**DATASET – 04**

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

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

Total statements classified: 29959

Score: 0.885875639458488

score length 5

Confusion matrix:

[[12676 1880]

[ 1661 13742]]

Total statements classified: 29959

Score: 0.9013070861345026

score length 5

Confusion matrix:

[[13081 1475]

[ 1557 13846]]

Total statements classified: 29959

Score: 0.885133233961563

score length 5

Confusion matrix:

[[12823 1733]

[ 1798 13605]]

Total statements classified: 29959

Score: 0.8951112557837888

score length 5

Confusion matrix:

[[13099 1457]

[ 1744 13659]]

Total statements classified: 29959

Score: 0.8861102967999143

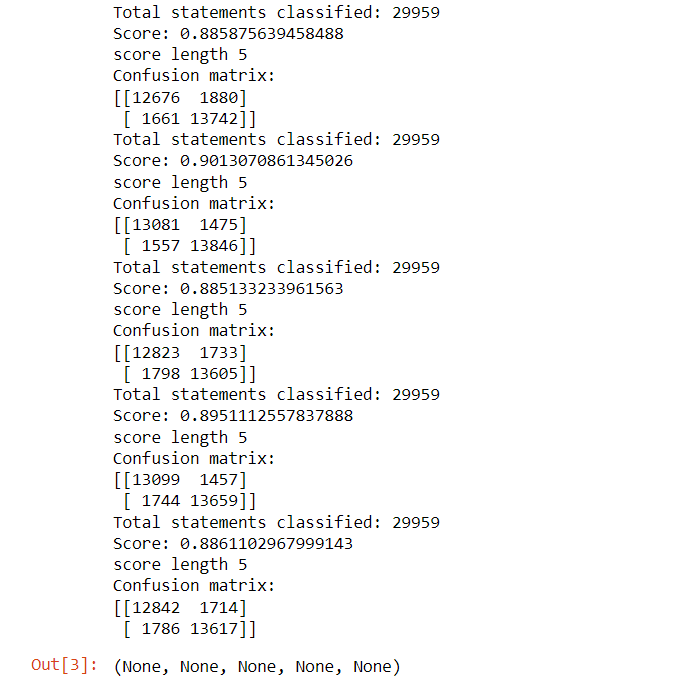
score length 5

Confusion matrix:

[[12842 1714]

[ 1786 13617]]

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: 29959

Score: 0.8910708903883693

score length 5

Confusion matrix:

[[12363 2193]

[ 1264 14139]]

Total statements classified: 29959

Score: 0.8903507056396729

score length 5

Confusion matrix:

[[12643 1913]

[ 1509 13894]]

Total statements classified: 29959

Score: 0.9028449550813213

score length 5

Confusion matrix:

[[13058 1498]

[ 1495 13908]]

Total statements classified: 29959

Score: 0.7569761123118762

score length 5

Confusion matrix:

[[ 4899 9657]

[ 141 15262]]

