**DATASET – 02**

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

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

Total statements classified: 17588

Score: 0.8869827243159225

score length 5

Confusion matrix:

[[8816 518]

[ 792 7462]]

Total statements classified: 17588

Score: 0.8767371549020586

score length 5

Confusion matrix:

[[8554 780]

[ 484 7770]]

Total statements classified: 17588

Score: 0.8708397485952719

score length 5

Confusion matrix:

[[8559 775]

[ 569 7685]]

Total statements classified: 17588

Score: 0.8715358932113016

score length 5

Confusion matrix:

[[8473 861]

[ 466 7788]]

Total statements classified: 17588

Score: 0.8495729022821523

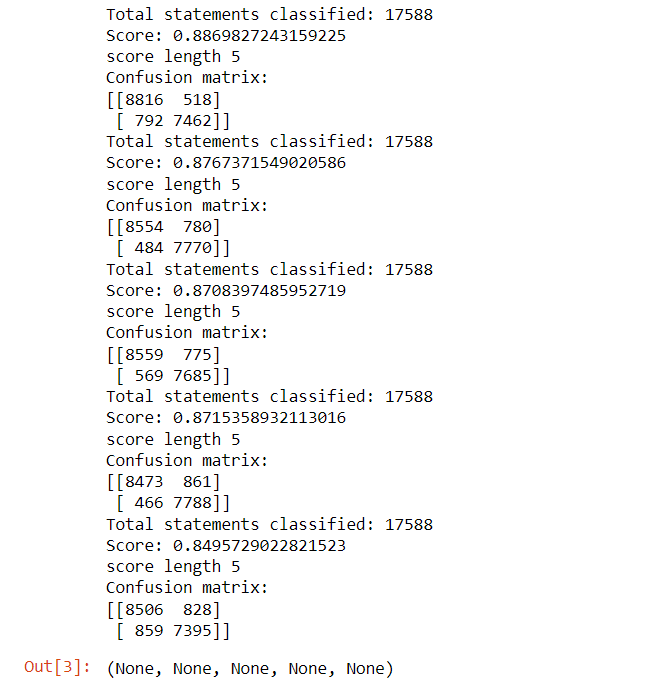
score length 5

Confusion matrix:

[[8506 828]

[ 859 7395]]

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: 17588

Score: 0.8091754450909786

score length 5

Confusion matrix:

[[8657 677]

[1990 6264]]

Total statements classified: 17588

Score: 0.814859346524069

score length 5

Confusion matrix:

[[8374 960]

[1331 6923]]

Total statements classified: 17588

Score: 0.8544229264341437

score length 5

Confusion matrix:

[[8547 787]

[ 897 7357]]

Total statements classified: 17588

Score: 0.5334620800191266

score length 5

Confusion matrix:

[[6934 2400]

[4454 3800]]

Total statements classified: 17588

Score: 0.7736037838519427

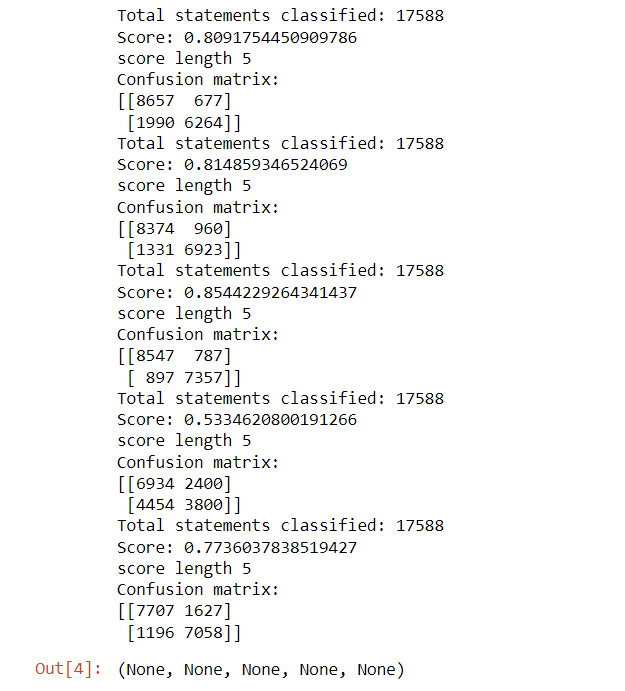
score length 5

Confusion matrix:

[[7707 1627]

