VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Title: Fake News Detection using machine learning approaches

CSE 3501 INFORMATION SECURITY MANAGEMENT PROJECT REVIEW 2

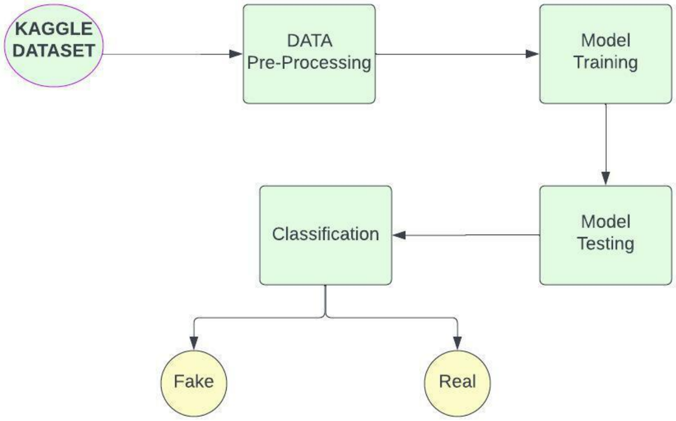
Team Members

MUGILAN - 19BIT0102
PIYUSH KUMAR – 19BIT0089
MAHALPURE PRANAV SUNIL – 19BIT0109

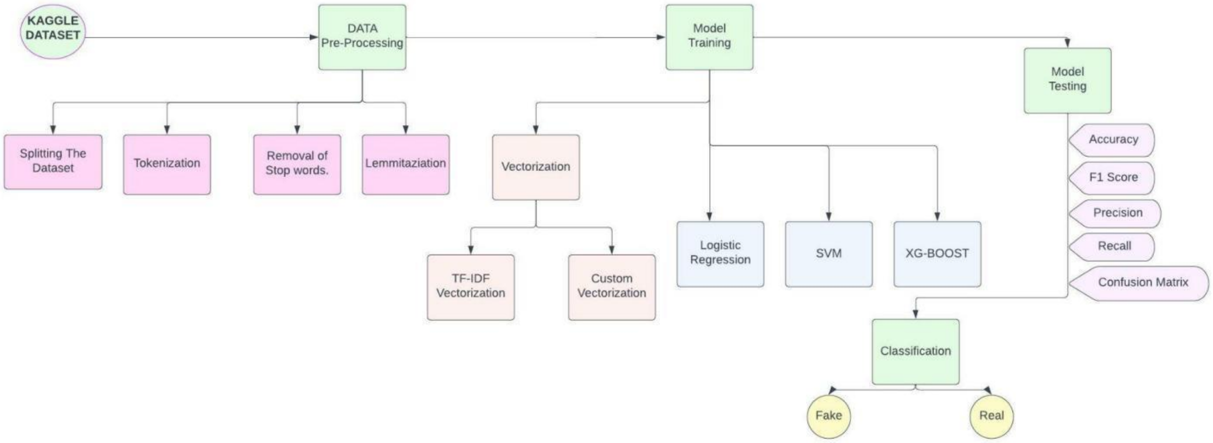
Abstract:

The issue of fake news spreading over social media and various media is a matter of serious concern in the sense that they have the ability to cause a lot of damage to the economy and leadership of the countries. The issue of fake news detection has been subjected to a lot of research. Social media and news outlets may publish fake news to increase readability among the citizens. This paper focuses on the analysis of various machine learning models and chooses the model which gives the best result. The primary aim is to classify given news as fake or true with the help of various tools like python, NLP, and scikit-learn. We have taken a Kaggle dataset for our analysis. The various models which are explored in this paper are Logistic Regression, Decision Tree classification, Gradient Boost classifier, Random Forest.

High-Level Design



Low-Level Design



Database:

The data consists of 20,800 files. It consists of 5 columns (‘id’,’Author’,’Title’,’Text’,’Label’) with the last column consisting of label values as 1 or 0 where 1 means False News and 0 means True News.
There are 10,387 true news and 10,413 false news in our database

Implementation of modules:

The technology used for building the code included python, sklearn and numpy.

