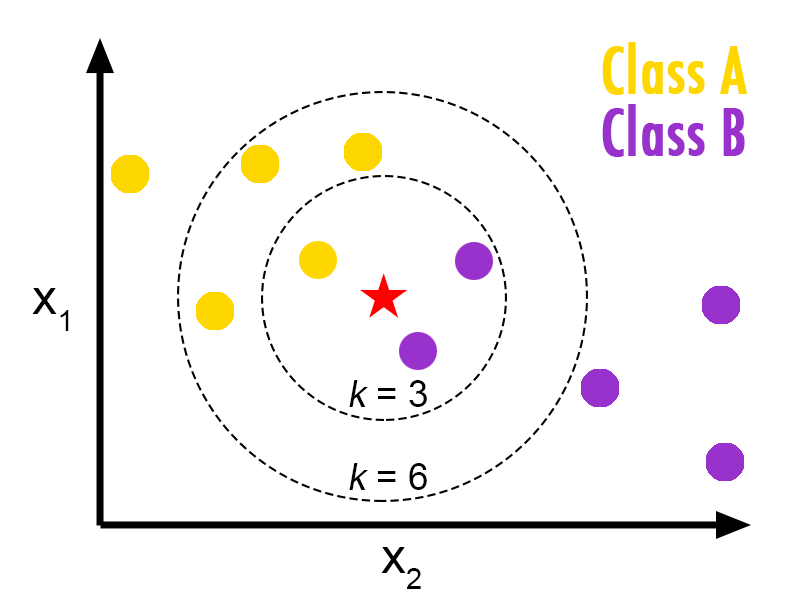
***Support Vector Machine:*** Support vector machines are basically a supervised learning algorithm which classifies the data points by drawing a linear curve and a non-linear curve depending on the data it is dealing with. The boundary that separated the 2 or more classes is called as a hyperplane, though there is a possibility of having some million hyperplanes for our data, but we need to find the hyperplane with maximum margin from all the training points, which makes the algorithm more efficient while predicting on new dataset, it can easily classify on which side the new data belongs to.



***References:***

1. <https://www.geeksforgeeks.org/support-vector-machine-algorithm/>
2. <https://scikit-learn.org/stable/modules/svm.html>

***Nearest Neighbour:*** KNN or K Nearest neighbours is a basic yet an efficient algorithm which is being used in most of the Machine learning application. Since it is a non-parametric i.e. This algorithm doesn’t make any underlying assumption like other algorithms do, such as having specify distribution of data to work with. So, this makes it very easy and understandable to all the users who are using it. The Technique KNN applies in predicting on new data is where it finds the nearest neighbours for the given point and takes a majority voting, whichever class is resided near to the new point, it will be considered as the new class for the new data point.



***References:***

1. <https://www.geeksforgeeks.org/k-nearest-neighbours/>
2. <https://www.ibm.com/topics/knn>

