**Network IntrusionDetection**

**--- By Pallavi Yadkikar**

**Problem Statement :**

In this project, we aim to detect the network intrusions and predict where it is bad or good connection using different sklearn Models, Neural Network and CNN.

**Methodology :**

* **Converted a CSV file into a Dataframe.**



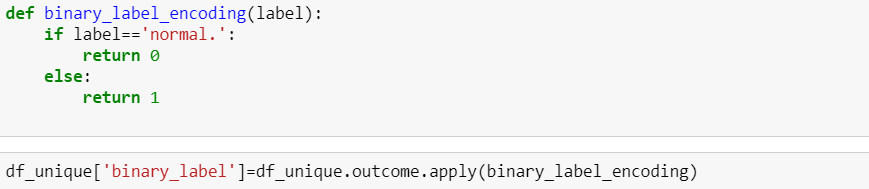
* Removed rows with any NULL values.



* Removed rows with any NULL values



* Performed Binary Classification (i.e 0-Good connection and 1-Bad connection)



* Performed one hot encoding for the columns : protocol\_type, service, flag and su\_ attempted.



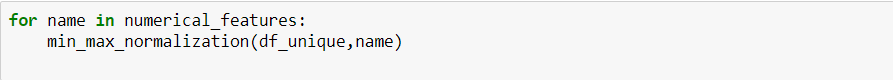




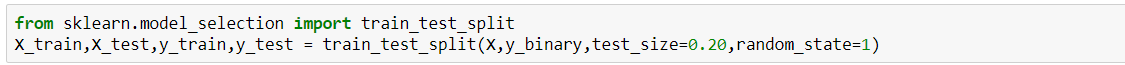


* Performed min max normalization for the numeric features





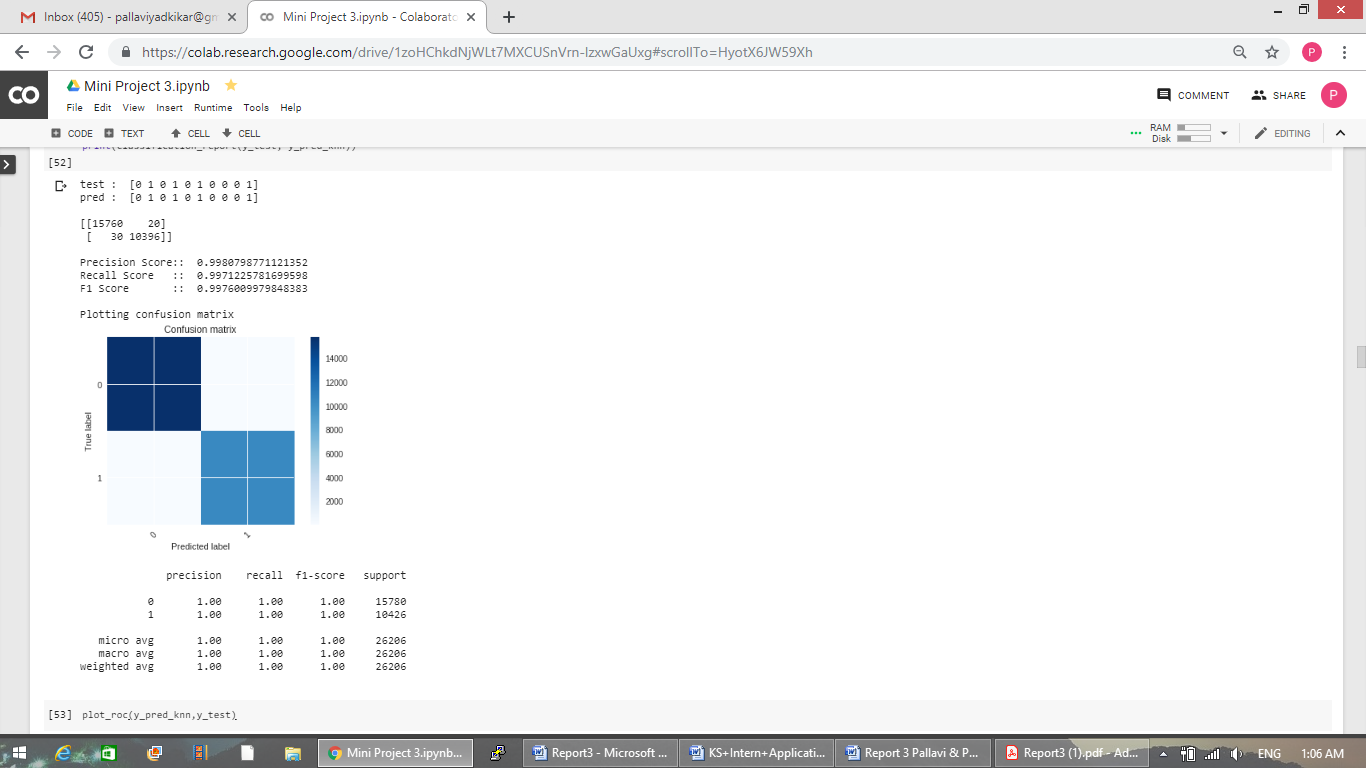
* Deleted is\_host\_login column and num\_outbound\_cmds column as the values is these columns have one unique value and by removing these columns will not change the results of the output.
* Splitted the data into Train and Test.

