**DATASET – 01**

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

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

Total statements classified: 10240

Score: 0.66961153965076

score length 5

Confusion matrix:

[[2118 2370]

[1664 4088]]

Total statements classified: 10240

Score: 0.6466692934443682

score length 5

Confusion matrix:

[[2254 2234]

[1936 3816]]

Total statements classified: 10240

Score: 0.6104687487924283

score length 5

Confusion matrix:

[[2260 2228]

[2246 3506]]

Total statements classified: 10240

Score: 0.6625556074172465

score length 5

Confusion matrix:

[[2174 2314]

[1756 3996]]

Total statements classified: 10240

Score: 0.6703371565005776

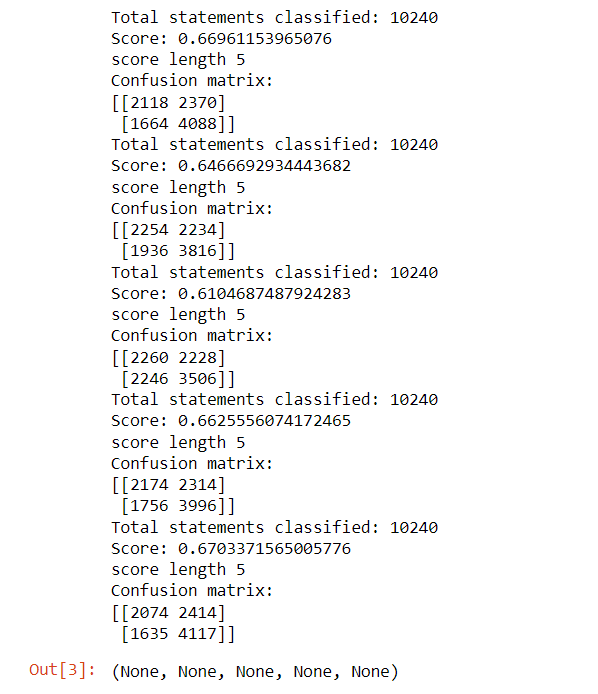
score length 5

Confusion matrix:

[[2074 2414]

[1635 4117]]

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: 10240

Score: 0.7224053159841455

score length 5

Confusion matrix:

[[ 758 3730]

[ 390 5362]]

Total statements classified: 10240

Score: 0.7044355553757985

score length 5

Confusion matrix:

[[1580 2908]

[1043 4709]]

Total statements classified: 10240

Score: 0.6790920142902143

score length 5

Confusion matrix:

[[2016 2472]

[1524 4228]]

Total statements classified: 10240

Score: 0.7190643331130575

score length 5

Confusion matrix:

[[ 5 4483]

[ 6 5746]]

Total statements classified: 10240

Score: 0.6330062374295281

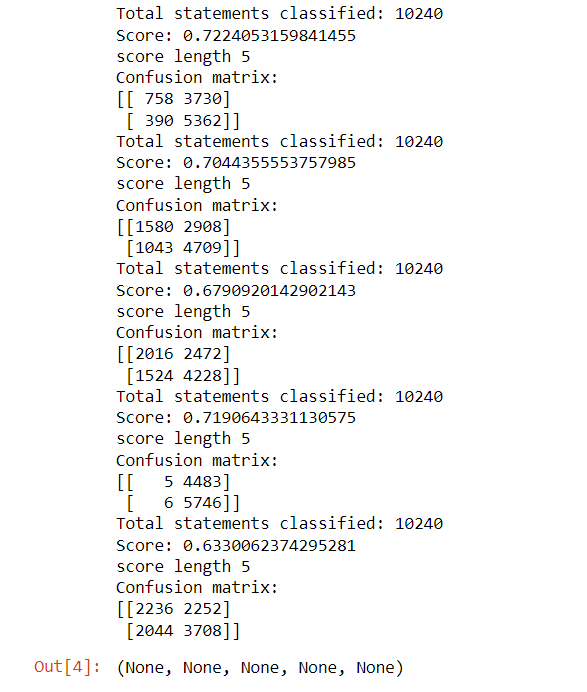
score length 5

Confusion matrix:

[[2236 2252]

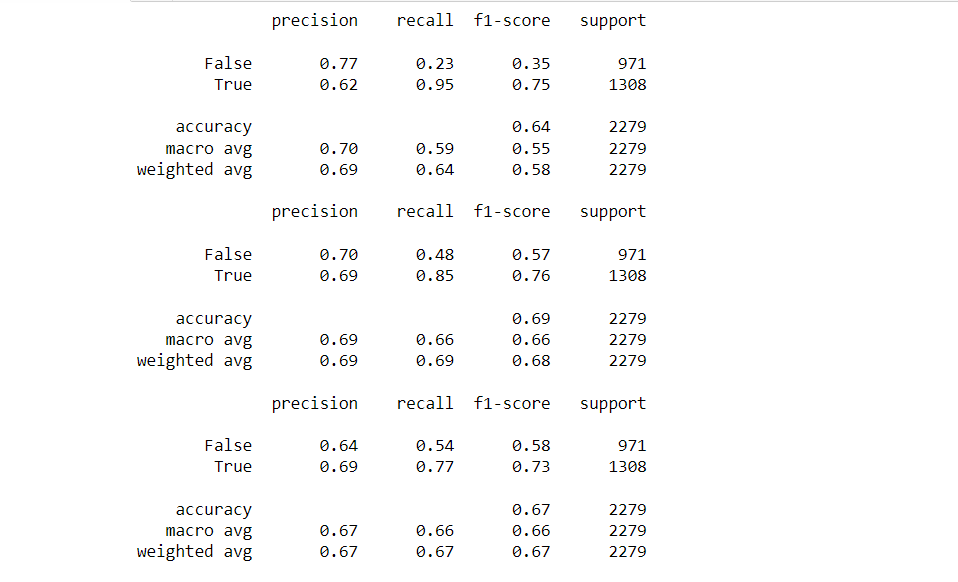
[2044 3708]]

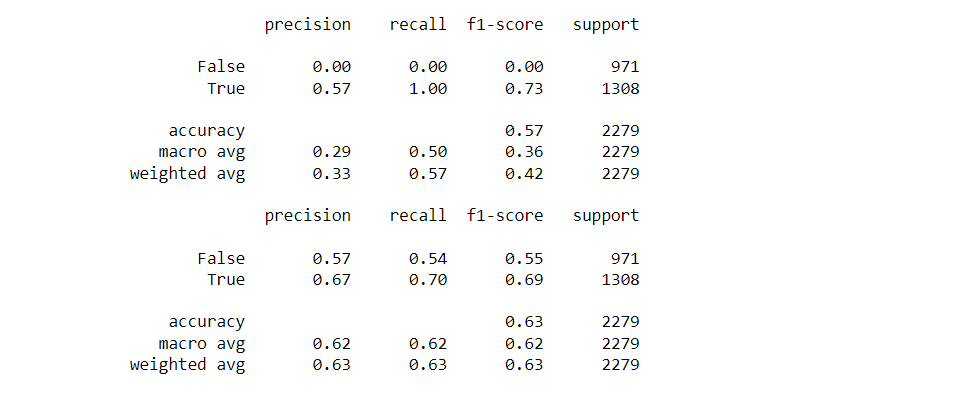
Out[4]: (None, None, None, None, None)



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

candidate models from the confusion matrix.