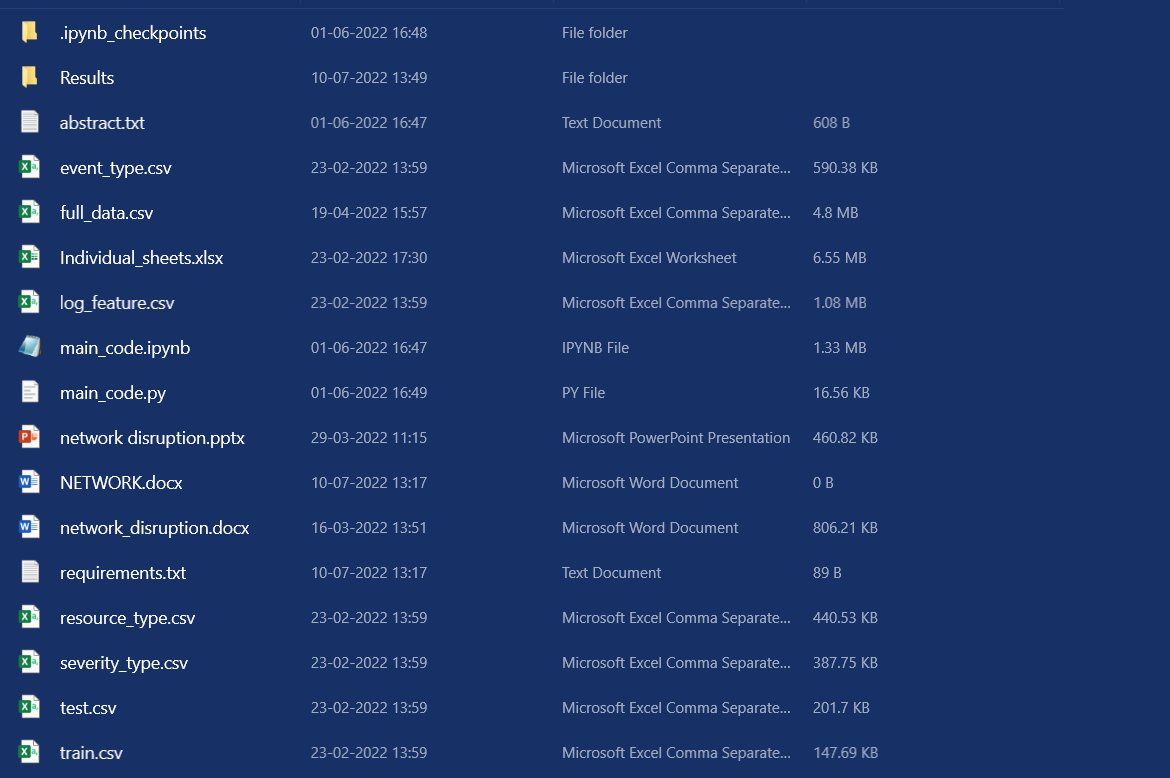
***Steps to Execute the Code!***

**Note:** Make sure you have added path while installing the software’s.

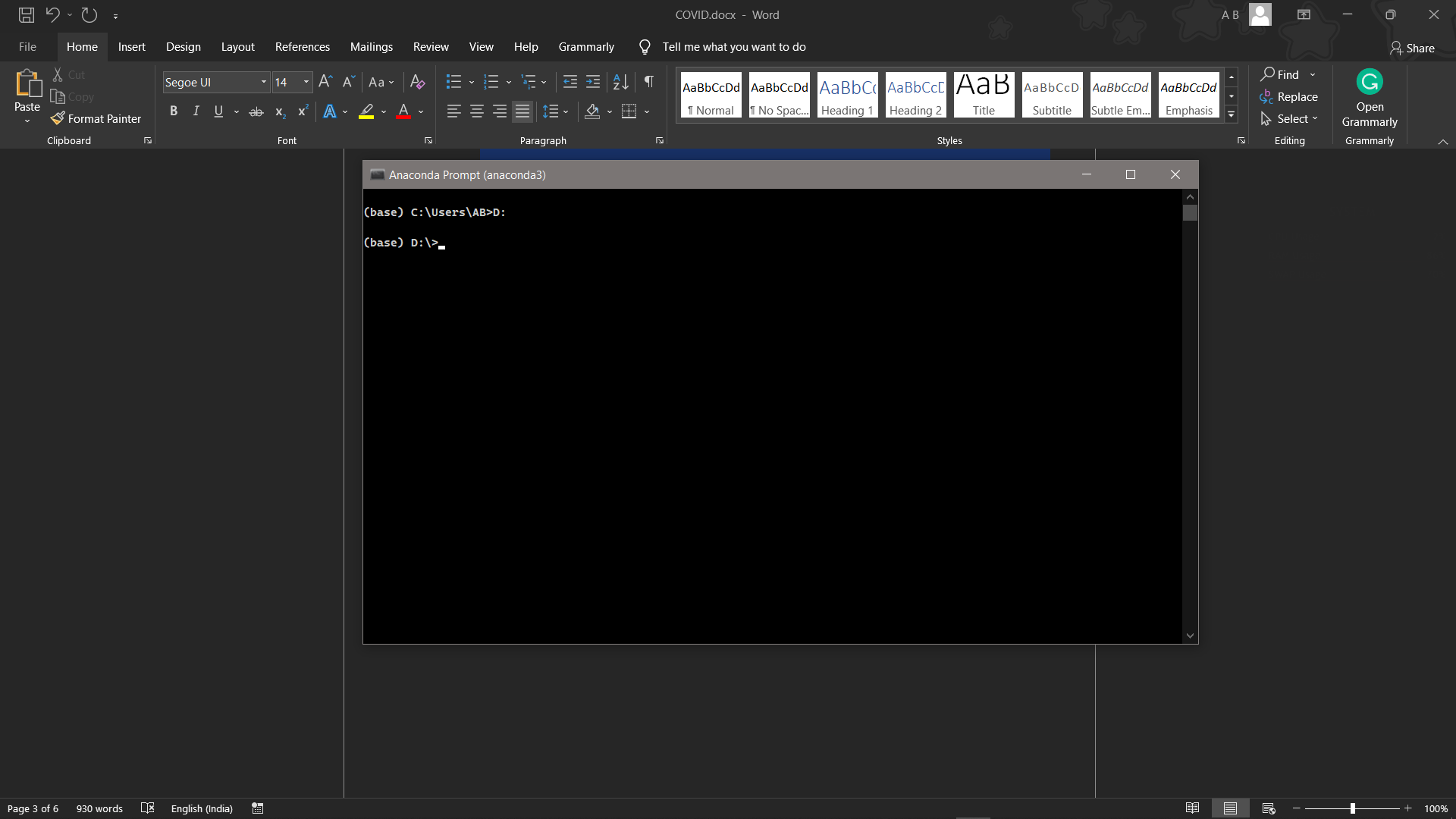
<https://techieyantechnologies.com/2022/07/how-to-install-anaconda/>