1) ***Preprocessing:***
   We have to first check if there are any null values in the data. We fill or replace the null values using appropriate data using data Imputation. Now, we merge the data values of ‘title’, ‘Author’ and ‘Text’ into a single column. Now, we apply preprocessing methods such as removing symbols, tokenization, stop words removal and Lemmatization.
2) ***Models training:***
   After Preprocessing, we have to vectorize the input data. Here, we use count vectorizer. After this we split the data into X_train, Y_train,X_Test,Y_Test. Now we can train our machine with the help of various classification models such as Logistic Regression, SVM and XG Boost based on this input.

3) ***Models testing:***
   Finally we evaluated the three algorithms on testing data using metrics like accuracy score , precision and recall for each individual model using sklearn library in python. We also drew the confusion matrix for better analysis.

4) ***Classification:***
   The data is a binary classification data which detects if there is any Botnet attack happening due to any traffic inside the network. This system works as an IDS for network anomalies and can prevent the network architecture from being compromised.

Models Used:

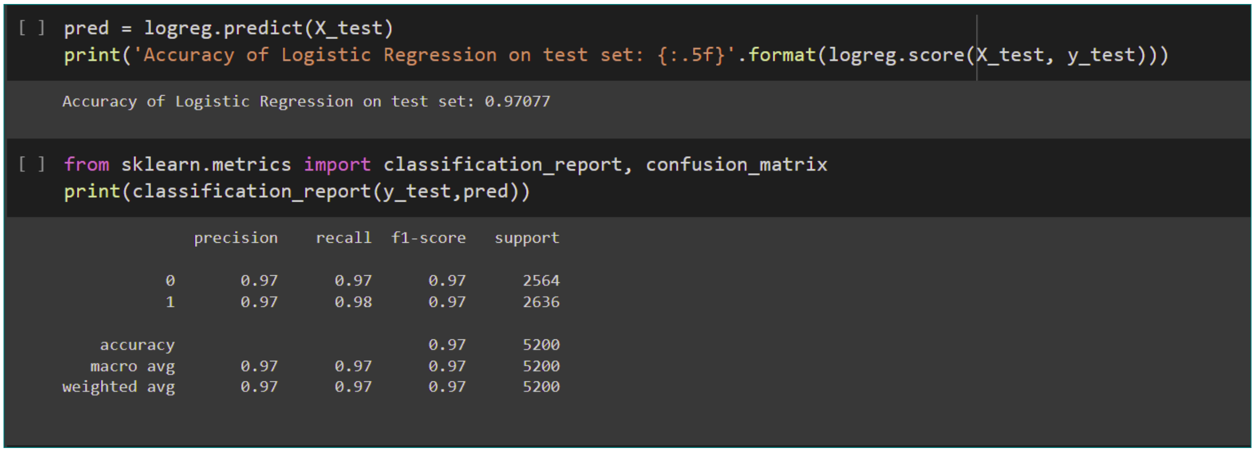
1)**Logistic Regression**: Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logic regression) is estimating the parameters of a logistic model a form of binary regression.

2)Linear-SVM Model: A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text. We are using Linear svm as it can scale better and gives more freedom with the loss function.

