University of Illinois at Chicago

IDS 566 Advanced Text Analytics

Final project report

Document classification using Machine Learning on 20 Newsgroups

Submitted by:

Akanxa Jain (671123766)

Anupam Sinha (675450883)

Jeyenthi Raman(665390415)

Kush Varma(672468804)

Shalini Singh (663877768)

I. ABSTRACT

The 20 newsgroups dataset is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering. Each record in the dataset is actually a text file (a series of words). The data is trained over different classifiers such as Support Vector Machines, Naïve Bayes Model, Random Forests, Stochastic Gradient Descent and Logistic Regression. We have used Python Jupyter notebook for code execution and packages from libraries like sklearn, nltk, re, langid etc to test the model parameters and accuracy measurements. By tuning the parameters and relevant initialization, we aim to test the outcomes to find out the best model with maximum accuracy.

II. INTRODUCTION

- In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation.
- The data input we have here is in the form of text and we'll train this data over various classification models
- Machine learning model accuracy is the measurement used to determine which model is best at identifying relationships and patterns between variables in a dataset based on the input, or training, data. The better a model can generalize to 'unseen' data, the better predictions and insights it produces that deliver more business value

- The various steps used by us in modelling the text data are:
  1. ***Data Exploration and Cleaning***
     1.1 Variable Identification
     1.2 Outlier Treatment
     1.3 Missing Value Treatment
  2. **Data Transformation / Feature Extraction**
  3. ***Models Implemented***
     3.1 Naïve Bayes Model
     3.2 Decision Trees
     3.3 Random Forest
     3.4 Logistic Regression
     3.5 Stochastic Gradient Descent
     3.6 Support Vector Machines
  4. ***Training and testing results of the models***
  5. ***Comparative analysis of accuracy to find the best model***

III. DATA EXPLORATION AND CLEANING

- *Variable Identification*: The data has 1707 columns and target variable has 20 categories.
- *Outlier treatment*: The outliers have been neglected by initializing the min_df = 0.01 and max_df = 0.85

- *Removal of Non Ascii characters*: We defined a function strip_non_ascii that returns the string without non-ASCII characters
- *Corpus cleaning:* We removed unwanted hyperlinks, punctuation, numeric words, words less than 2 characters, @mentions by defining the remove_features function and converting the data to lowercase
- *Lemmatization:* Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. We defined a function 'lemmatize' and applied that to our train and test data to perform data cleaning

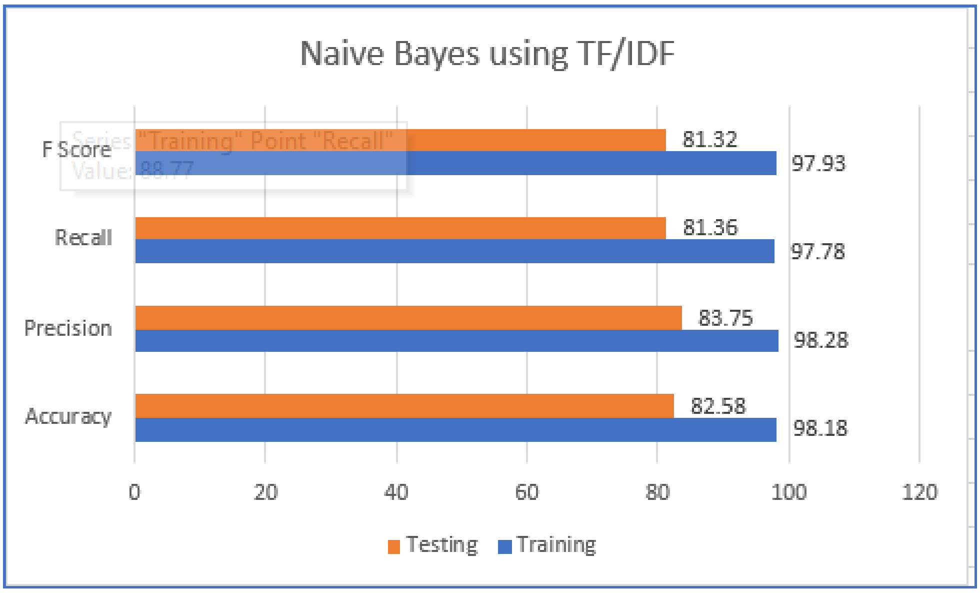
IV. FEATURE EXTRACTION/ DATA TRANSFORMATION

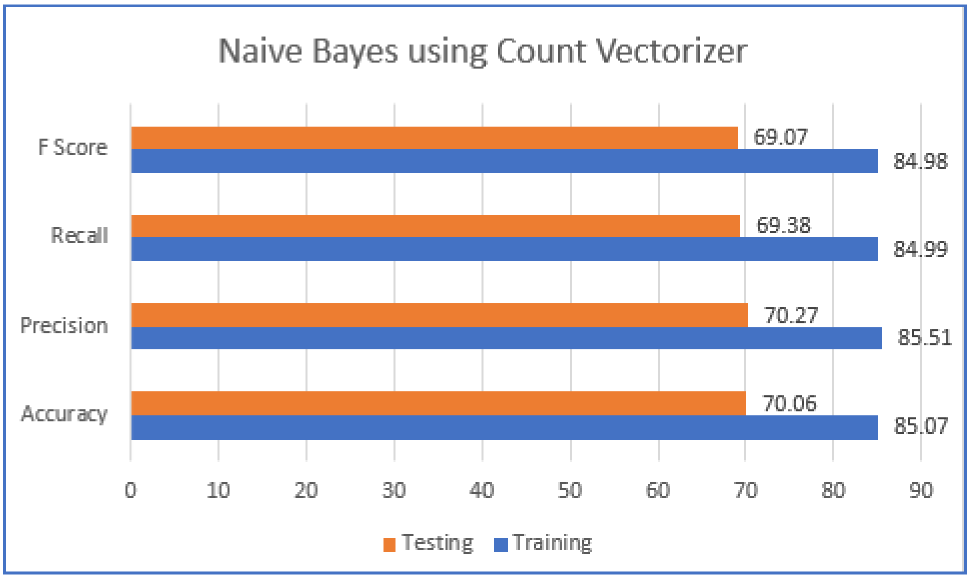
- Feature engineering is the science (and art) of extracting more information from existing data. In data modelling, transformation refers to the replacement of a variable by a function.
- We first created a TF IDF vectorizer that ignores all stop words of the English dictionary and removed the outliers as well. We then fit this to our training and test data.
- Using count vectorizer, we created the document term matrix for training and test data.
- By performing the above tasks, we have numerically encoded the text corpus, where each of the column is token from the text for each of the documents. In case of tf-idf, the values are normalized for the features with respect to the number of the documents containing a term. While count vectorizer simply contains count of all the tokens appearing in the documents.