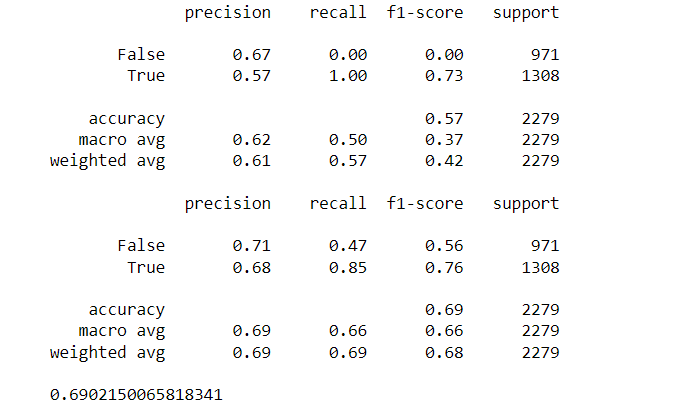




|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DATASET - 01 | Precision(True) | Precision(False) | Recall(True) | Recall(False) | F1-Score(True) | F1-Score(False) | Accuracy |
| Naïve -Bayes | 0.62 | 0.77 | 0.95 | 0.23 | 0.75 | 0.35 | 0.64 |
| Logistic -Regression | 0.69 | 0.70 | 0.85 | 0.48 | 0.76 | 0.57 | 0.69 |
| SVM | 0.69 | 0.64 | 0.77 | 0.54 | 0.73 | 0.58 | 0.67 |
| SGD | 0.57 | 0.00 | 1.00 | 0.00 | 0.73 | 0.00 | 0.57 |
| Random - Forest | 0.67 | 0.57 | 0.70 | 0.54 | 0.69 | 0.55 | 0.63 |

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).

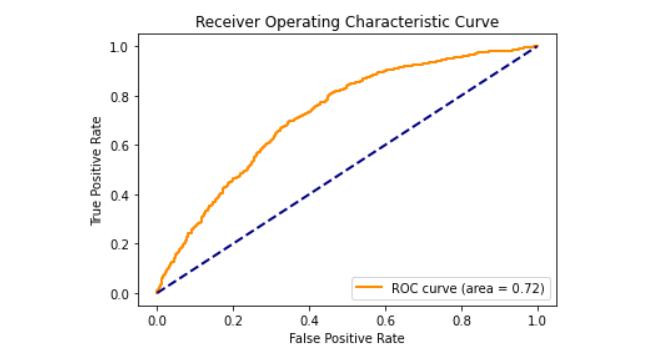
Running both random forest and logistic regression models again with best parameter found with GridSearch method.



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.

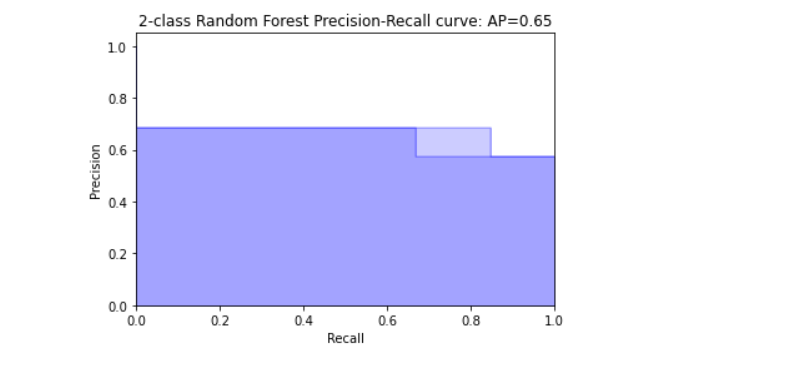
We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier I,e;Random Forest with an accuracy of 69%.

**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")

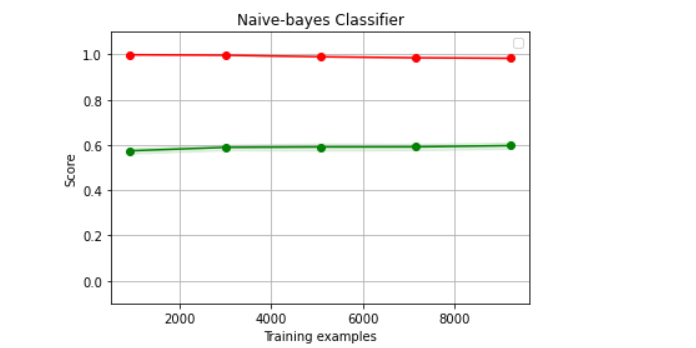
plt.fill\_between(train\_sizes, test\_scores\_mean - test\_scores\_std, test\_scores\_mean + test\_scores\_std, alpha=0.1, color="g")