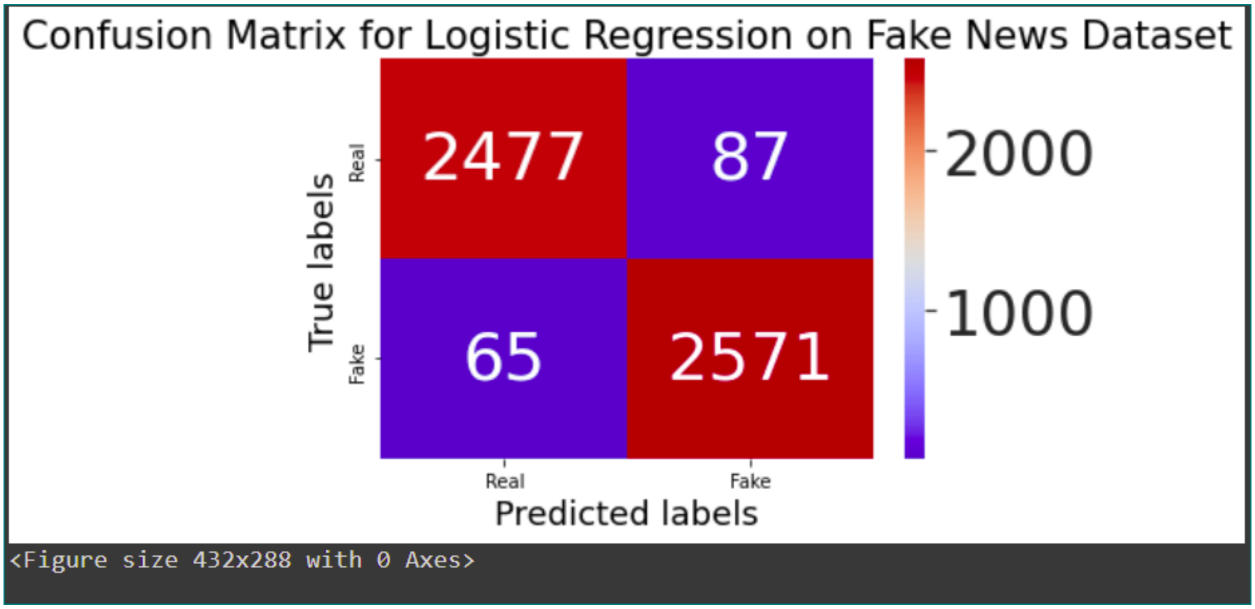
3) XG Boost : In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

Results :

LOGISTIC REGRESSION:

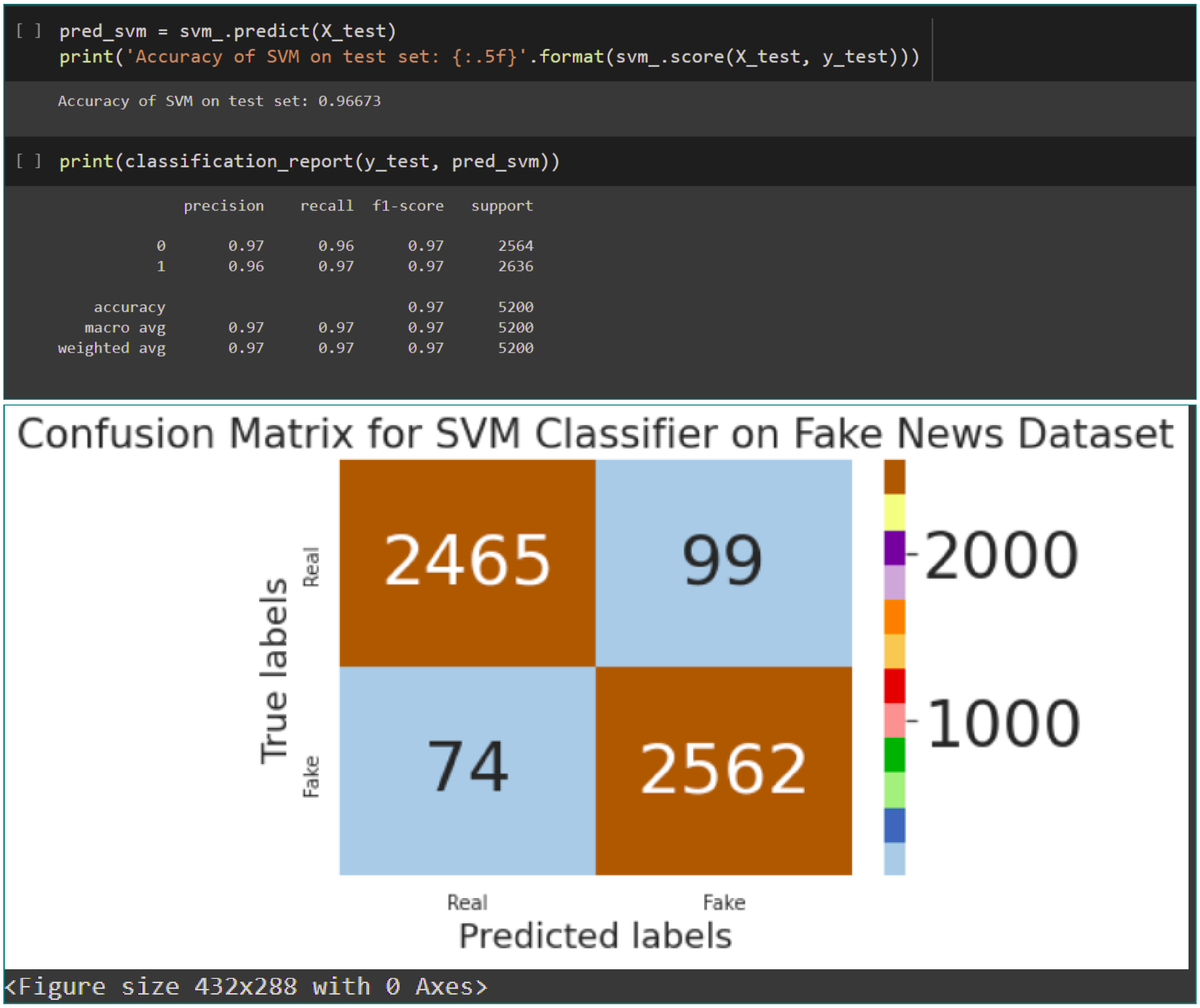


```
[ ] pred = logreg.predict(X_test)
    print('Accuracy of Logistic Regression on test set: {:.5f}'.format(logreg.score(X_test, y_test)))

    Accuracy of Logistic Regression on test set: 0.97077

[ ] from sklearn.metrics import classification_report, confusion_matrix
    print(classification_report(y_test,pred))

                  precision    recall  f1-score   support

               0       0.97      0.97      0.97      2564
               1       0.97      0.98      0.97      2636

        accuracy                           0.97      5200
       macro avg       0.97      0.97      0.97      5200
    weighted avg       0.97      0.97      0.97      5200
```



```
<Figure size 432x288 with 0 Axes>
```