V. DATA MODELLING

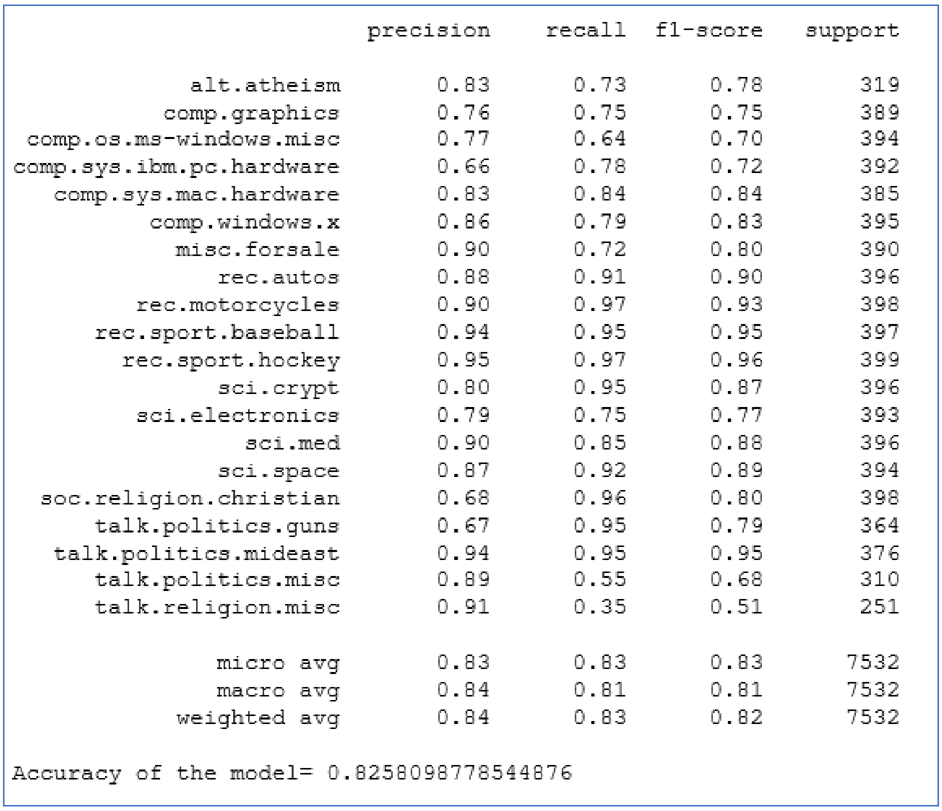
1. *Multinomial Naïve Bayes Model*

- It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a feature in a class is unrelated to the presence of any other feature.
- Multinomial Naive Bayes classifier is a specific instance of a Naive Bayes classifier which uses a multinomial distribution for each of the features
- Multinomial Naive Bayes simply lets us know that each p(fi|c) is a multinomial distribution, rather than some other distribution for class c and fi is probability of observing features. This works well for data which can easily be turned into counts, such as word counts in text. [1]
- In our code, we imported MultinomialNB from sklearn.naive_bayes and initialized alpha=0.1. We then run the Multinomial Naïve Bayes model on training and test data to measure the accuracy, precision, recall and F score. The results were obtained as under


Naive Bayes using TF/IDF
Series "Training" Point "Recall"
Value: 88.77
F Score
Recall
Precision
Accuracy
81.32
97.93
81.36
97.78
83.75
98.28
82.58
98.18
0
20
40
60
80
100
120
Testing
Training

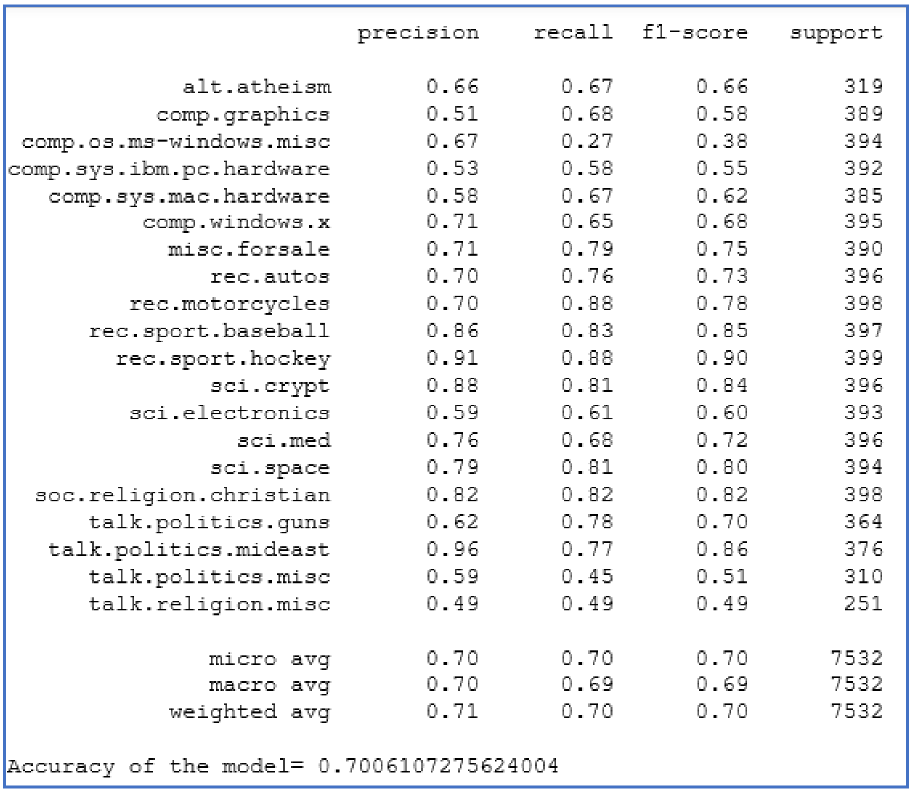

Naive Bayes using Count Vectorizer
F Score
Recall
Precision
Accuracy
69.07
84.98
69.38
84.99
70.27
85.51
70.06
85.07
0
10
20
30
40
50
60
70
80
90
Testing
Training