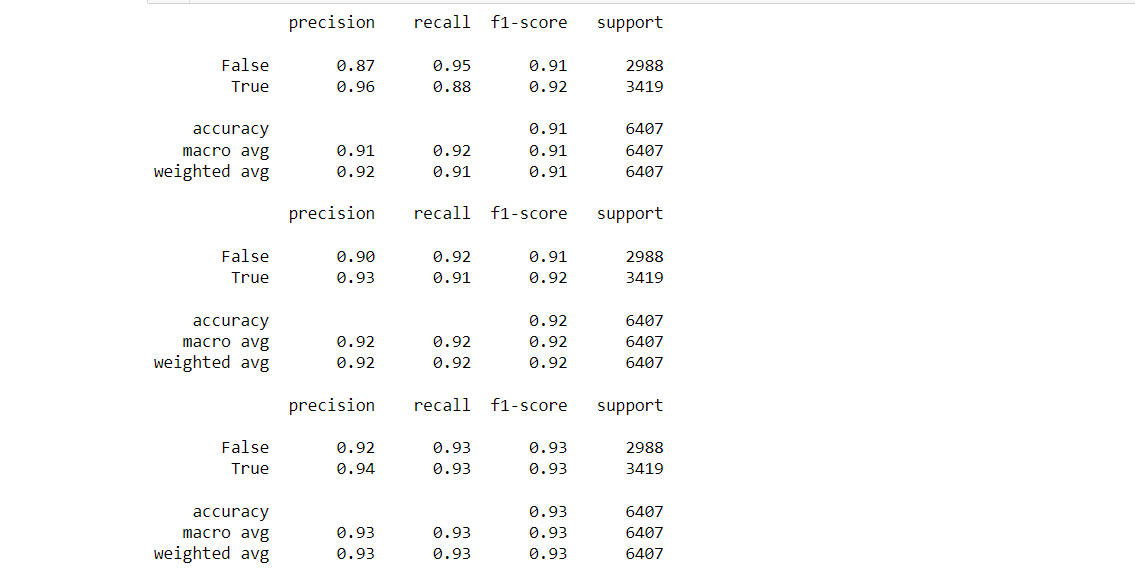
[1196 7058]]

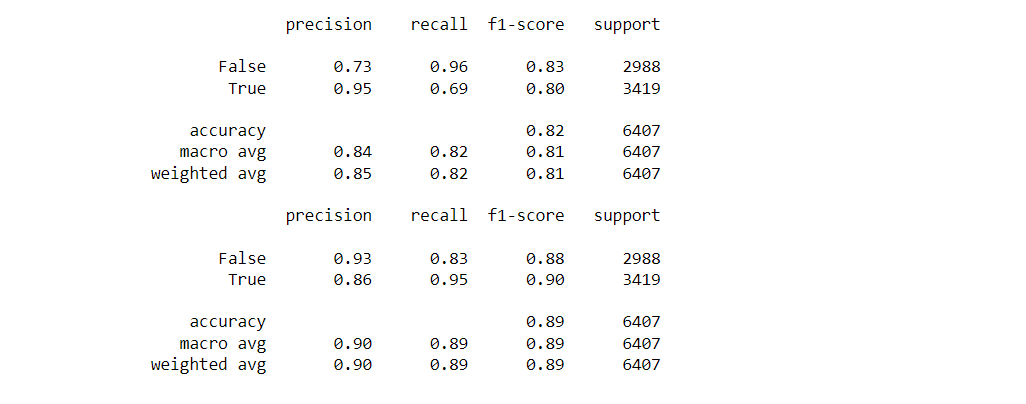
Out[4]:(None, None, None, None, None)



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

candidate models from the confusion matrix.

