**DATASET – 03**

**K-fold cross validation for all classifiers**

**Bag of words confusion matrix and F1 scores**

Total statements classified: 31403

Score: 0.9491494182792882

score length 5

Confusion matrix:

[[13921 1028]

[ 664 15790]]

Total statements classified: 31403

Score: 0.9609398497060028

score length 5

Confusion matrix:

[[14512 437]

[ 833 15621]]

Total statements classified: 31403

Score: 0.9582464568580642

score length 5

Confusion matrix:

[[14400 549]

[ 814 15640]]

Total statements classified: 31403

Score: 0.9502748918693611

score length 5

Confusion matrix:

[[14508 441]

[ 1160 15294]]

Total statements classified: 31403

Score: 0.9409495874264616

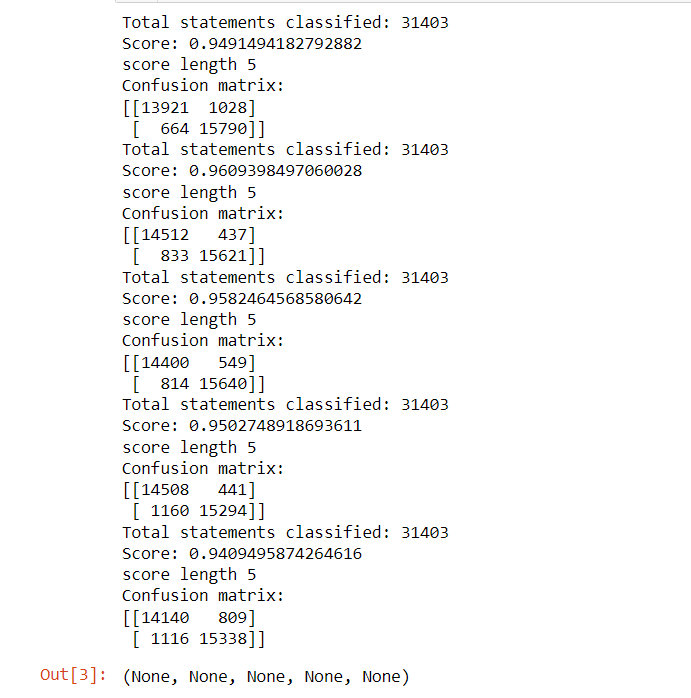
score length 5

Confusion matrix:

[[14140 809]

[ 1116 15338]]

Out[3]:(None, None, None, None, None)



So far we have used bag of words technique to extract the features and passed those featuers into classifiers. We have also seen the f1 scores of these classifiers. now lets enhance these features using term frequency weights with various n-grams.

**n-grams & tfidf confusion matrix and F1 scores**

Total statements classified: 31403

Score: 0.9426988269249872

score length 5

Confusion matrix:

[[13504 1445]

[ 495 15959]]

Total statements classified: 31403

Score: 0.9374283540182601

score length 5

Confusion matrix:

[[13871 1078]

[ 987 15467]]

Total statements classified: 31403

Score: 0.9517893572928662

score length 5

Confusion matrix:

[[14116 833]

[ 757 15697]]

Total statements classified: 31403

Score: 0.8346097309632091

score length 5

Confusion matrix:

[[ 8769 6180]

[ 244 16210]]