- The precision, recall, F score and support were calculated for each category of the target variable were obtained as under for TF/IDF:



```
                          precision    recall  f1-score   support

             alt.atheism       0.83      0.73      0.78       319
           comp.graphics       0.76      0.75      0.75       389
 comp.os.ms-windows.misc       0.77      0.64      0.70       394
comp.sys.ibm.pc.hardware       0.66      0.78      0.72       392
   comp.sys.mac.hardware       0.83      0.84      0.84       385
          comp.windows.x       0.86      0.79      0.83       395
            misc.forsale       0.90      0.72      0.80       390
               rec.autos       0.88      0.91      0.90       396
         rec.motorcycles       0.90      0.97      0.93       398
      rec.sport.baseball       0.94      0.95      0.95       397
        rec.sport.hockey       0.95      0.97      0.96       399
               sci.crypt       0.80      0.95      0.87       396
         sci.electronics       0.79      0.75      0.77       393
                 sci.med       0.90      0.85      0.88       396
               sci.space       0.87      0.92      0.89       394
  soc.religion.christian       0.68      0.96      0.80       398
      talk.politics.guns       0.67      0.95      0.79       364
   talk.politics.mideast       0.94      0.95      0.95       376
      talk.politics.misc       0.89      0.55      0.68       310
      talk.religion.misc       0.91      0.35      0.51       251

               micro avg       0.83      0.83      0.83      7532
               macro avg       0.84      0.81      0.81      7532
            weighted avg       0.84      0.83      0.82      7532

Accuracy of the model= 0.8258098778544876
```

- The precision, recall, F score and support were calculated for each category of the target variable were obtained as under for Count Vectorizer:

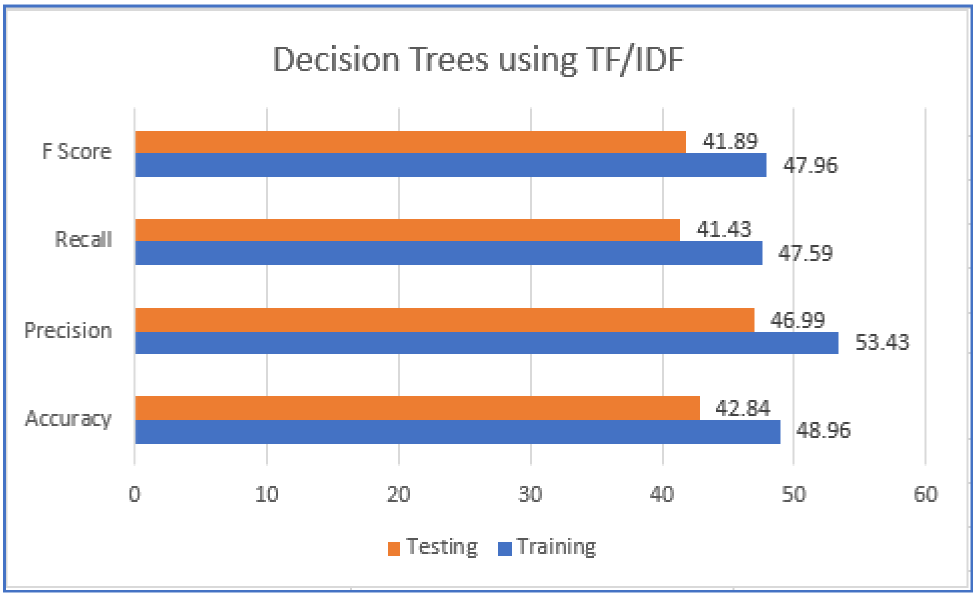


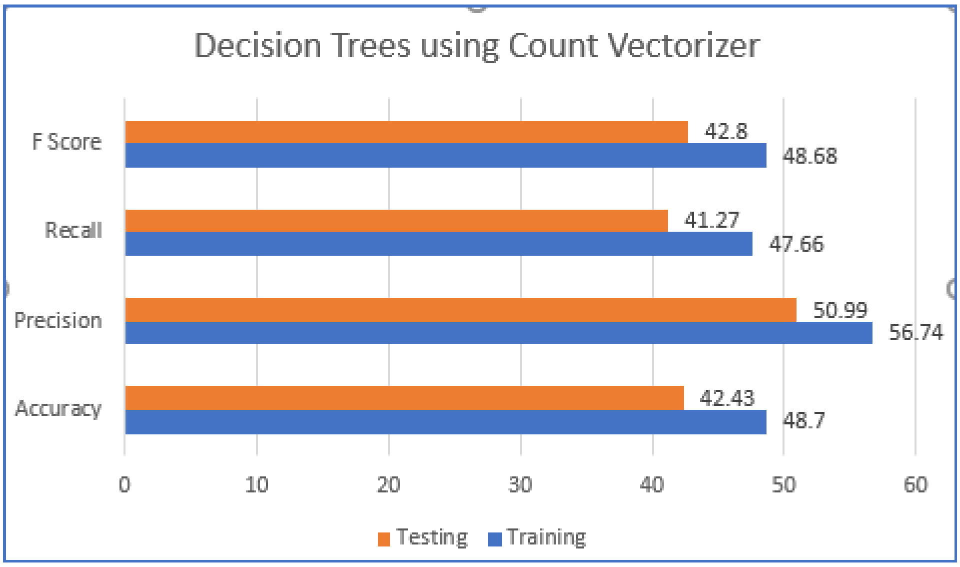
```
                          precision    recall  f1-score   support

             alt.atheism       0.66      0.67      0.66       319
           comp.graphics       0.51      0.68      0.58       389
 comp.os.ms-windows.misc       0.67      0.27      0.38       394
comp.sys.ibm.pc.hardware       0.53      0.58      0.55       392
   comp.sys.mac.hardware       0.58      0.67      0.62       385
          comp.windows.x       0.71      0.65      0.68       395
            misc.forsale       0.71      0.79      0.75       390
               rec.autos       0.70      0.76      0.73       396
         rec.motorcycles       0.70      0.88      0.78       398
      rec.sport.baseball       0.86      0.83      0.85       397
        rec.sport.hockey       0.91      0.88      0.90       399
               sci.crypt       0.88      0.81      0.84       396
         sci.electronics       0.59      0.61      0.60       393
                 sci.med       0.76      0.68      0.72       396
               sci.space       0.79      0.81      0.80       394
  soc.religion.christian       0.82      0.82      0.82       398
      talk.politics.guns       0.62      0.78      0.70       364
   talk.politics.mideast       0.96      0.77      0.86       376
      talk.politics.misc       0.59      0.45      0.51       310
      talk.religion.misc       0.49      0.49      0.49       251

               micro avg       0.70      0.70      0.70      7532
               macro avg       0.70      0.69      0.69      7532
            weighted avg       0.71      0.70      0.70      7532

Accuracy of the model= 0.7006107275624004
```