SVM:



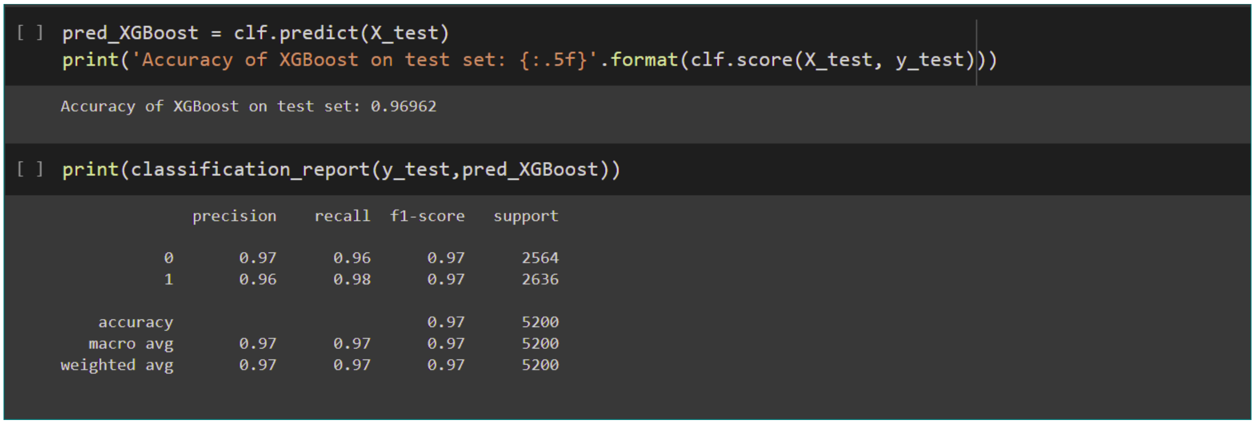
```
[ ] pred_svm = svm_.predict(X_test)
    print('Accuracy of SVM on test set: {:.5f}'.format(svm_.score(X_test, y_test)))

    Accuracy of SVM on test set: 0.96673
```

```
[ ] print(classification_report(y_test, pred_svm))

                  precision    recall  f1-score   support

               0       0.97      0.96      0.97      2564
               1       0.96      0.97      0.97      2636

        accuracy                           0.97      5200
       macro avg       0.97      0.97      0.97      5200
    weighted avg       0.97      0.97      0.97      5200
```