<https://techieyantechnologies.com/2022/06/get-started-with-creating-new-environment-in-anaconda-configuring-jupyter-notebook-and-installing-libraries-using-requirements-txt-2/>

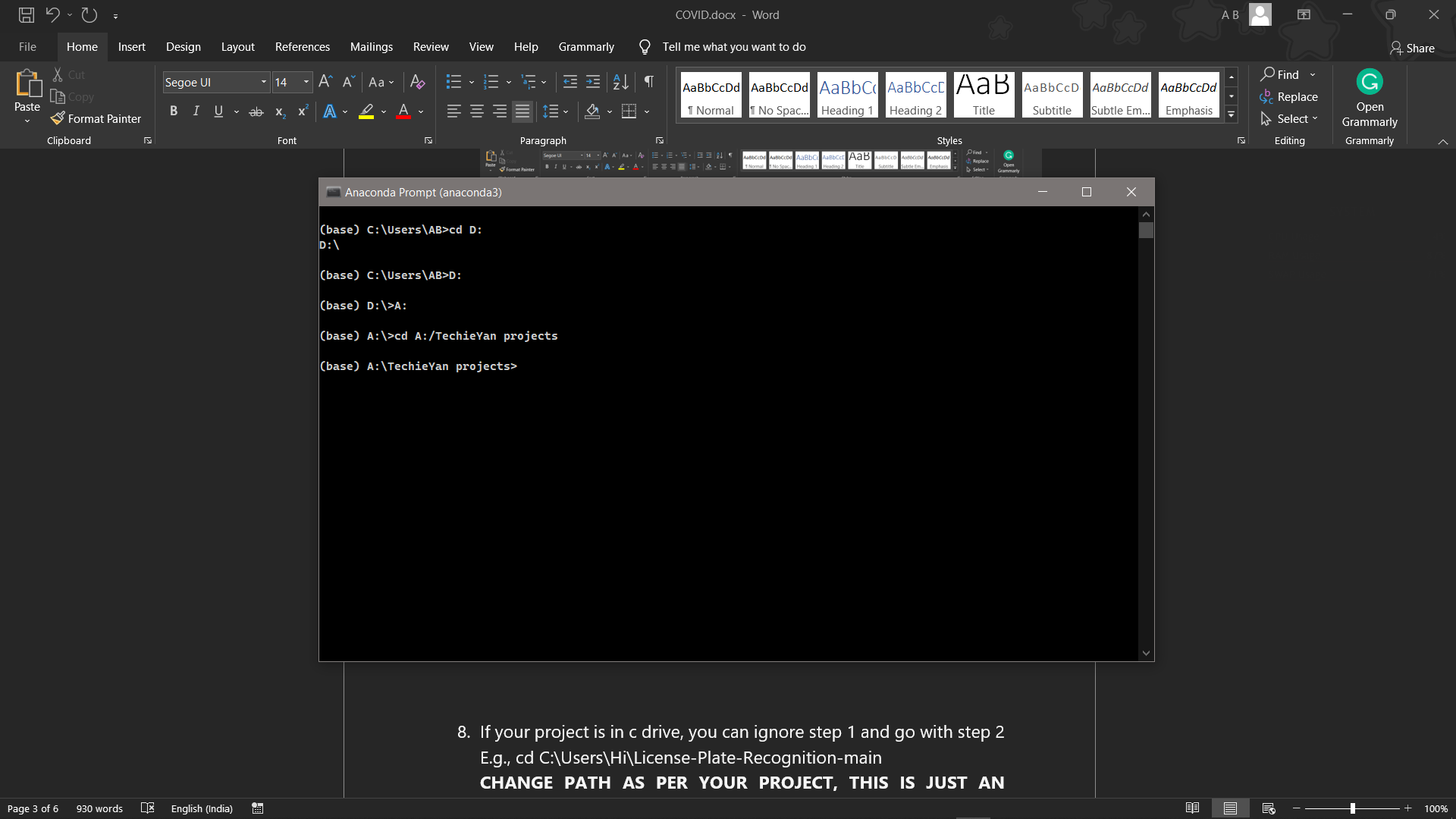
1. Install the prerequisites/software’s required to execute the code from reading the above blog which is provided in the link above.
2. Press windows key and type in anaconda prompt a terminal opens up.
3. Go to the directory where your requirement.txt file is present, not just requirement.txt, if you want to execute any .py or .ipynb files, you need to go to that specific folder or path, where they are saved.