Total statements classified: 31403

Score: 0.907757925710716

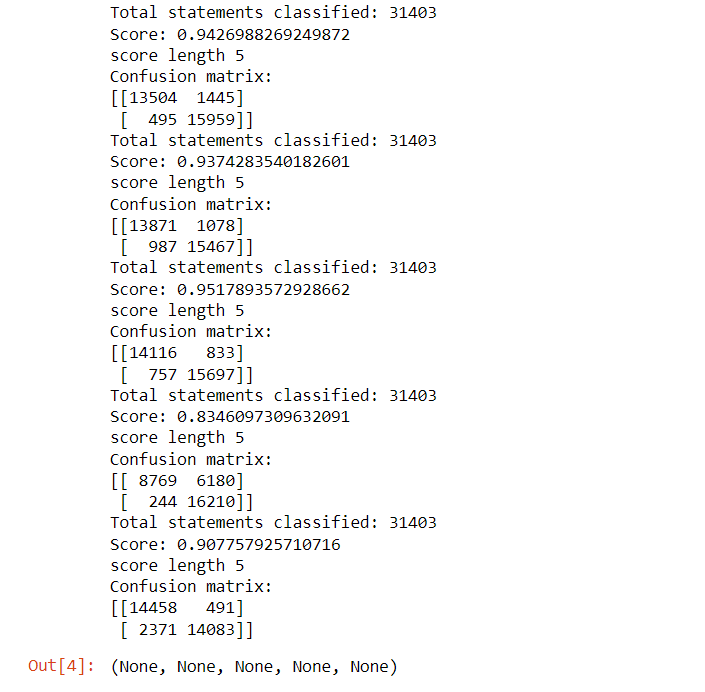
score length 5

Confusion matrix:

[[14458 491]

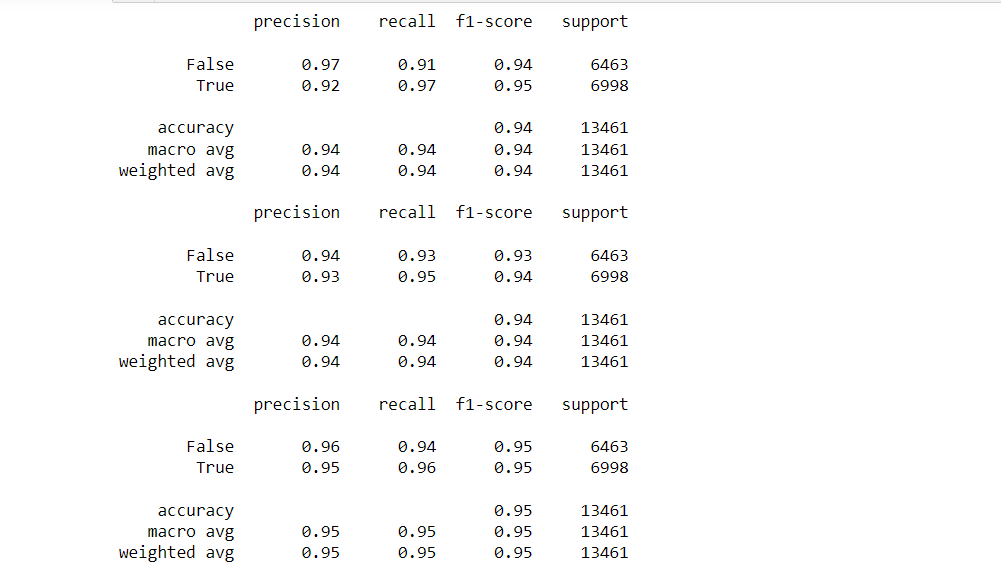
[ 2371 14083]]

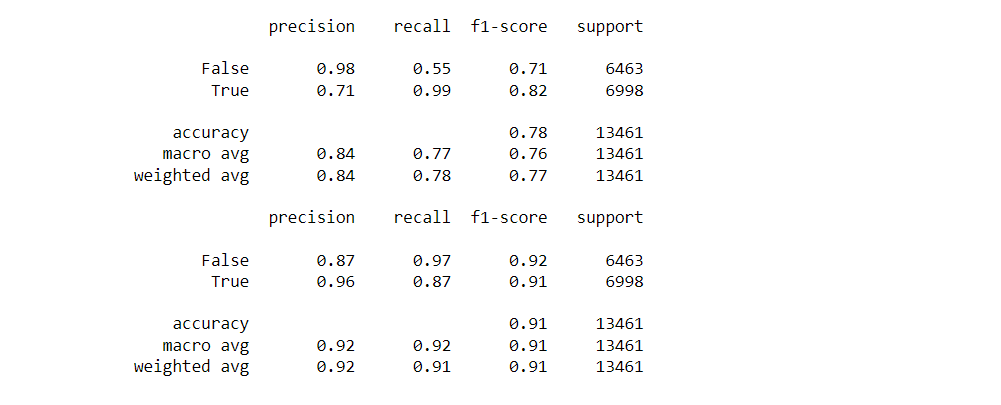
Out[4]:(None, None, None, None, None)



Out of all the models fitted, we would take 2 best performing model. we would call them as

candidate models from the confusion matrix.