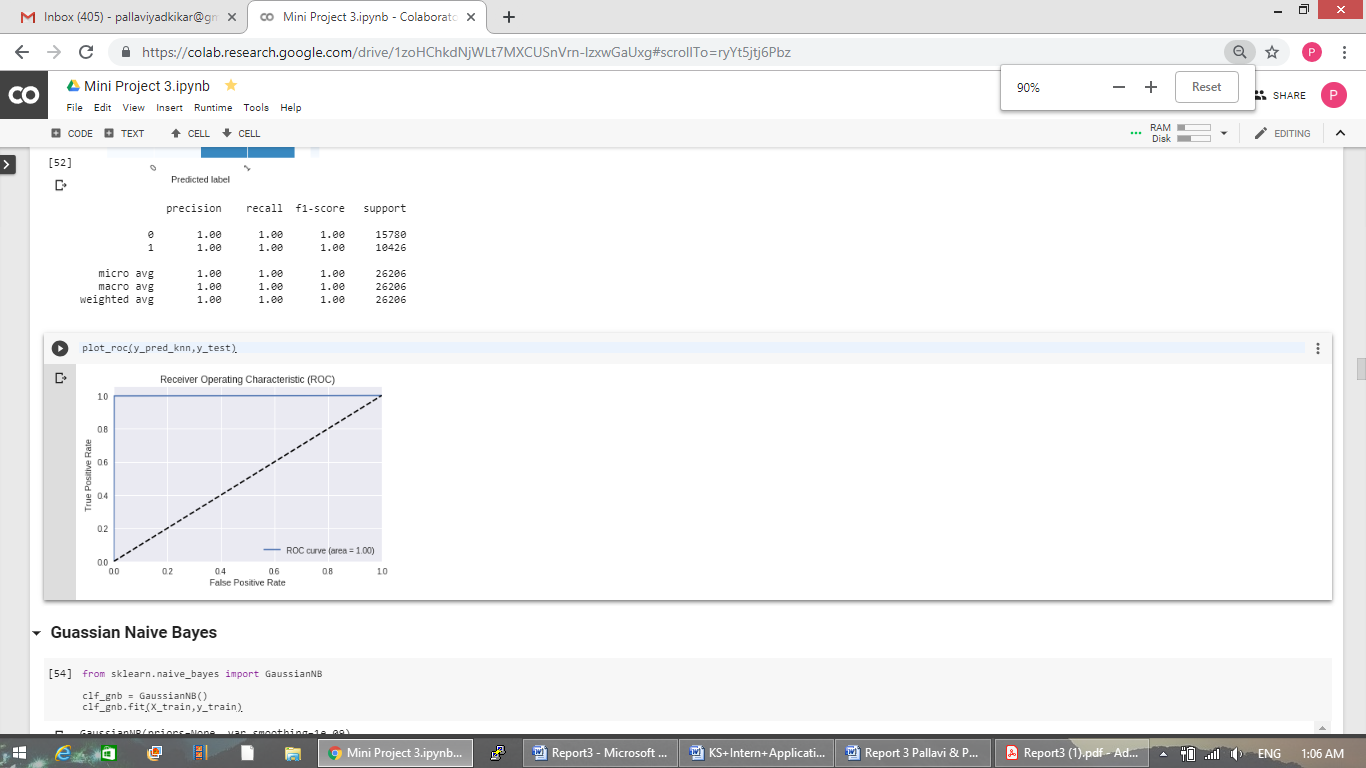


* Trained all the models: SVM, KNN, Logistic Regression, Gaussian Naïve Bayes, Neural Networks and CNN(Covolutional Neural Networks).
* Predicted all models with their F-1scores, Confusion matrix and ROC curve.