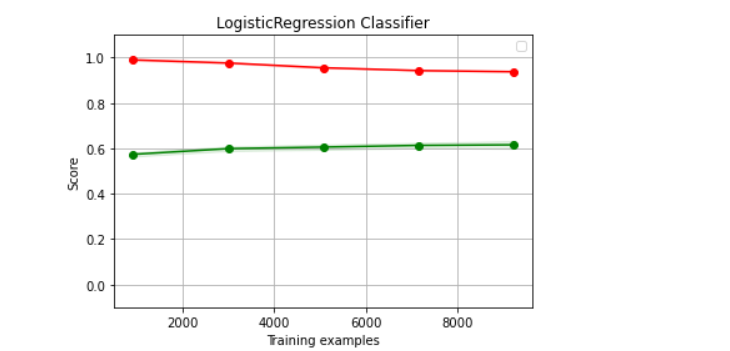
**plot the average training and test score lines at each training set size**.

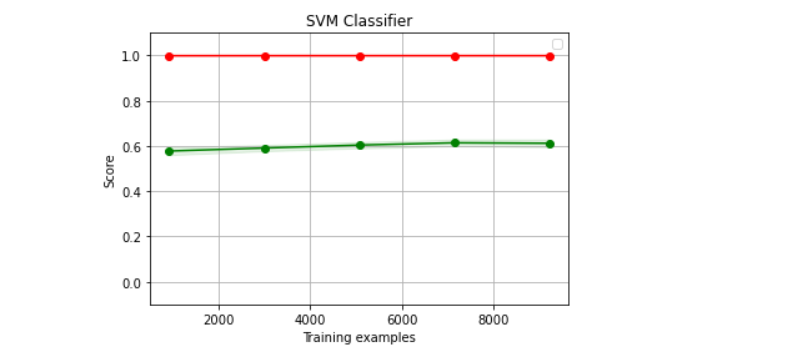
train\_sizes, train\_scores\_mean, 'o-', color="r", label="Training score"

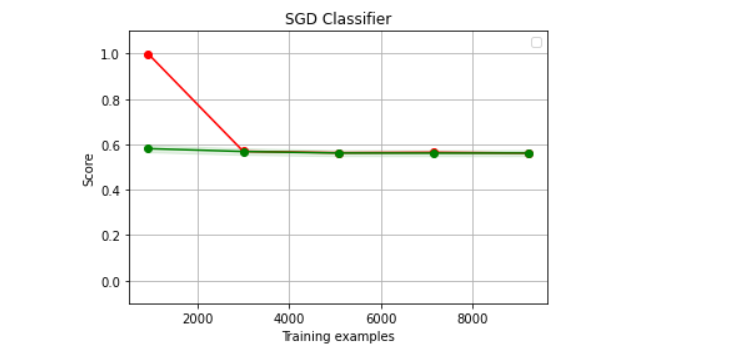
train\_sizes, test\_scores\_mean, 'o-', color="g", label="Cross-validation score”

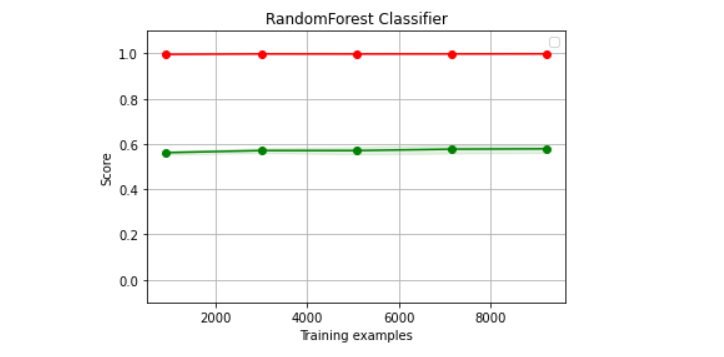
Sizes the window for readability and displays the plot and it shows error from 0 to 1.1.

**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.