Confusion Matrix for SVM Classifier on Fake News Dataset

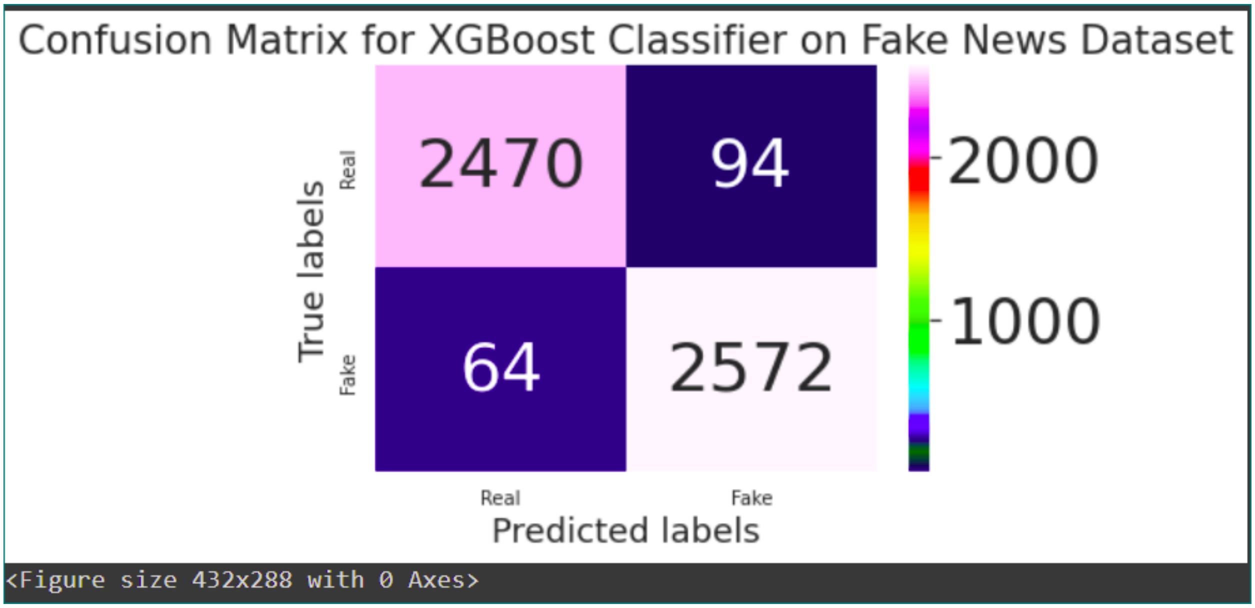
```
<Figure size 432x288 with 0 Axes>
```

XG-BOOST:



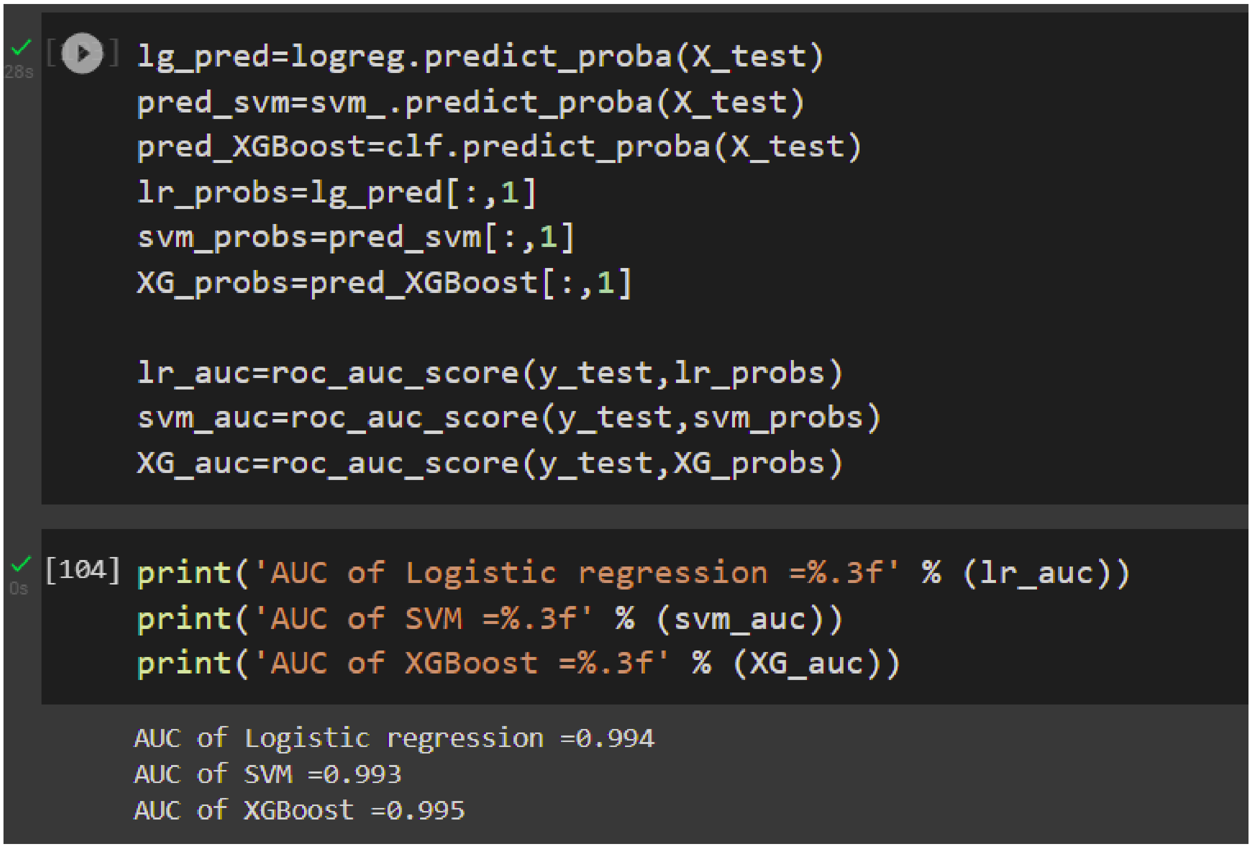
```
[ ] pred_XGBoost = clf.predict(X_test)
    print('Accuracy of XGBoost on test set: {:.5f}'.format(clf.score(X_test, y_test)))

    Accuracy of XGBoost on test set: 0.96962
```

```
[ ] print(classification_report(y_test,pred_XGBoost))

                  precision    recall  f1-score   support

               0       0.97      0.96      0.97      2564
               1       0.96      0.98      0.97      2636

        accuracy                           0.97      5200
       macro avg       0.97      0.97      0.97      5200
    weighted avg       0.97      0.97      0.97      5200
```



```
<Figure size 432x288 with 0 Axes>
```