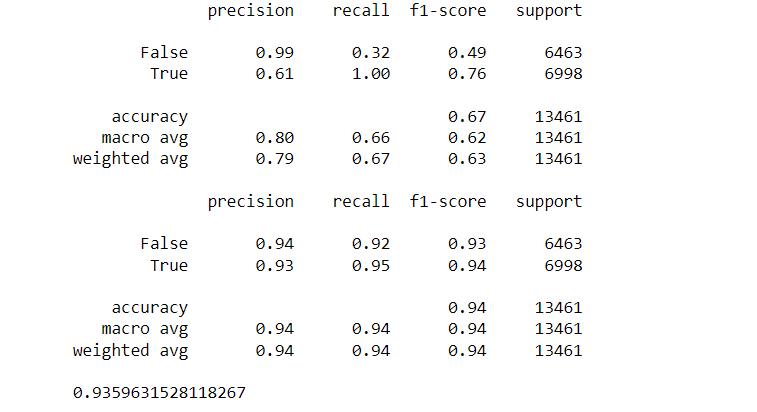




|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DATASET - 03 | Precision(True) | Precision(False) | Recall(True) | Recall(False) | F1-Score(True) | F1-Score(False) | Accuracy |
| Naïve -Bayes | 0.92 | 0.97 | 0.97 | 0.91 | 0.95 | 0.94 | 0.94 |
| Logistic -Regression | 0.93 | 0.94 | 0.95 | 0.93 | 0.94 | 0.93 | 0.94 |
| SVM | 0.95 | 0.96 | 0.96 | 0.94 | 0.95 | 0.95 | 0.95 |
| SGD | 0.71 | 0.98 | 0.99 | 0.55 | 0.82 | 0.71 | 0.78 |
| Random - Forest | 0.96 | 0.87 | 0.87 | 0.97 | 0.91 | 0.92 | 0.91 |

we can see that random forest and logistic regression are best performing in terms of precision and recall (take a look into false positive and true negative counts which appears to be low compared to rest of the models).

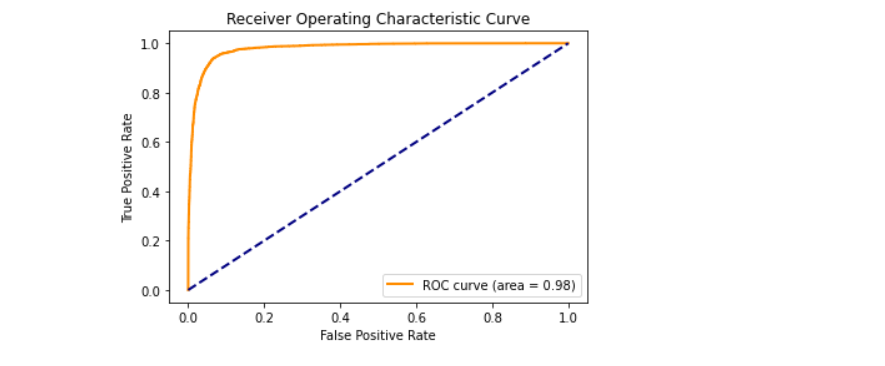
Running both random forest and logistic regression models again with best parameter found with GridSearch method.

****

by running both random forest and logistic regression with GridSearch's best parameter estimation, we found that for random forest model with n-gram has better accuracty than with the parameter estimated. The logistic regression model with best parameter has almost similar performance as n-gram model so logistic regression will be out choice of model for prediction.

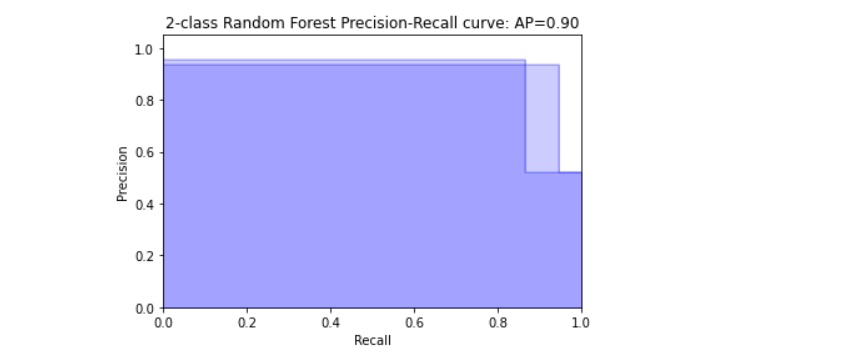
We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier I,e;Random Forest with an accuracy of 93.5%.