1. <<directory of your file:>>. E.g., If my file is in d drive, then
2. Type **d:**



1. **cd d:\License-Plate-Recognition-main** #CHANGE PATH AS PER YOUR PROJECT, THIS IS JUST AN EXAMPLE

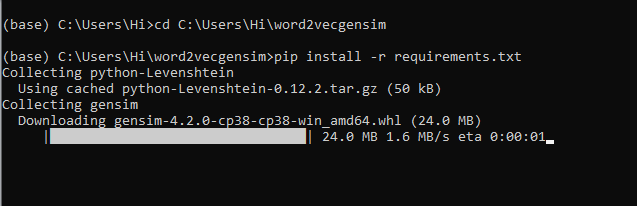


1. If your project is in c drive, you can ignore step 4 and 5.

E.g., **cd C:\Users\Hi\License-Plate-Recognition-main**

**CHANGE PATH AS PER YOUR PROJECT, THIS IS JUST AN EXAMPLE**

1. Run pip install -r requirements.txt or conda install requirements.txt (Requirements.txt is a text file consisting of all the necessary libraries required for executing this python file. If it gives any error while installing libraries, you might need to install them individually.), example: pip install “**module\_name**” i.e., pip install **pandas**



1. If you would like to run **main\_code.py**, please follow the above steps 4,5,6,7 and please check you are in the path where your executable file/folder is saved in your anaconda terminal and then type **python main.py**, before typing this please open the **main\_code.py** and change the path of the dataset.
2. If you would like to run **main\_code.ipynb** file, After the above steps type jupyter notebook in your terminal, You will be redirected to the local server there you can select which ever .ipynb file you’d like to run and click on it and execute each cell one by one by pressing shift+enter, Please [follow the link to setup and open jupyter notebook](https://techieyantechnologies.com/2022/06/get-started-with-creating-new-environment-in-anaconda-configuring-jupyter-notebook-and-installing-libraries-using-requirements-txt-2/), If you face any trouble opening jupyter notebook.

**Make sure to change the path of the dataset in the code**

***Dataset Description***

The Dataset is collected form Kaggle Repository which contains 61840 Instances with 8 features. Multiple csv files have been merged together to work on one single mission. They have been merged by ID as a primary Key. Features with high correlation coefficient are Severity type, log\_feature, location, event-type and volume. The target column is represented as fault\_severity which ranges from 1,2,3 with 1 being minimum and 3 being maximum. Data balancing techniques have been incorporated since the data was quite imbalanced from 1 class been dominated by other.



***Issues Faced***

1. We might face an issue while installing specific libraries, in this case, you might need to install the libraires manually. Example: pip install “module name/library” i.e., pip install **pandas**

2. Make sure you have the latest or specific version of python, since sometimes it might cause version mismatch.

3. Adding path to environment variables in order to run python files and anaconda environment in code editor, specifically in any code editor.

4. Make sure to change the **paths in the code** accordingly where your dataset/model is saved.

**Refer to the Below links to get more details on installing python and anaconda and how to configure it.**

<https://techieyantechnologies.com/2022/07/how-to-install-anaconda/>

<https://techieyantechnologies.com/2022/06/get-started-with-creating-new-environment-in-anaconda-configuring-jupyter-notebook-and-installing-libraries-using-requirements-txt-2/>

***Note:***

**All the required data has been provided over here. Please feel free to contact me for model weights and if you face any issues.**

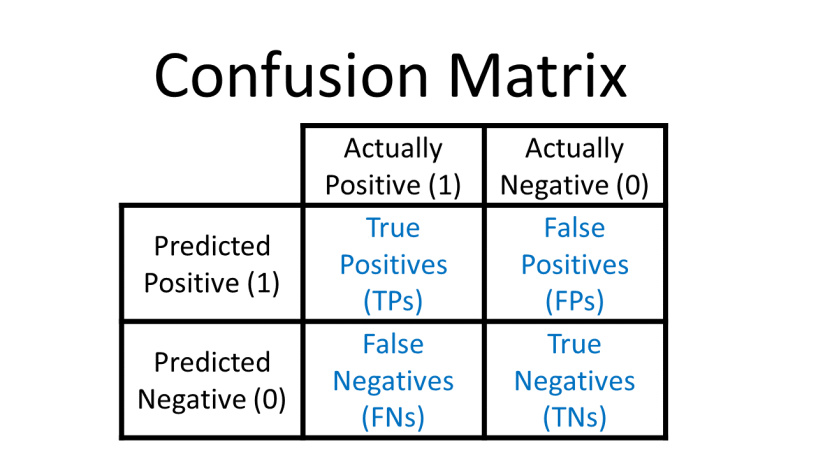
<https://www.linkedin.com/in/abhinay-lingala-5a3ab7205/>

***Yes, you now have more knowledge than yesterday, Keep Going.***

***Evaluation Metrics***

Evaluation metrics are considered as one of the most important steps in any machine learning and deep learning projects, where it will allow us to evaluate how good our model is performing on the new data or on unseen data. There are a lot of evaluation metrics which can be used in order to assess how good our model is performing such as roc\_auc\_curve, f1\_score, recall, precision and each of which work for specific problem we deal. So, for our project we have gone with confusion matrix and classification report which helps us to evaluate not just the accuracy of the model but also the other metrics such as precision, recall and f1\_score.