Total statements classified: 29959

Score: 0.8559358867328873

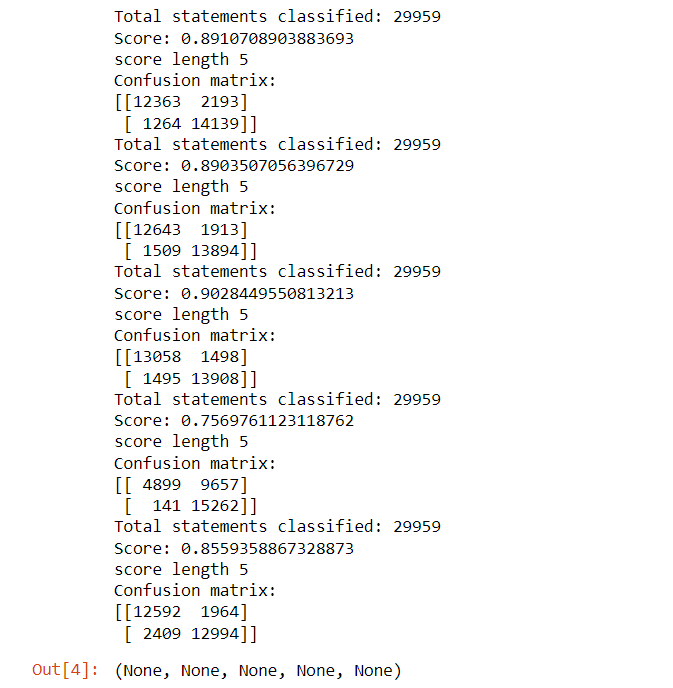
score length 5

Confusion matrix:

[[12592 1964]

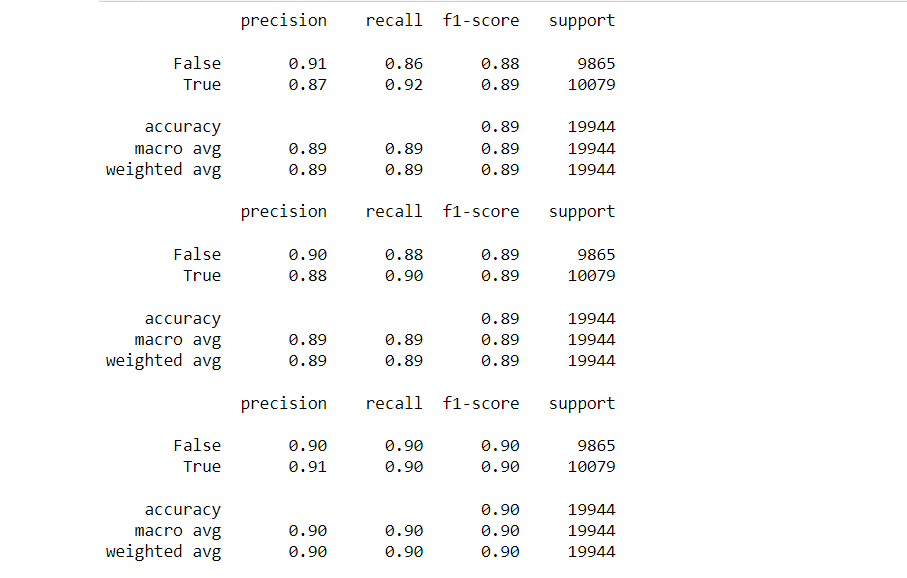
[ 2409 12994]]

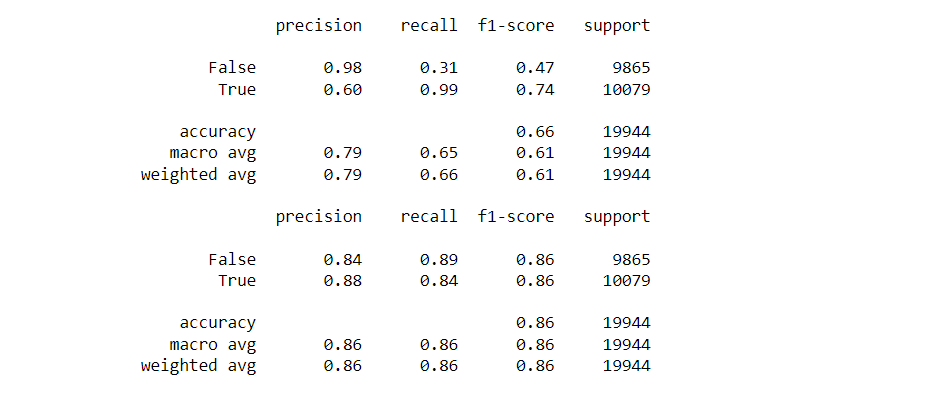
Out[4]:(None, None, None, None, None)



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

candidate models from the confusion matrix.