**The Result Of Random – Forest Values in the form of ROC – CURVE:**



We have also used Precision-Recall and learning curves to see how training and test set performs when we increase the amount of data in our classifiers.

**plotting Precision-Recall curve Of Random Forest.**



**saving best model to the disk i.e**; **” final\_model.sav”.**

**Plotting learing curves:**

**Plot the std deviation as a transparent range at each training set size .**

plt.fill\_between(train\_sizes, train\_scores\_mean - train\_scores\_std, train\_scores\_mean + train\_scores\_std, alpha=0.1, color="r")

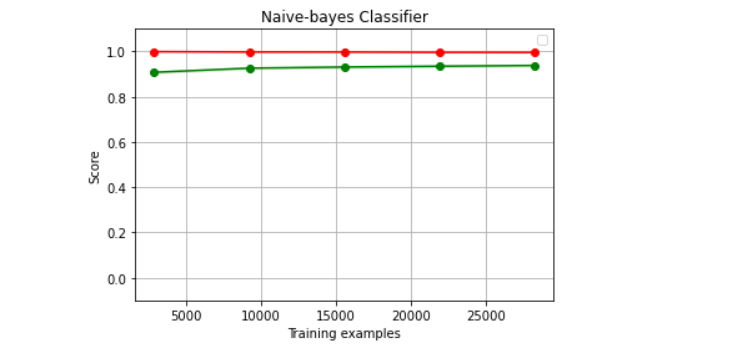
plt.fill\_between(train\_sizes, test\_scores\_mean - test\_scores\_std, test\_scores\_mean + test\_scores\_std, alpha=0.1, color="g")

**plot the average training and test score lines at each training set size**.

train\_sizes, train\_scores\_mean, 'o-', color="r", label="Training score"

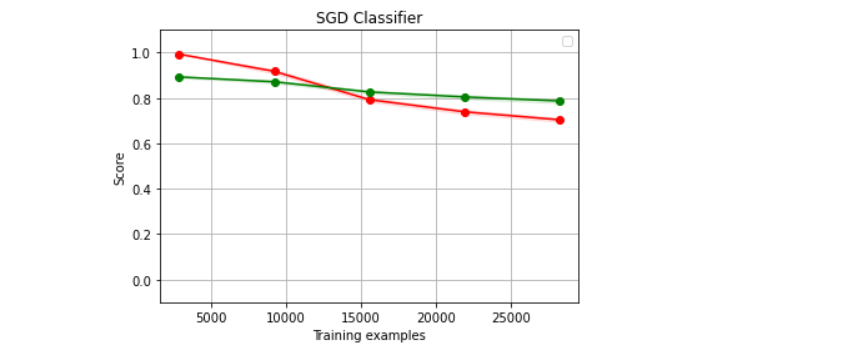
train\_sizes, test\_scores\_mean, 'o-', color="g", label="Cross-validation score”

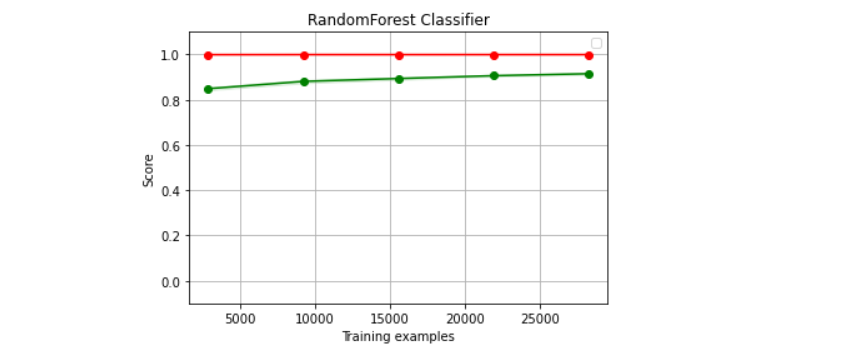
**Learing curves for each of the classifiers are as follows:**











We have used Naive-bayes, Logistic Regression, Linear SVM, Stochastic gradient descent and Random forest classifiers from sklearn. Each of the extracted features were used in all of the classifiers. Once fitting the model, we compared the f1 score and checked the confusion matrix. After fitting all the classifiers, 2 best performing models were selected as candidate models for fake news classification. We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier. Finally selected model was used for fake news detection with the probability of truth.