***Confusion matrix:***



***Classification Report:*** Classification report helps us to understand and evaluate how good the model is performing which consists of different evaluation metrics, such as

**Precision:** This will count number of correct predictions of a single class divided by the total number of observations of the same class.

Precision = TP/(TP + FP)

**Recall:** Recall is calculated by counting the actual number of classes in a single class divided by total number of observations of that particular class.

Recall = TP/(TP+FN)

F1\_score: This is a Harmonic average of Precision and recall.

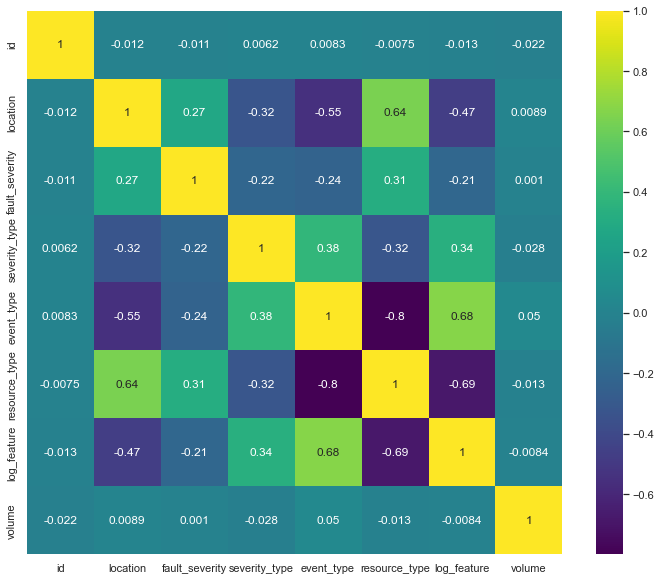
F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

***Reference:***

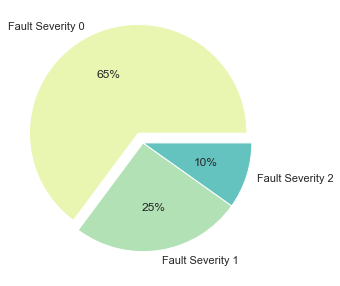
1. <https://scikitlearn.org/stable/modules/generated/sklearn.metrics.classification_report.html>
2. <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/>