2. *Decision Trees*

- Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
- The result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision. [2]
- The topmost decision node in a tree which corresponds to the best predictor called root node.
- DecisionTreeClassifier is a class capable of performing multi-class classification on a dataset.
- As with other classifiers, DecisionTreeClassifier takes as input two arrays: an array X, sparse or dense, of size [n_samples, n_features] holding the training samples, and an array Y of integer values, size [n_samples], holding the class labels for the training samples.

- The accuracy measures obtained for the model are:

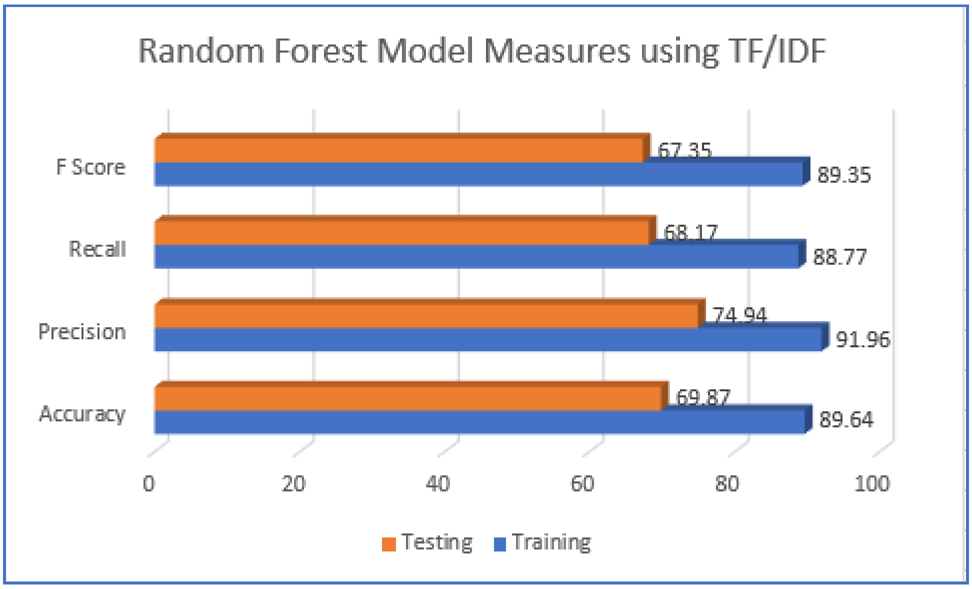


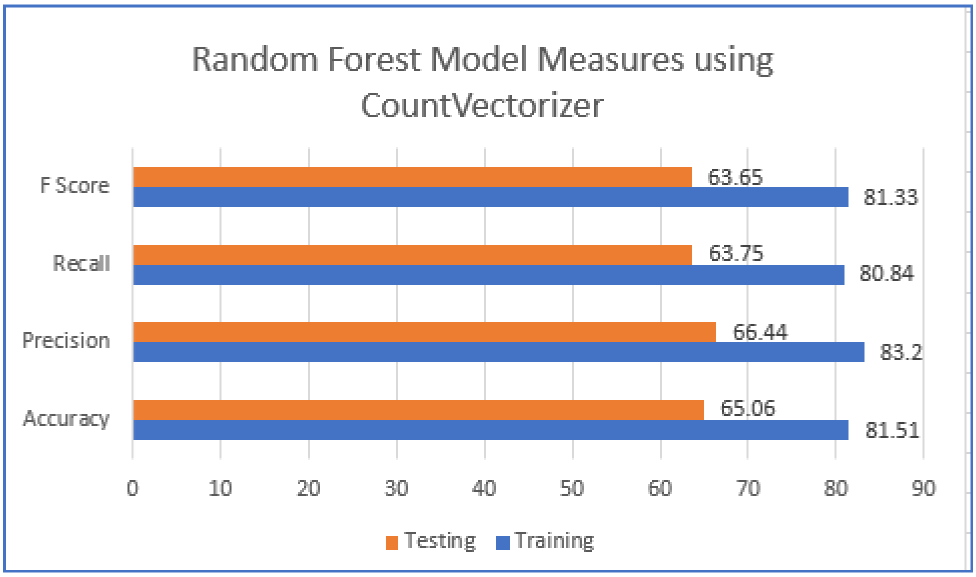


3. *Random Forest Classifier*

- Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees
- A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default) [3]
- We used the RandomForestClassifier from sklearn with 100 estimators

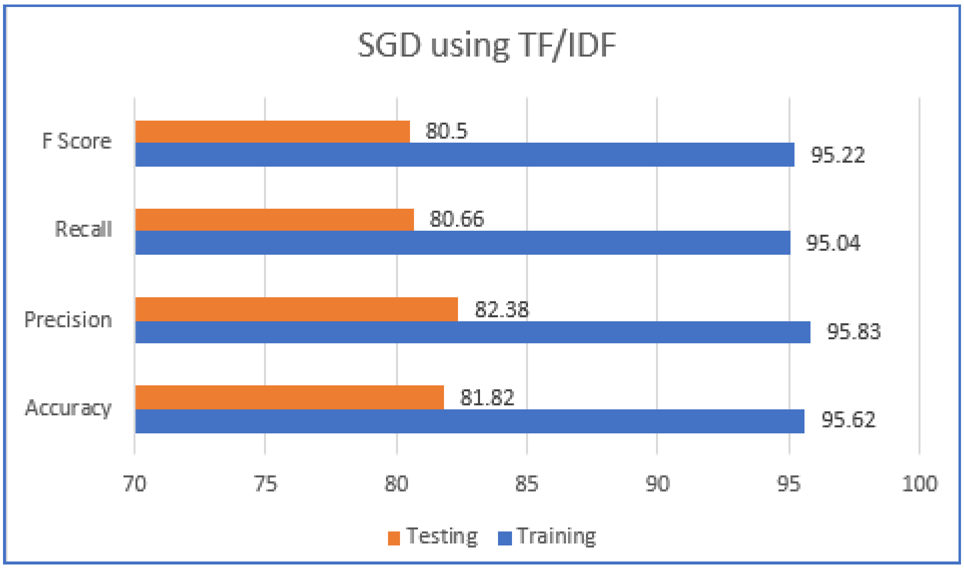
- The model measures are:

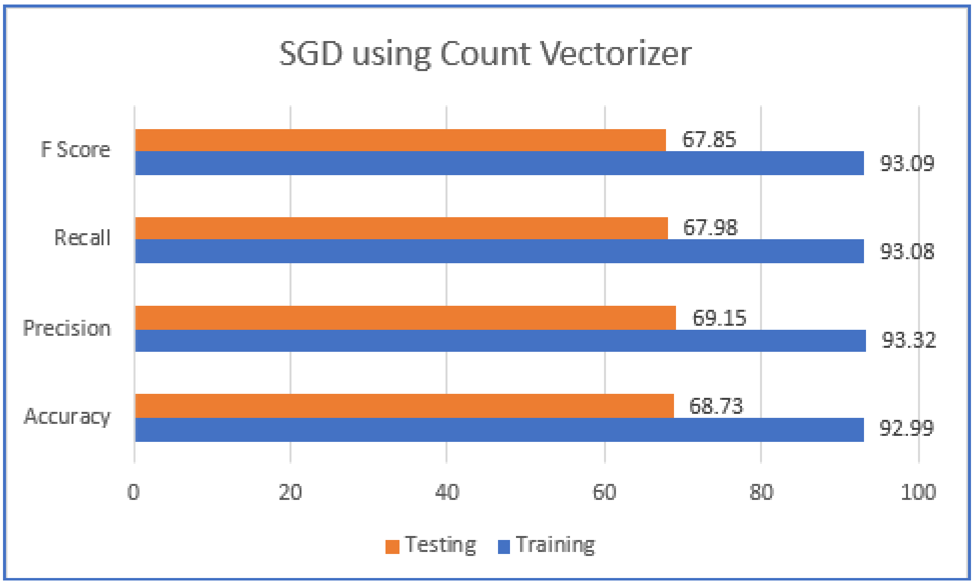




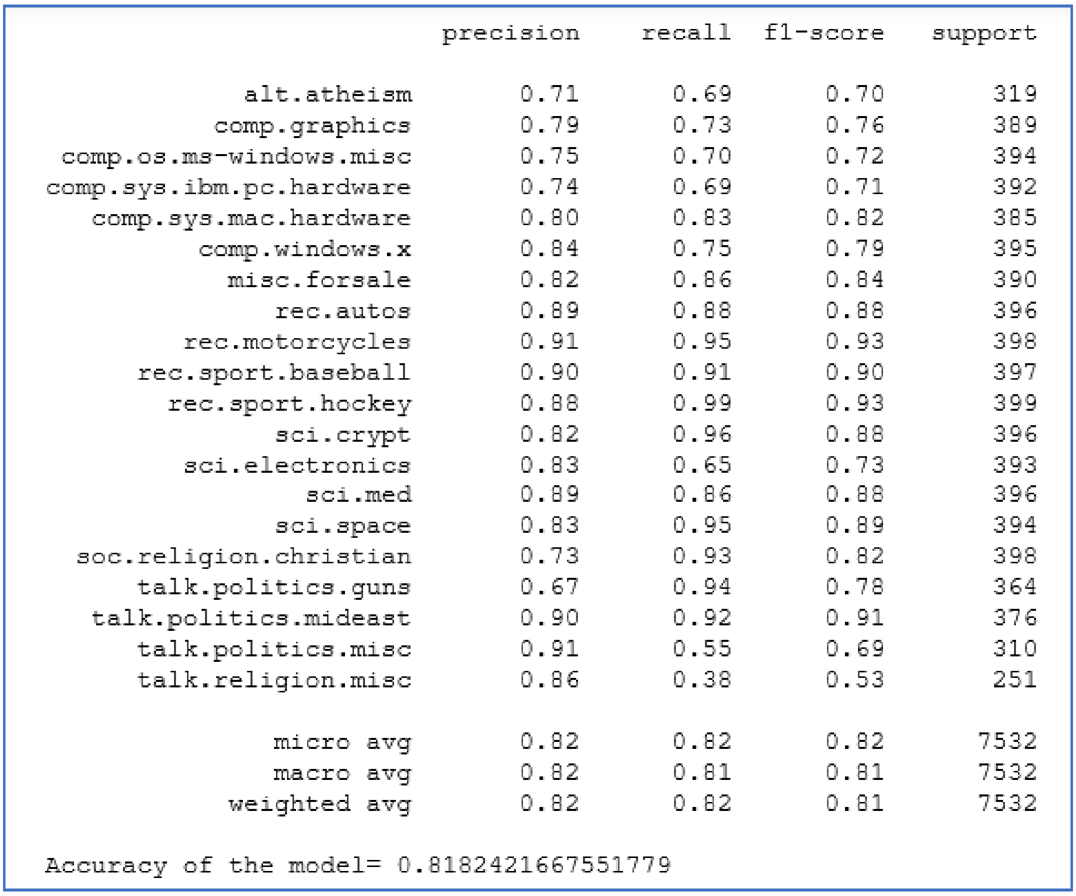
4. *Stochastic Gradient Descent*

- Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. [4]
- The class SGDClassifier implements a plain stochastic gradient descent learning routine which supports different loss functions and penalties for classification
- We use shuffle=True to shuffle the training data after each iteration.
- The model measures are:



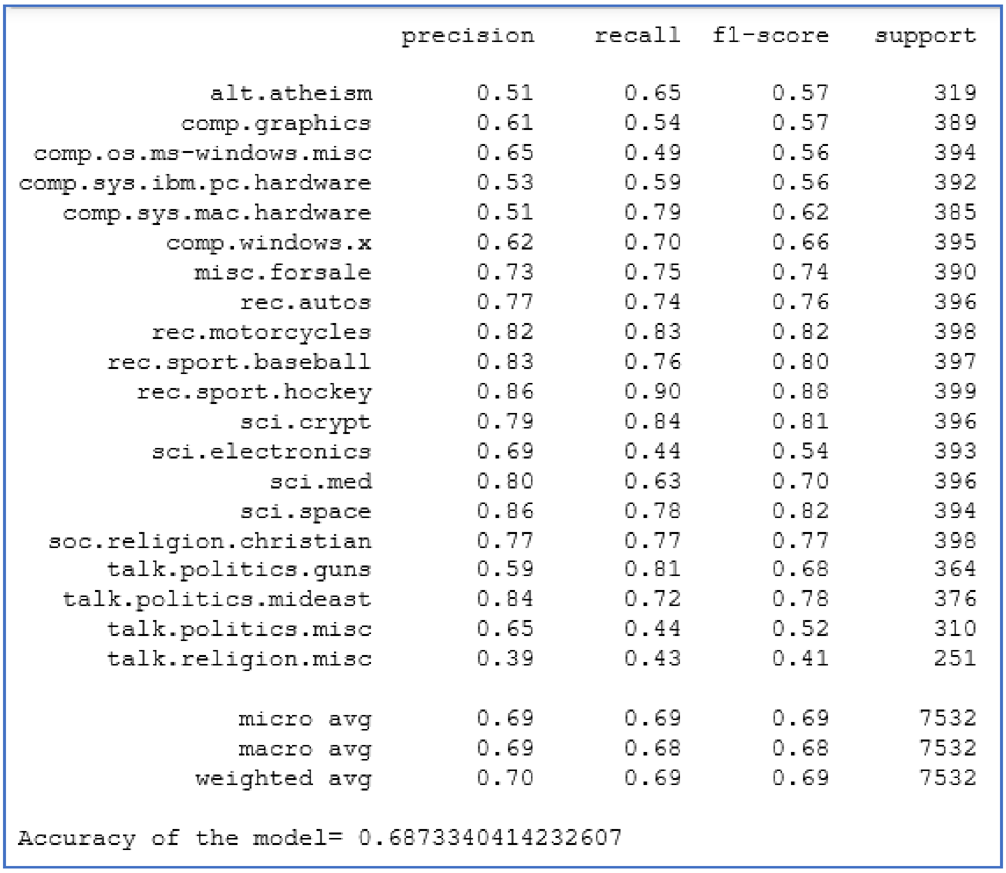


- The precision, recall, F score and support were calculated for each category of the target variable were obtained as under for TF/IDF



```
                          precision    recall  f1-score   support

             alt.atheism       0.71      0.69      0.70       319
           comp.graphics       0.79      0.73      0.76       389
 comp.os.ms-windows.misc       0.75      0.70      0.72       394
comp.sys.ibm.pc.hardware       0.74      0.69      0.71       392
   comp.sys.mac.hardware       0.80      0.83      0.82       385
          comp.windows.x       0.84      0.75      0.79       395
            misc.forsale       0.82      0.86      0.84       390
               rec.autos       0.89      0.88      0.88       396
         rec.motorcycles       0.91      0.95      0.93       398
      rec.sport.baseball       0.90      0.91      0.90       397
        rec.sport.hockey       0.88      0.99      0.93       399
               sci.crypt       0.82      0.96      0.88       396
         sci.electronics       0.83      0.65      0.73       393
                 sci.med       0.89      0.86      0.88       396
               sci.space       0.83      0.95      0.89       394
  soc.religion.christian       0.73      0.93      0.82       398
      talk.politics.guns       0.67      0.94      0.78       364
   talk.politics.mideast       0.90      0.92      0.91       376
      talk.politics.misc       0.91      0.55      0.69       310
      talk.religion.misc       0.86      0.38      0.53       251

               micro avg       0.82      0.82      0.82      7532
               macro avg       0.82      0.81      0.81      7532
            weighted avg       0.82      0.82      0.81      7532

Accuracy of the model= 0.8182421667551779
```

- The precision, recall, F score and support were calculated for each category of the target variable were obtained as under for TF/IDF:

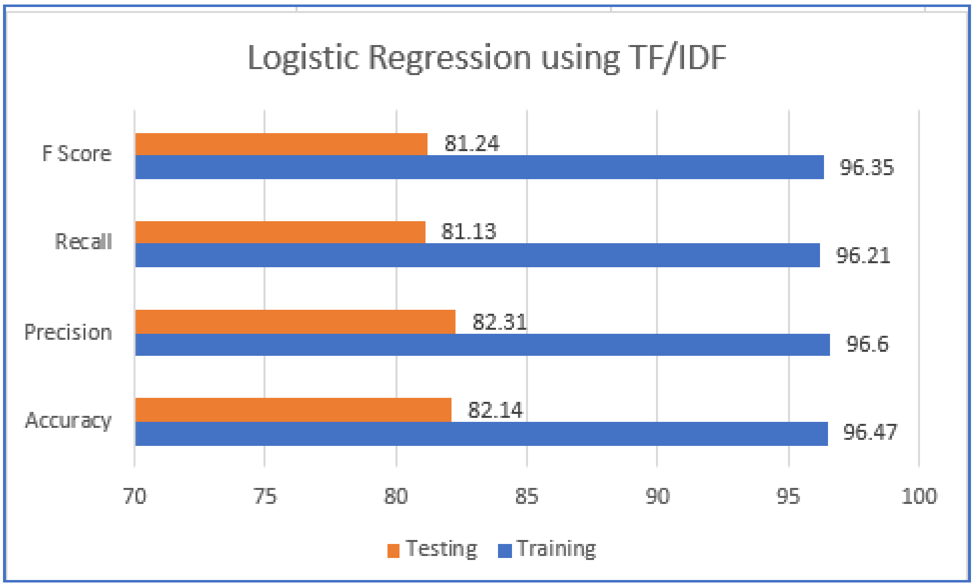


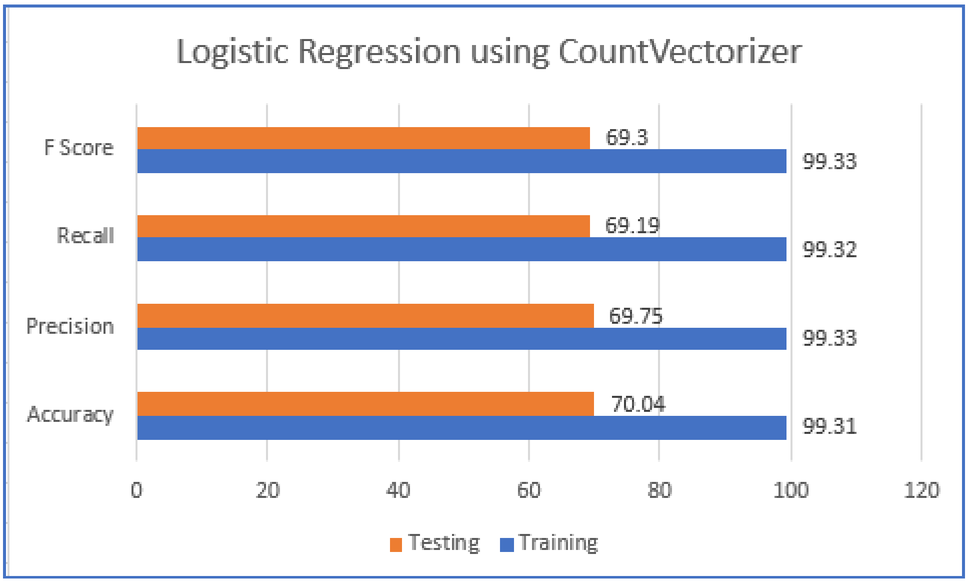
```
                          precision    recall  f1-score   support

             alt.atheism       0.51      0.65      0.57       319
           comp.graphics       0.61      0.54      0.57       389
 comp.os.ms-windows.misc       0.65      0.49      0.56       394
comp.sys.ibm.pc.hardware       0.53      0.59      0.56       392
   comp.sys.mac.hardware       0.51      0.79      0.62       385
          comp.windows.x       0.62      0.70      0.66       395
            misc.forsale       0.73      0.75      0.74       390
               rec.autos       0.77      0.74      0.76       396
         rec.motorcycles       0.82      0.83      0.82       398
      rec.sport.baseball       0.83      0.76      0.80       397
        rec.sport.hockey       0.86      0.90      0.88       399
               sci.crypt       0.79      0.84      0.81       396
         sci.electronics       0.69      0.44      0.54       393
                 sci.med       0.80      0.63      0.70       396
               sci.space       0.86      0.78      0.82       394
  soc.religion.christian       0.77      0.77      0.77       398
      talk.politics.guns       0.59      0.81      0.68       364
   talk.politics.mideast       0.84      0.72      0.78       376
      talk.politics.misc       0.65      0.44      0.52       310
      talk.religion.misc       0.39      0.43      0.41       251

               micro avg       0.69      0.69      0.69      7532
               macro avg       0.69      0.68      0.68      7532
            weighted avg       0.70      0.69      0.69      7532

Accuracy of the model= 0.6873340414232607
```

5. *Logistic Regression*

- It is a statistical method for analyzing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).
- The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables.
- We imported LogisticRegression from scikit learn. This class implements regularized logistic regression using the ‘liblinear’ library, ‘newton-cg’, ‘sag’, ‘saga’ and ‘lbfgs’ solvers. The regularization is applied by default. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted [5]

- The accuracy measures are:

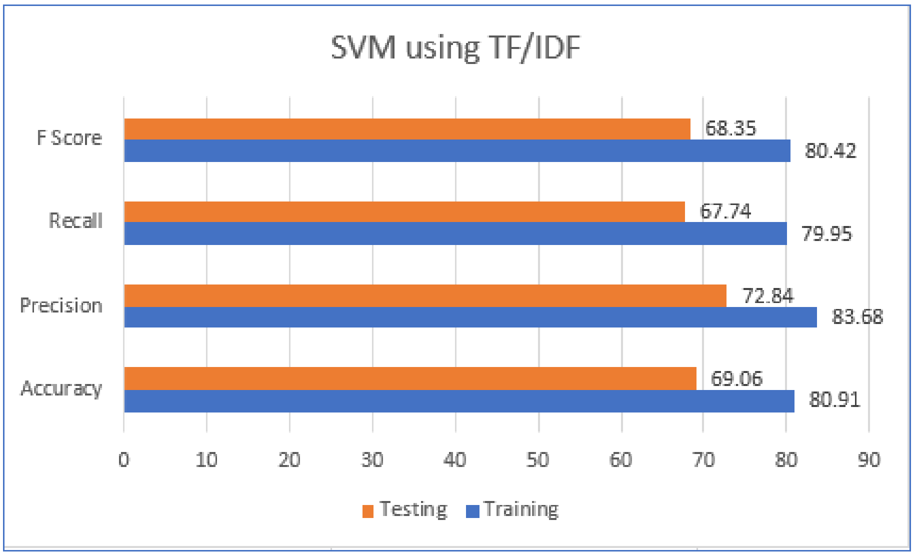


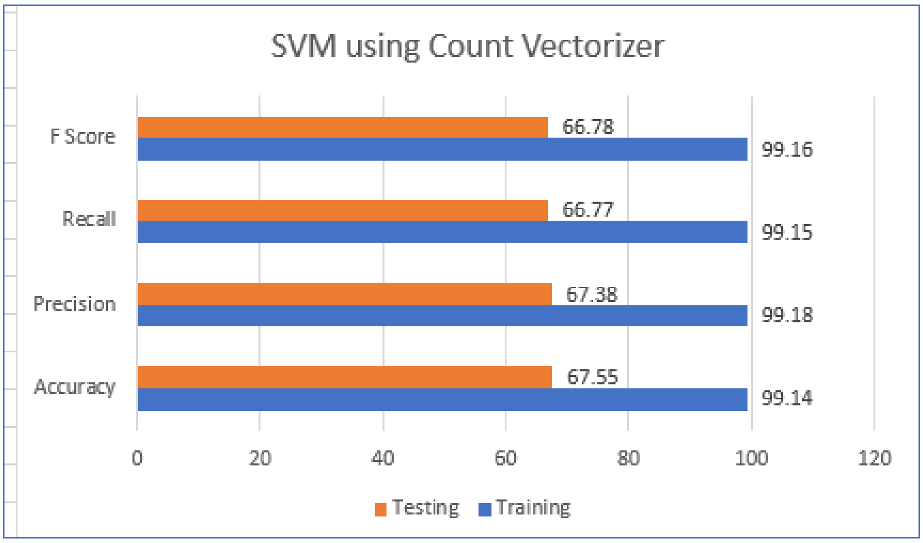


6. *Support Vector Machines*

- A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side [6]
- The support vector machines in scikit-learn support both dense (numpy.ndarray and convertible to that by numpy.asarray) and sparse (any scipy.sparse) sample vectors as input. However, to use an SVM to make predictions for sparse data, it must have been fit on such data.