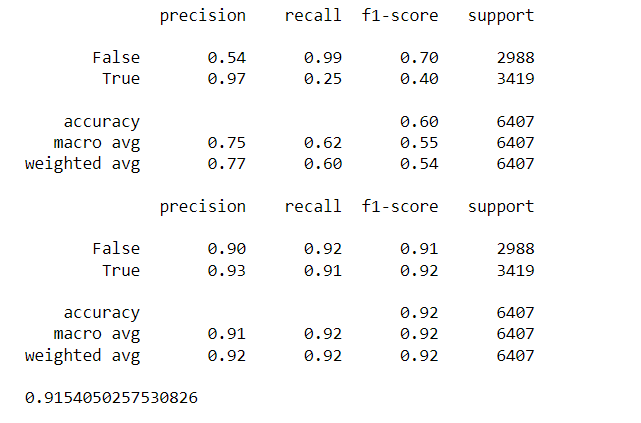




|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DATASET - 01 | Precision(True) | Precision(False) | Recall(True) | Recall(False) | F1-Score(True) | F1-Score(False) | Accuracy |
| Naïve -Bayes | 0.96 | 0.87 | 0.88 | 0.95 | 0.92 | 0.91 | 0.91 |
| Logistic -Regression | 0.93 | 0.90 | 0.91 | 0.92 | 0.92 | 0.91 | 0.92 |
| SVM | 0.94 | 0.92 | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 |
| SGD | 0.95 | 0.73 | 0.69 | 0.96 | 0.80 | 0.83 | 0.82 |
| Random - Forest | 0.86 | 0.93 | 0.95 | 0.83 | 0.90 | 0.88 | 0.89 |

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).

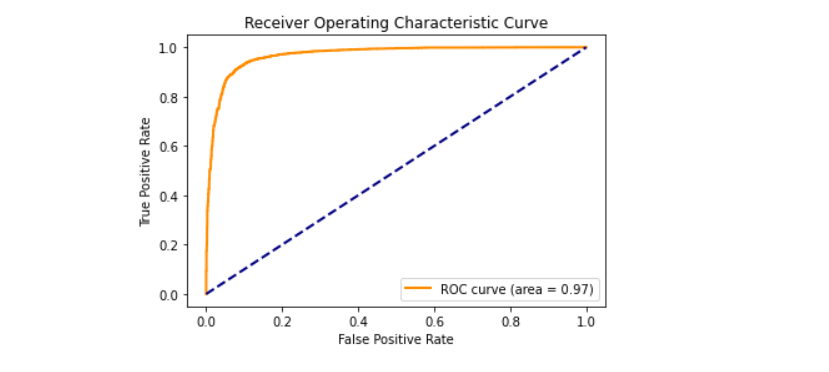
Running both random forest and logistic regression models again with best parameter found with GridSearch method.



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.

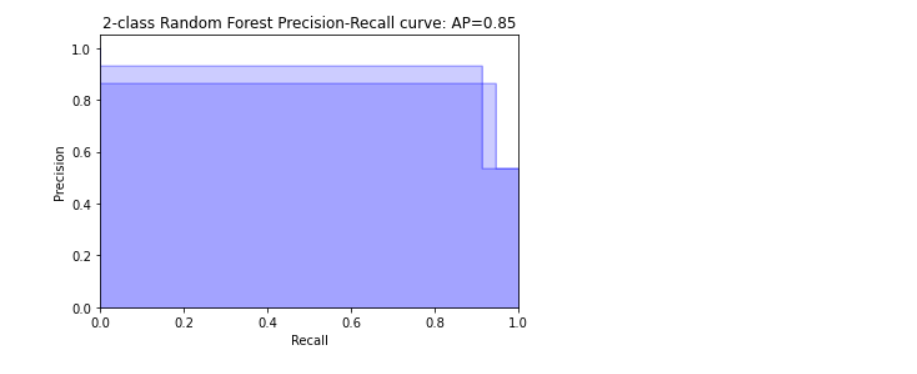
We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier I,e;Random Forest with an accuracy of 91.5%.

**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")

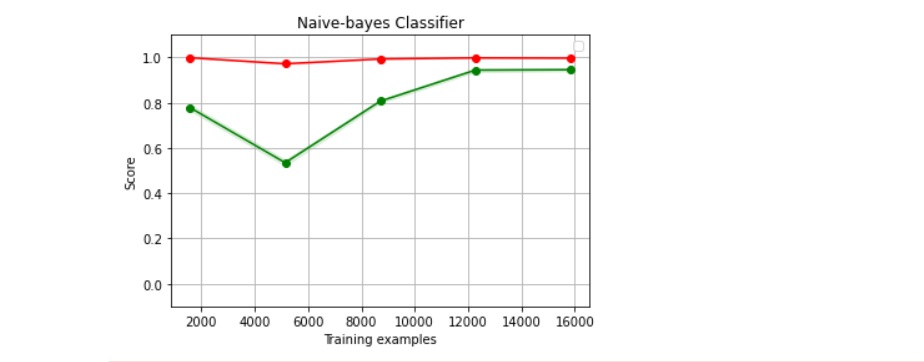
**plot the average training and test score lines at each training set size**.