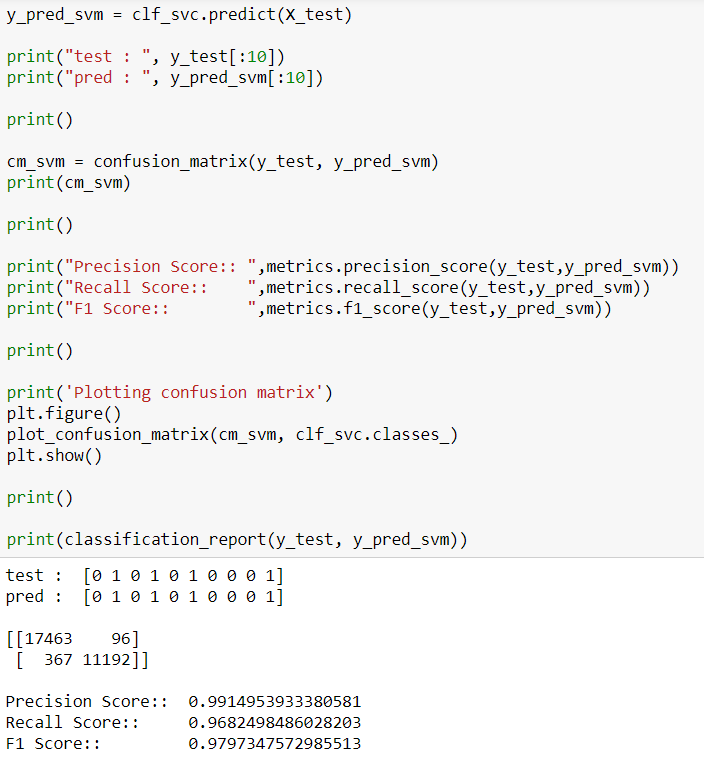
**Experimental Results and Analysis**

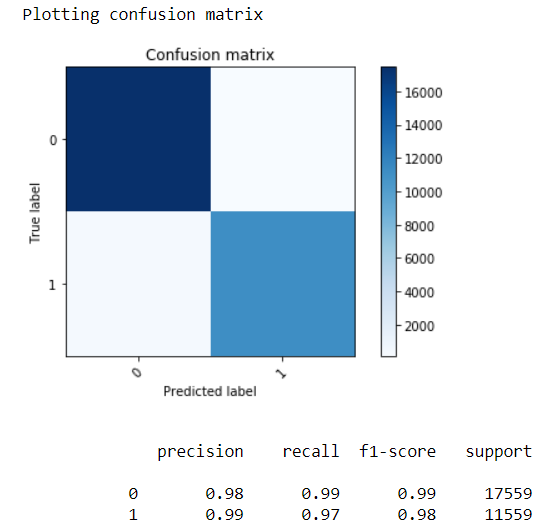
**1. KNN**

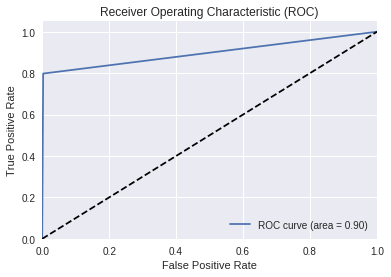
****

****

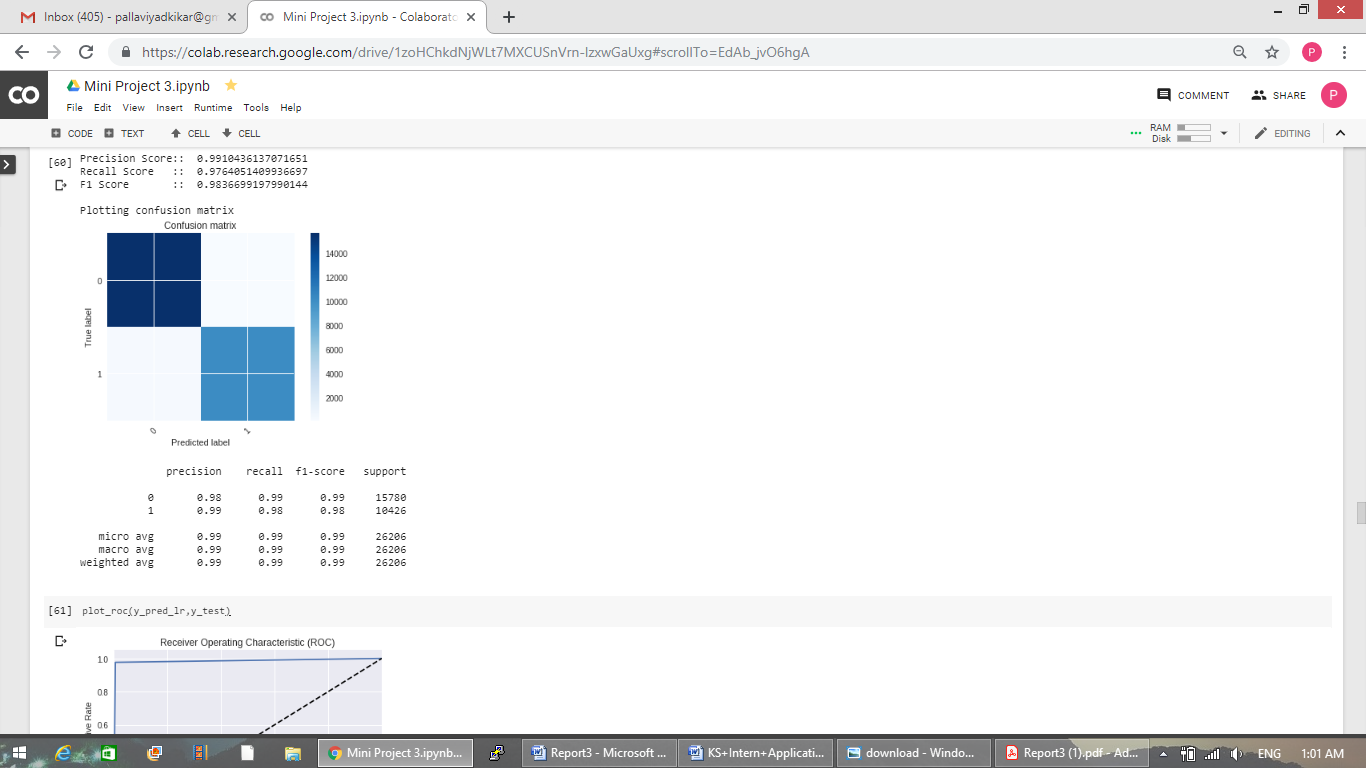
**2. SVM**

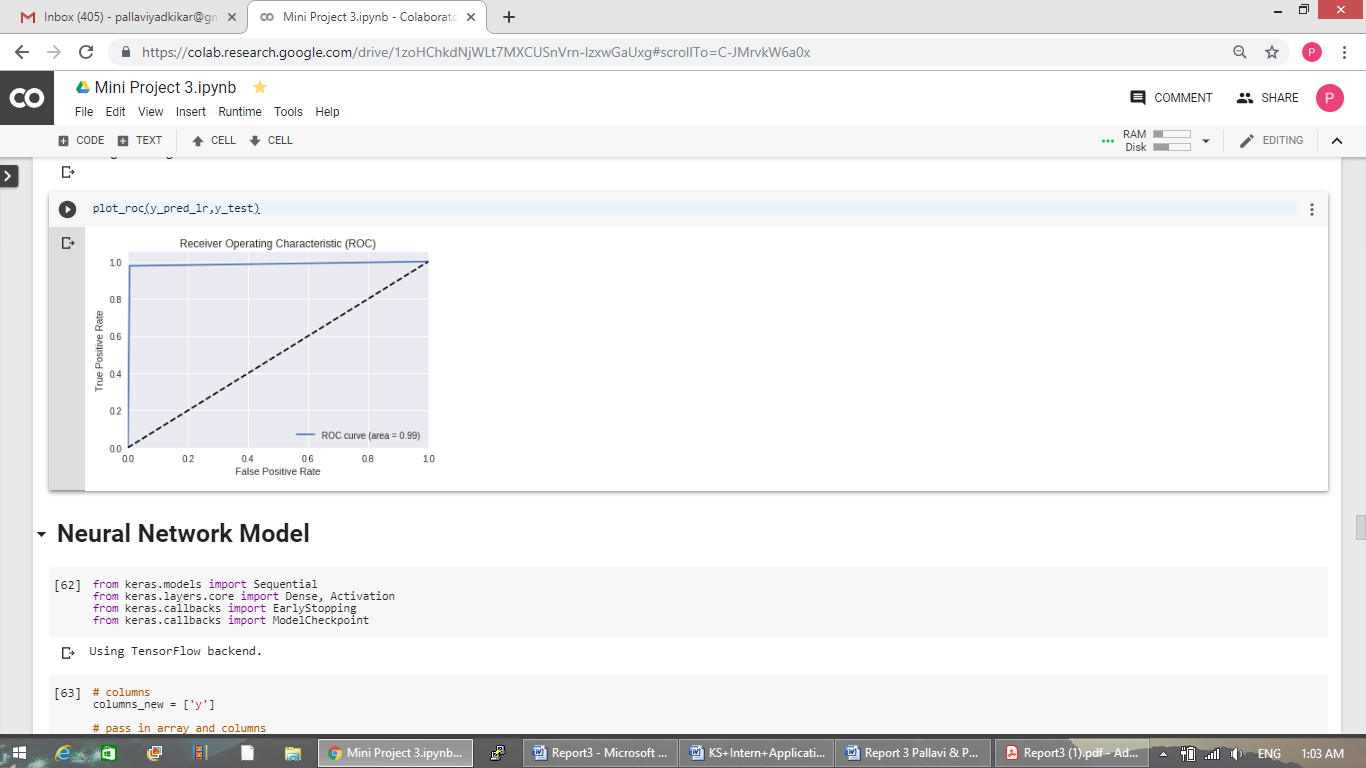




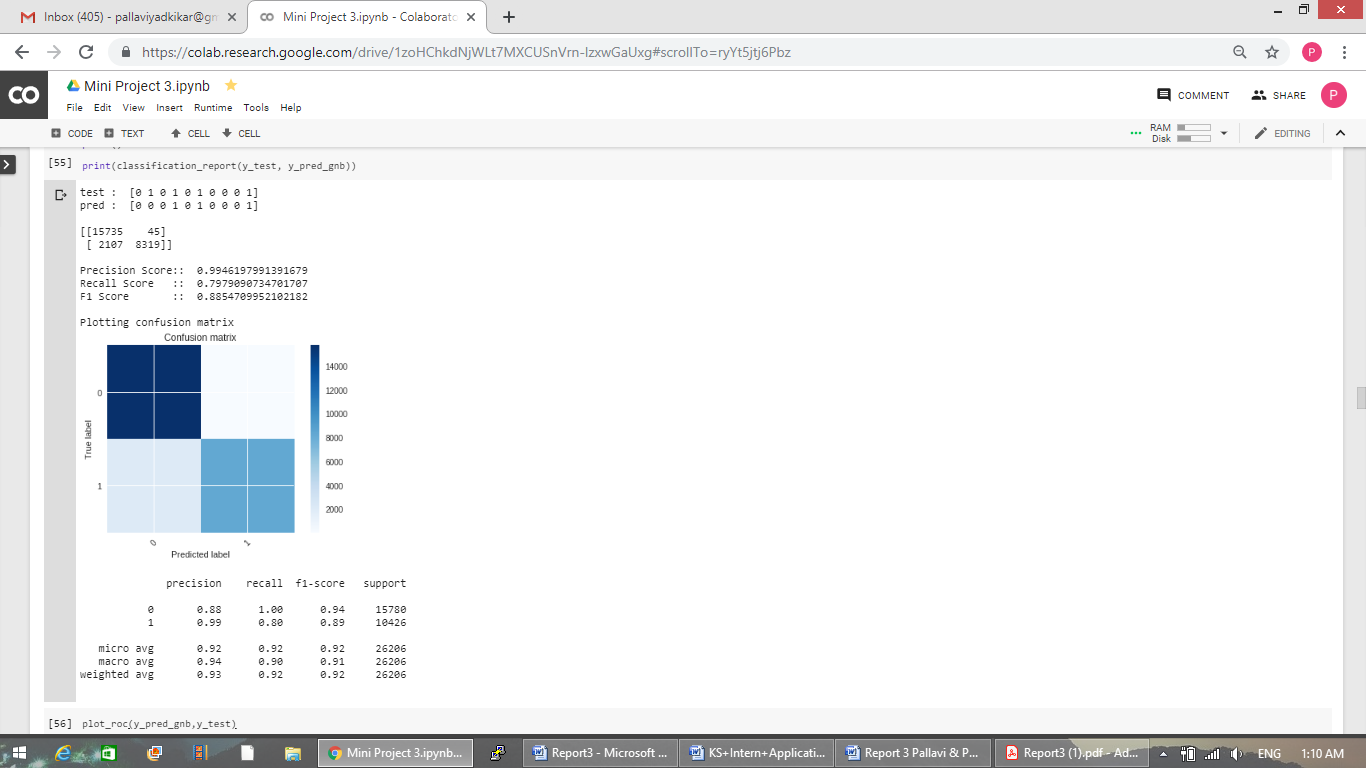
****

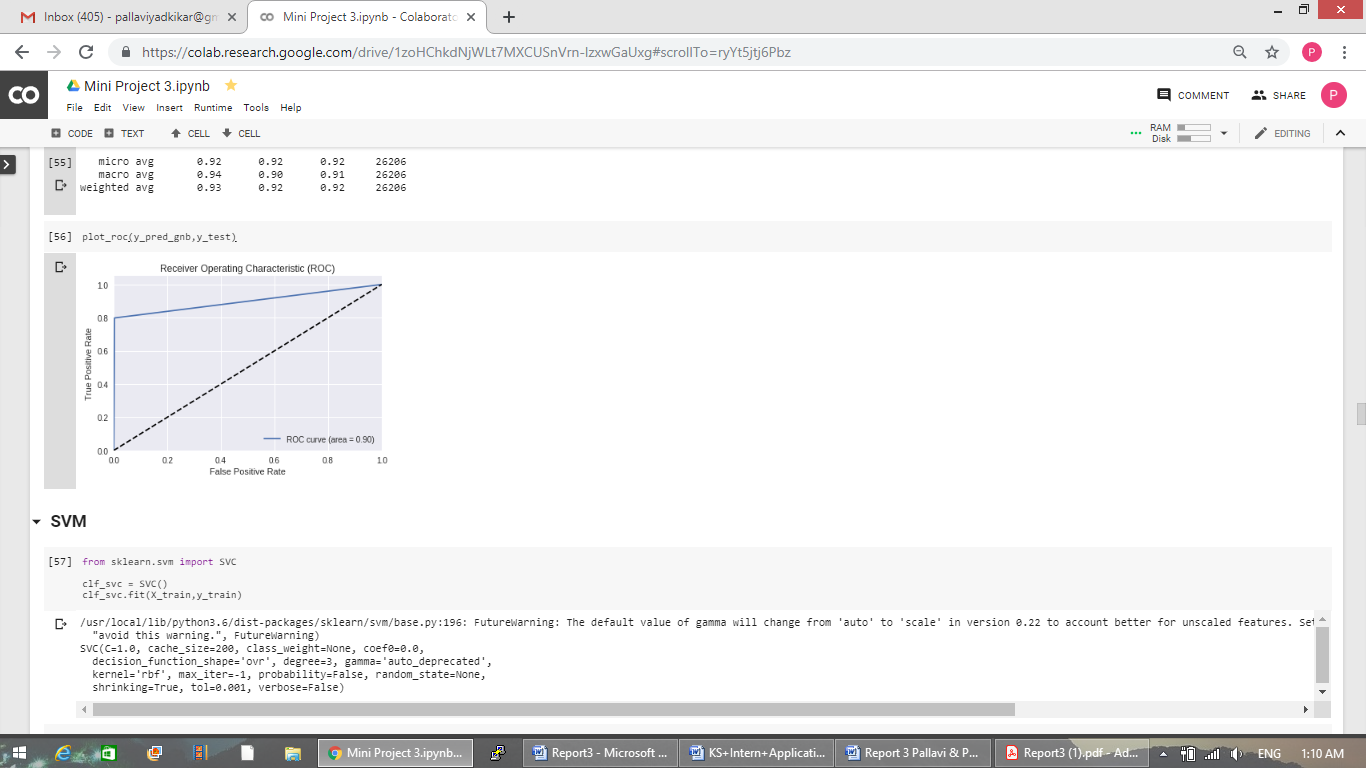
**3. Logistic Regression**

****

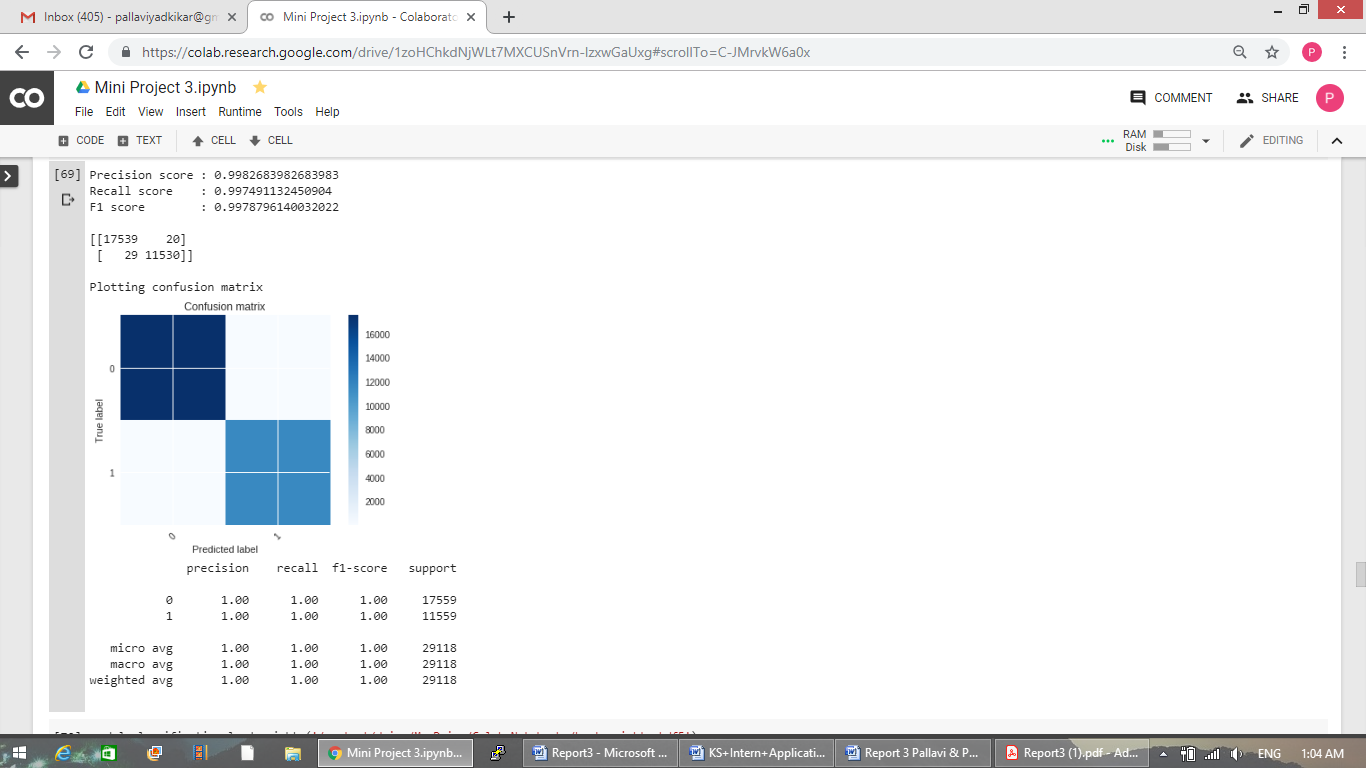
****

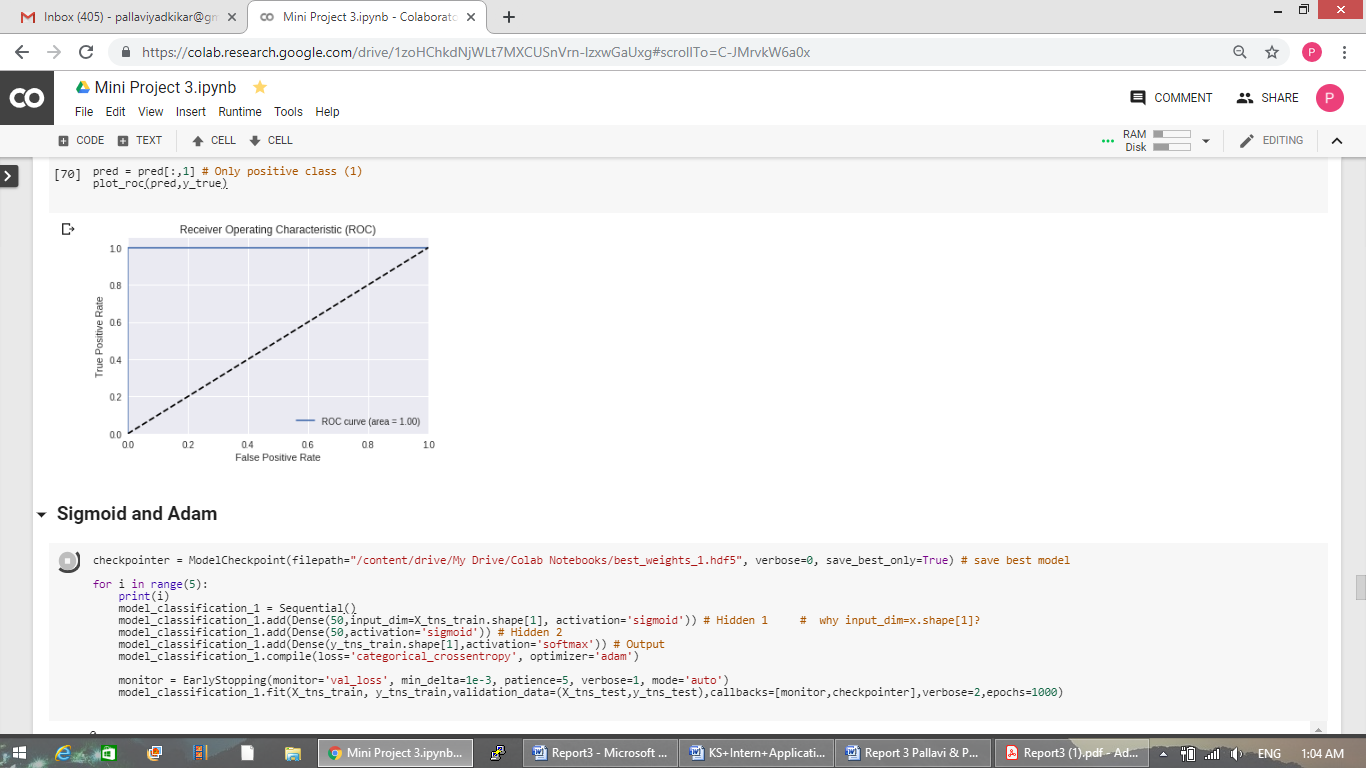