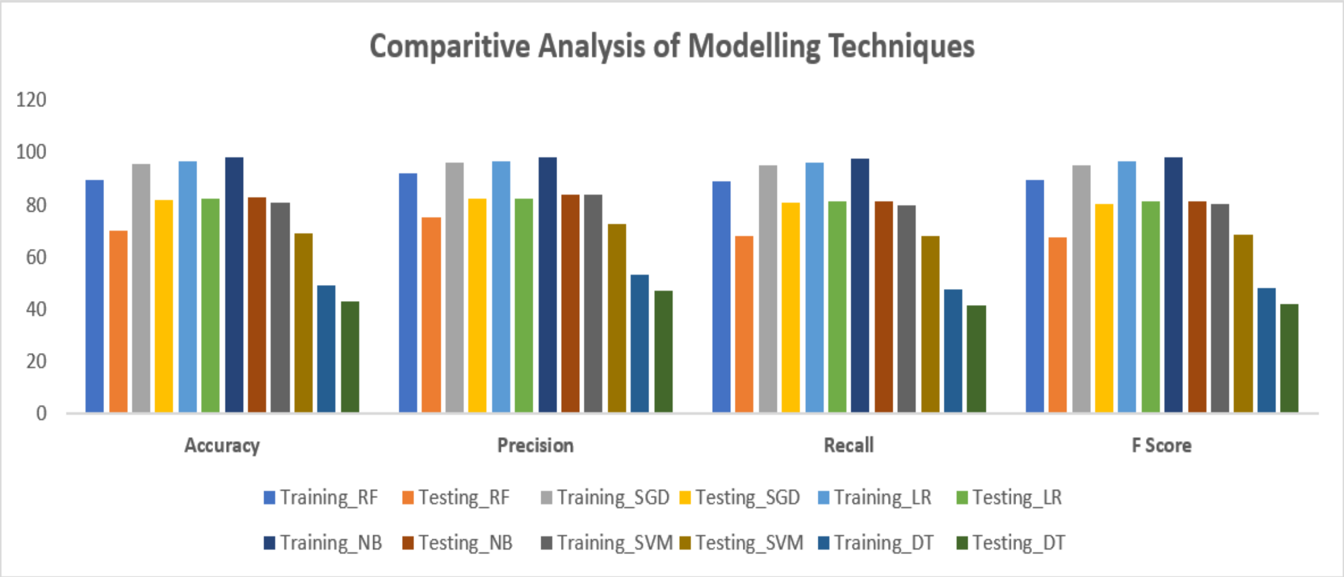
- The model measures are:

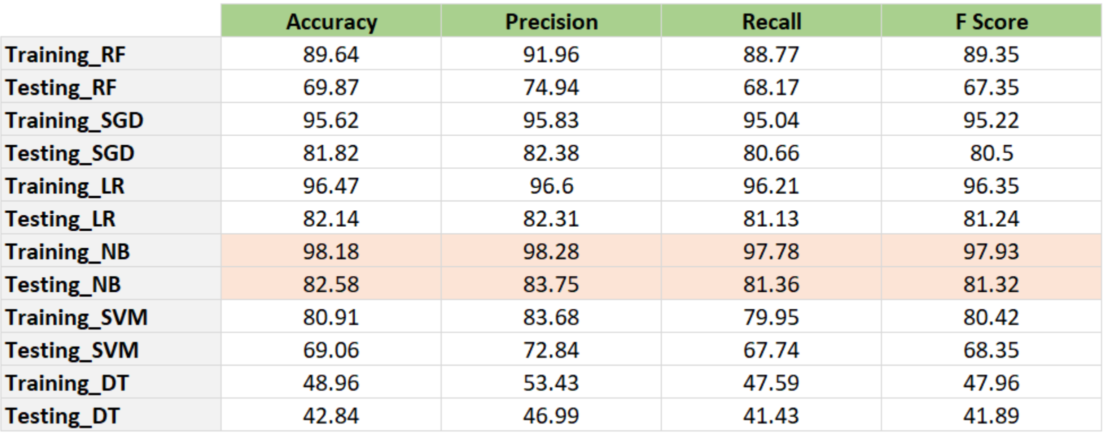




VI. RESULT AND ANALYSIS

We plotted the measures to evaluate the best model





	Accuracy	Precision	Recall	F Score
Training_RF	89.64	91.96	88.77	89.35
Testing_RF	69.87	74.94	68.17	67.35
Training_SGD	95.62	95.83	95.04	95.22
Testing_SGD	81.82	82.38	80.66	80.5
Training_LR	96.47	96.6	96.21	96.35
Testing_LR	82.14	82.31	81.13	81.24
Training_NB	98.18	98.28	97.78	97.93
Testing_NB	82.58	83.75	81.36	81.32
Training_SVM	80.91	83.68	79.95	80.42
Testing_SVM	69.06	72.84	67.74	68.35
Training_DT	48.96	53.43	47.59	47.96
Testing_DT	42.84	46.99	41.43	41.89

As evident from the graph, out of the 6 models the ***best model is that of Naïve Bayes***. The model doesn’t overfit and gives best testing and training accuracy.

VII. CONCLUSION

We observed that for almost all the models performed well for TFIDF as compared to Count Vectorizer except for logistic regression where count vectorizer wins but at the cost of precision and recall values. The 6 models tried are as follows:

1. Naive Bayes
2. SGD Classifier
3. Support Vector Machines
4. Random Forest
5. Logistic Regression
6. Decision Tree

From the above models tried we observed that Naive Bayes Classifier using TFIDF performs best closely followed by SGD Classifier. Though both these techniques have good performance, while looking at the precision and recall of the individual categories.

Thus to conclude text classification is a very interesting area of unstructured data in machine learning and can be explored in depth to understand which techniques performs best and at what circumstances. Several popular ones were explored in this project like Naive Bayes and Support Vector Machines and SGD classifier.

VIII. REFERENCES

[1] C.D. Manning, P. Raghavan and H. Schuetze (2008). Introduction to Information Retrieval. Cambridge University Press, pp. 234-265. https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html

[2]https://medium.com/@Mandysidana/machine-learning-types-of-classification-9497bd4f2e14

[3]https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifie.html

[4] https://developers.google.com/machine-learning/crash-course/reducing-loss/stochastic-gradient-descent

[5]https://scikitlearn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[6]https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72