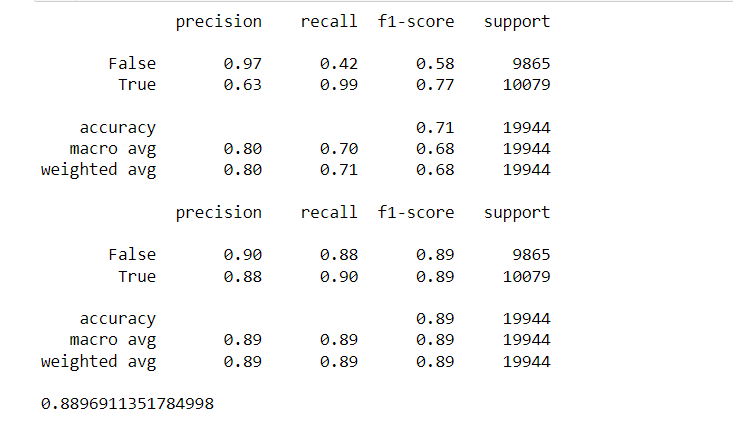




|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DATASET - 04 | Precision(True) | Precision(False) | Recall(True) | Recall(False) | F1-Score(True) | F1-Score(False) | Accuracy |
| Naïve -Bayes | 0.87 | 0.91 | 0.92 | 0.86 | 0.89 | 0.88 | 0.89 |
| Logistic -Regression | 0.88 | 0.90 | 0.90 | 0.88 | 0.89 | 0.89 | 0.89 |
| SVM | 0.91 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| SGD | 0.60 | 0.98 | 0.99 | 0.31 | 0.74 | 0.47 | 0.66 |
| Random - Forest | 0.88 | 0.84 | 0.84 | 0.89 | 0.86 | 0.86 | 0.86 |

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).

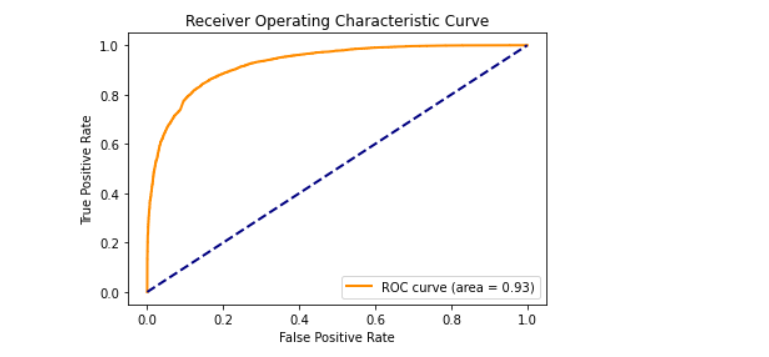
Running both random forest and logistic regression models again with best parameter found with GridSearch method.



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.

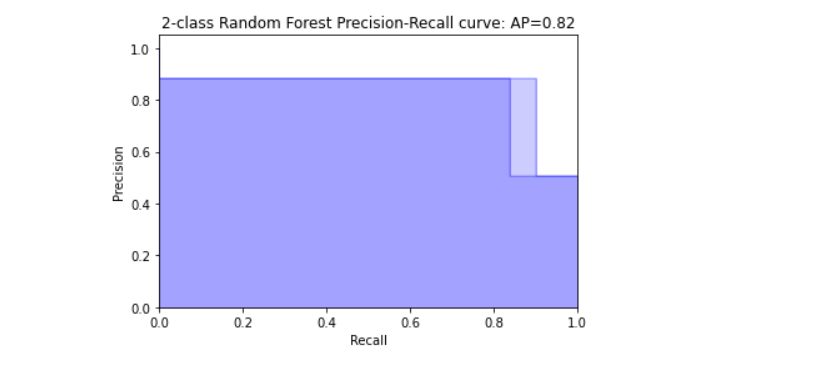
We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifier I,e;Random Forest with an accuracy of 88.9%.