***Exploratory Data Analysis***

1. ***Correlation Heatmap***



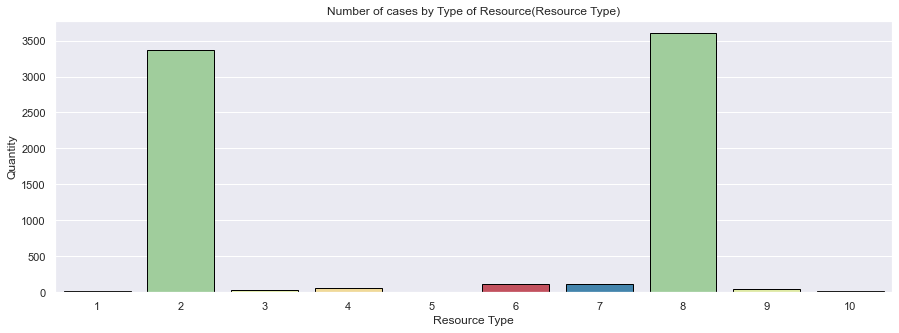
1. ***Pie Chart***



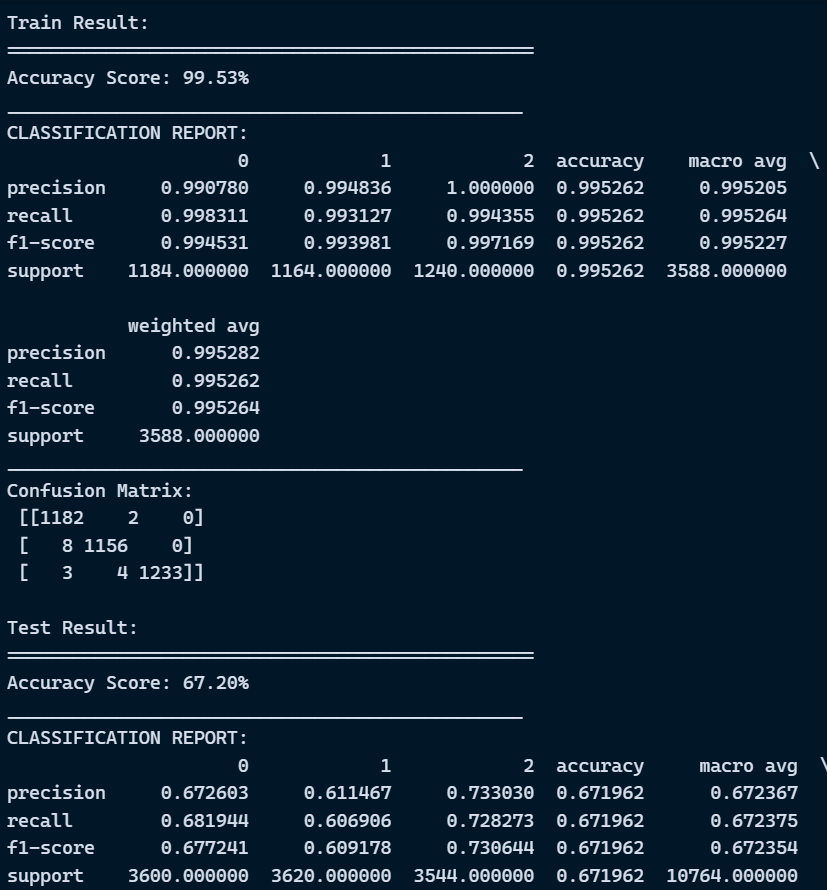
1. ***Pair Plot***



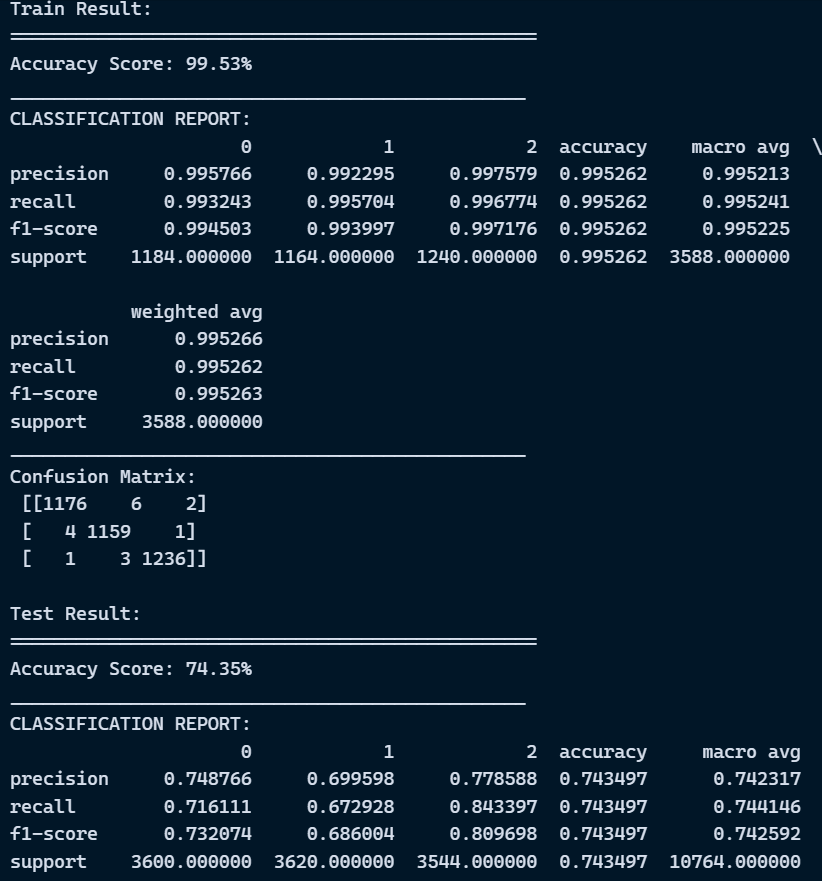
1. ***Bar chart of Resources available***