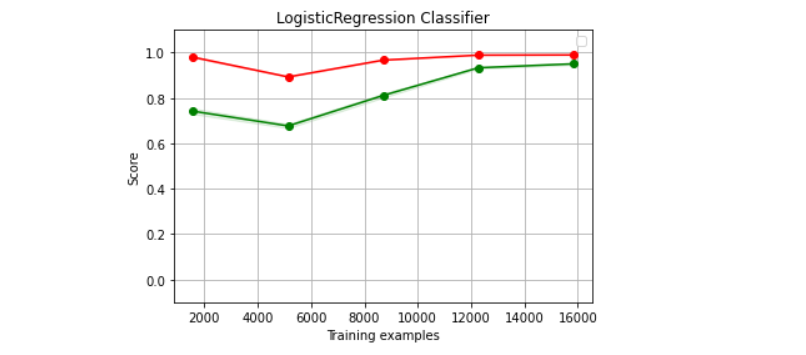
train\_sizes, train\_scores\_mean, 'o-', color="r", label="Training score"

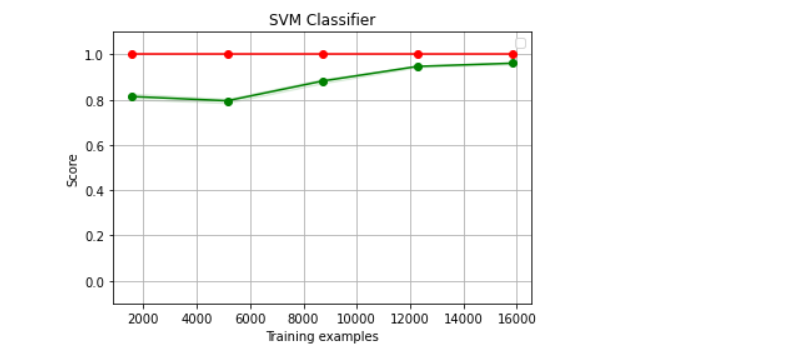
train\_sizes, test\_scores\_mean, 'o-', color="g", label="Cross-validation score”

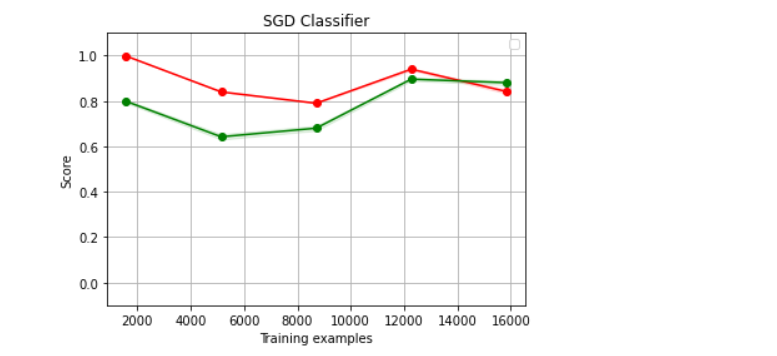
Sizes the window for readability and displays the plot and it shows error from 0 to 1.1.

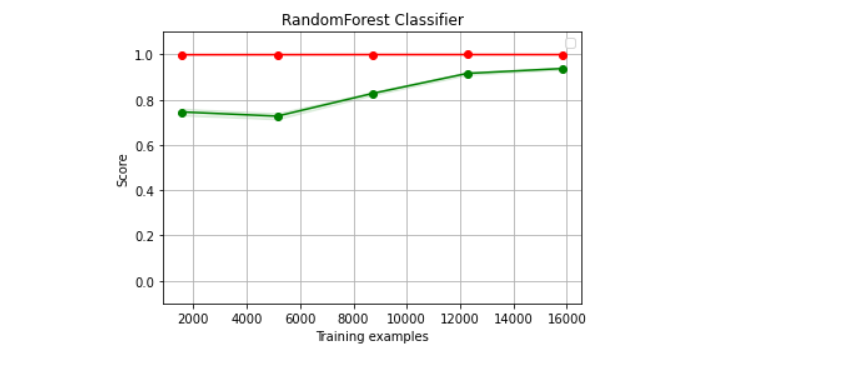
**Learing curves for each of the classifiers are as follows**:











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.