**4. Gaussian Naïve Bayes**

****

****

**5. Neural Network**

****

****

**6. CNN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classification Models Analysis** | | | |  |
| **Model** | **F1 score** | **Precision score** | **Recall Score** | **Accuracy** |
| **SVM** | 0.9836 | 0.9868 | 0.9873 | 0.9840 |
| **KNN** | 0.9975 | 0.9973 | 0.9977 | 0.9980 |
| **Logistic Regression** | 0.9867 | 0.9887 | 0.9893 | 0.9871 |
| **Gaussian Naïve Bayes** | 0.5498 | 0.9710 | 0.4597 | 0.9178 |
| **Best Neural Network model in Tensorflow (Tanh with Adam)** | 0.9984 | 0.9976 | 0.9980 | 0.9984 |
| **Best Convolutional Neural Network model in Tensorflow(Relu and SoftMax)** | 0.9924 | 0.9852 | 0.9887 | 0.9957 |

**Task Division and Project Reflection :**

**Pallavi Yadkikar :**

Worked on CoLab.

Normalized Numeric features using min max normalization.

Split the data into train and test data.

Implemented 2 models (Support Vector Machine, Logistic Regression)

Prediction for the Test data and compared actual and predicted result.

ROC Curve for each model.

Convolutional Neural Network and Parameter Tuning for CNN (Kernel size, Kernel no. and Strides)

**Palak Patel :**

Worked on System and Complete Data Set

Removed rows with any null values.