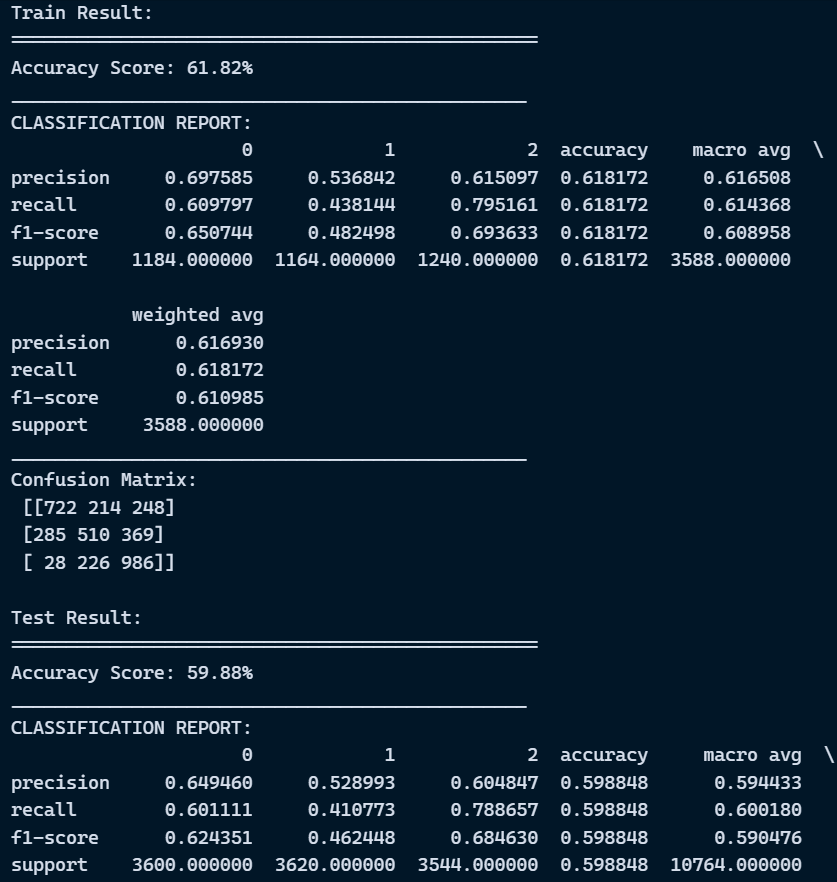
1. ***Decision Tree***



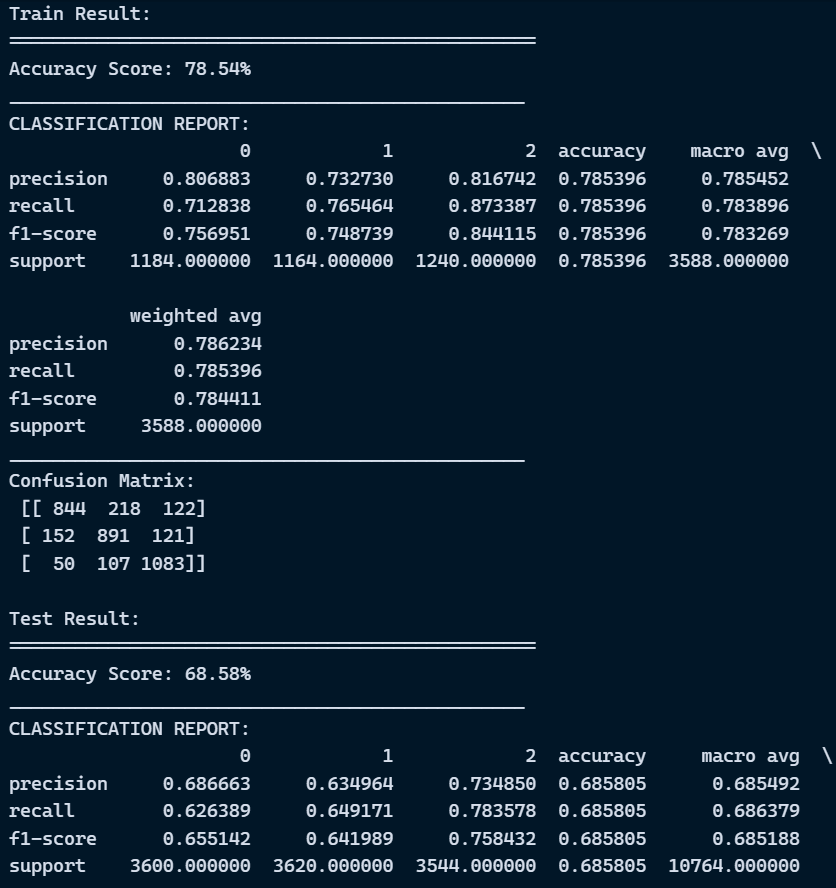
1. ***Random forest***



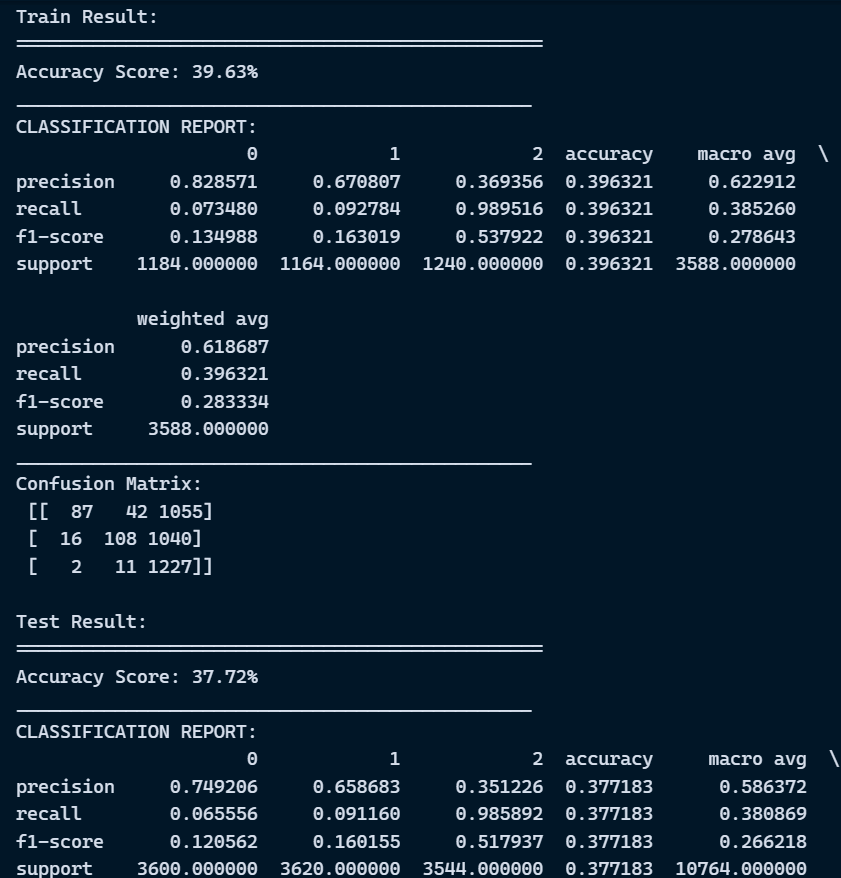
1. ***Support vector machine***



1. ***KNearest Neighbours***

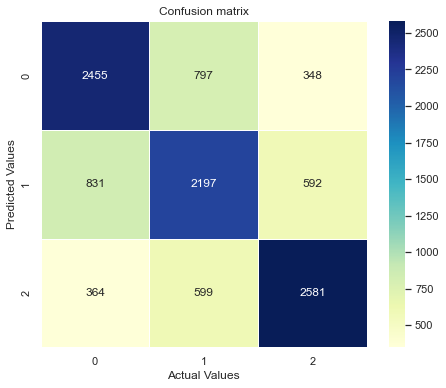


1. ***Gaussian NB***

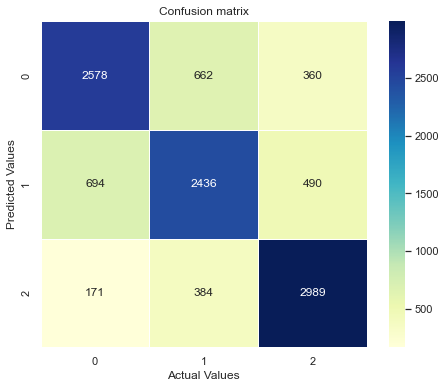