**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.**



**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")

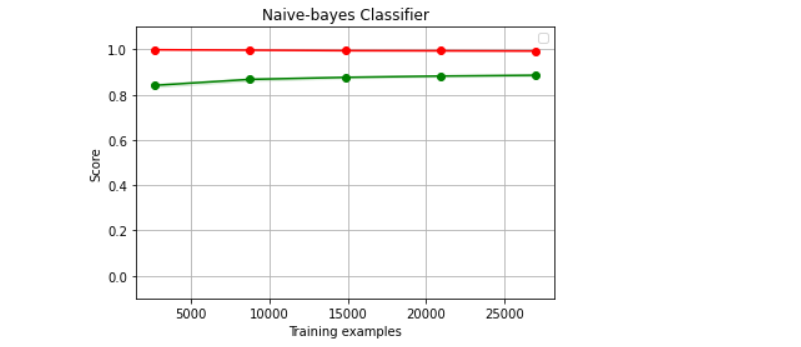
**plot the average training and test score lines at each training set size**.

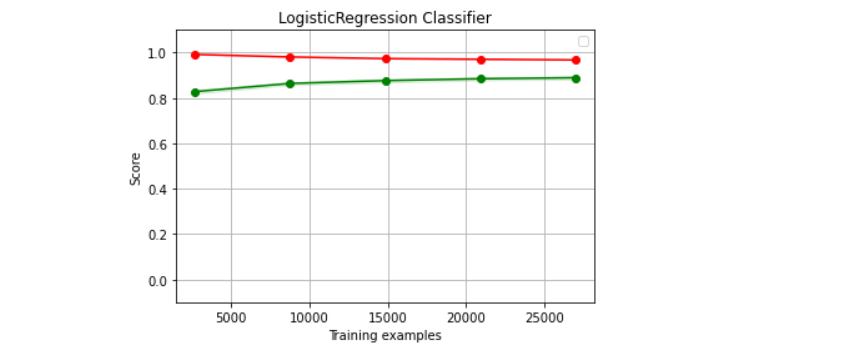
train\_sizes, train\_scores\_mean, 'o-', color="r", label="Training score"

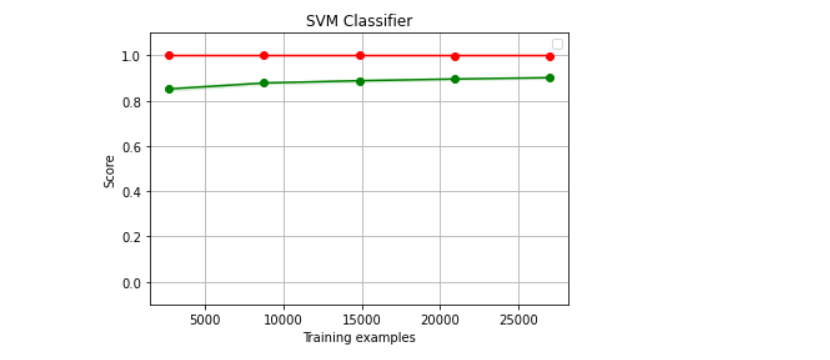
train\_sizes, test\_scores\_mean, 'o-', color="g", label="Cross-validation score”

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

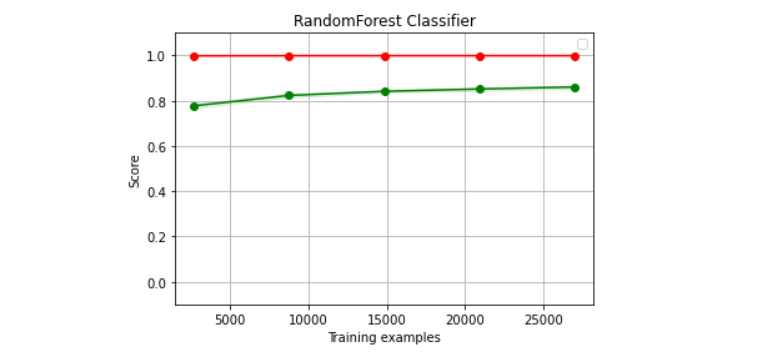
**Learing curves for each of the classifiers are as follows:**











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.