Removed duplicate rows

Encoded Categorical features

Implemented 2 models (Nearest Neighbor, Gaussian Naïve Bayes).

Confusion Matrix for each model.

Neural Network and the Parameter Tuning for Classification Models.

**Additional Features :**

1. Performed multi-class classification with all 23 types of intrusions.
2. Below is the analysis for multi-class classification for all sklearn models and Neural Network.
3. Detail Insights of model which can be seen in the output and code submitted with the report.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Multiclass Classification Models Analysis** | | | |  |
| **Model** | **F1 score** | **Precision score** | **Recall Score** | **Accuracy** |
| **SVM** | 0.9797 | 0.9682 | 0.9914 | 0.9873 |
| **KNN** | 0.9934 | 0.9964 | 0.9705 | 0.9977 |
| **Logistic Regression** | 0.9867 | 0.9887 | 0.9893 | 0.9893 |
| **Gaussian Naïve Bayes** | 0.8854 | 0.9946 | 0.7979 | 0.4597 |
| **Best Neural Network model in Tensorflow(Relu with Adam)** | 0.9980 | 0.9979 | 0.9982 | 0.9982 |

**Parameter Tuning:**

**Analysis:**

1. How to do feature normalization (min-max scaling).
2. Numpy , pandas and various operations on numpy array and dataframe
3. Applying the models and generating their scores and comparing their performance.
4. How to apply Gaussian Naïve Bayes Model.
5. How to implement neural network using tensorflow and keras.
6. How to use Early stopping and Save and Use saved best weights of Neural Networks.
7. Parameter Tuning of Neural Networks like optimizer , no. of hidden layers , no. of neurons in each layer and different activation functions.
8. How to implement Convolutional Neural Networks.
9. Parameter Tuning for CNN(Kernel size, Kernel no. and Strides)