***Results***

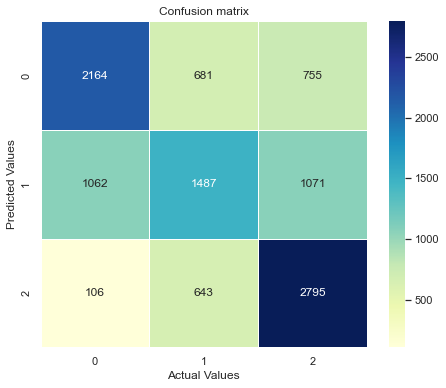
1. ***Decision tree Confusion matrix***



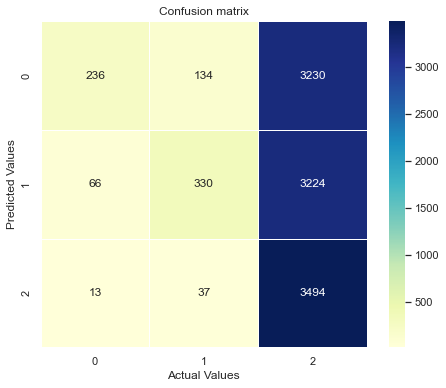
1. ***Random forest Confusion matrix***



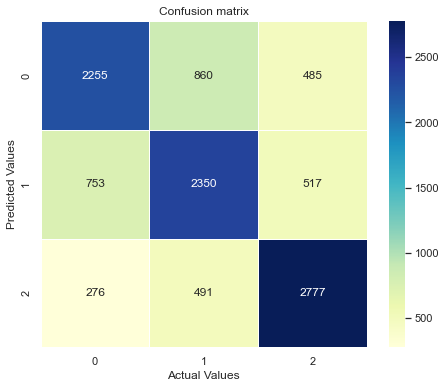
1. ***Support Vector Machine Confusion matrix***



1. ***Gaussian Confusion matrix***



1. ***KNN Confusion matrix***



1. ***Improved Model(RF) Confusion matrix***