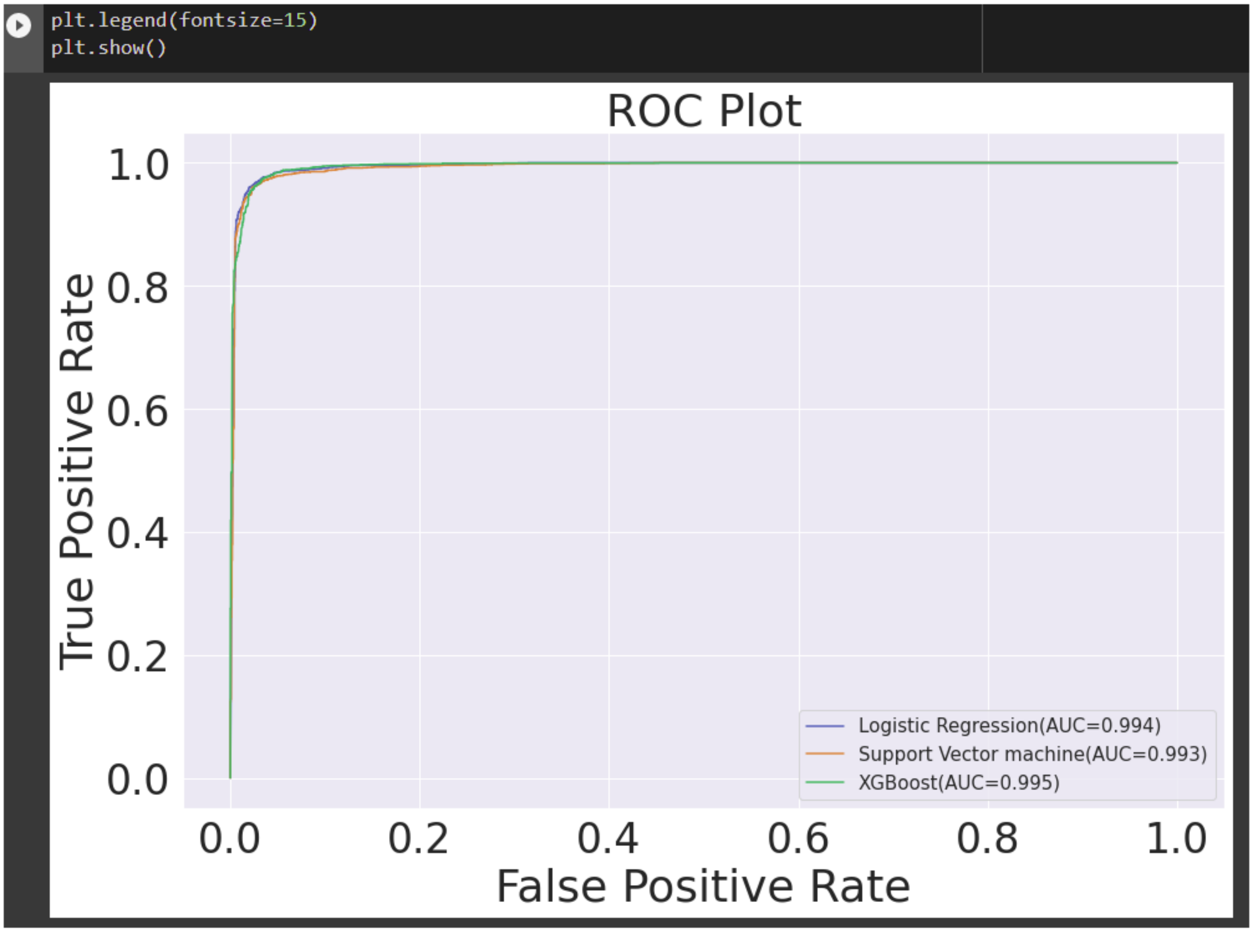
ROC curve and AUC score



```
lg_pred=logreg.predict_proba(X_test)
pred_svm=svm_.predict_proba(X_test)
pred_XGBoost=clf.predict_proba(X_test)
lr_probs=lg_pred[:,1]
svm_probs=pred_svm[:,1]
XG_probs=pred_XGBoost[:,1]

lr_auc=roc_auc_score(y_test,lr_probs)
svm_auc=roc_auc_score(y_test,svm_probs)
XG_auc=roc_auc_score(y_test,XG_probs)
```

```
print('AUC of Logistic regression =%.3f' % (lr_auc))
print('AUC of SVM =%.3f' % (svm_auc))
print('AUC of XGBoost =%.3f' % (XG_auc))
```

```
AUC of Logistic regression =0.994
AUC of SVM =0.993
AUC of XGBoost =0.995
```



```
plt.legend(fontsize=15)
plt.show()
```

Conclusion:

It is observed that Logistic regression gives the highest accuracy of 97.07% and SVM resulted in the lowest accuracy of 96.673%. The XG-Boost model resulted in an accuracy of 96.962%
We can improve this accuracy with the help of better pre-processing. It can further be increased by applying Data Augmentation to increase the size of Data.

Source Code Link:

https://colab.research.google.com/drive/18HYTxeM1em5Ed1g9hgo0l0NJ0WDuQtiJ?usp=sharing