SENTIMENT ANALYSIS

PROJECT REPORT

MACHINE LEARNING AND NATURAL LANGUAGE PROCESSING

Submitted by

DEEPANSHU SAINI (19103105)
HARSH CHAUHAN (19103296)
JATIN KANSAL (19103270)

Submitted to

Dr. Himanshu Sekhar

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

Abstract

In this report, we address the problem of sentiment classification on Election dataset. We use a number of machine learning methods to perform sentiment analysis. In the end, we presented the outcome using a confusion matrix for all the used machine learning algorithms.

Problem Statement

Twitter is a popular social networking website where members create and interact with messages known as “tweets”. This serves as a means for individuals to express their thoughts or feelings about different subjects. Various different parties such as consumers and marketers have done sentiment analysis on such tweets to gather insights into products or to conduct market analysis. Furthermore, with the recent advancements in machine learning algorithms, we are able to improve the accuracy of our sentiment analysis predictions. In this report, we will attempt to conduct sentiment analysis on “tweets” using various different machine learning algorithms. We attempt to classify the polarity of the tweet where it is either positive or negative. If the tweet has both positive and negative elements, the more dominant sentiment should be picked as the final label. We use the dataset from Kaggle which was crawled and labeled positive/negative. The data provided comes with emoticons, usernames and hashtags which are required to be processed and converted into a standard form. We use various machine learning algorithms to conduct sentiment analysis using the extracted features. However, just relying on individual models did not give a high accuracy so we did a comparative study using multiple models. Finally, we report our experimental results and findings at the end.

TOOLS AND TECHNOLOGIES

Kaggle- dataset
Python
Matplotlib
SNS
Seaborn
Collections
sklearn

CONCEPTS USED

NAIVE BAYES:

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification tasks. The crux of the classifier is based on the Bayes theorem.

FORMULA:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is,

the presence of one particular feature does not affect the other. Hence it is called naive.

Multinomial Naive Bayes:

This is mostly used for document classification problems, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

GRADIENT BOOSTING CLASSIFIER:

This algorithm builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage `n_classes_` regression trees fit on the negative gradient of the loss function, e.g. binary or multiclass log loss. Binary classification is a special case where only a single regression tree is induced.

RANDOM FOREST:

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance

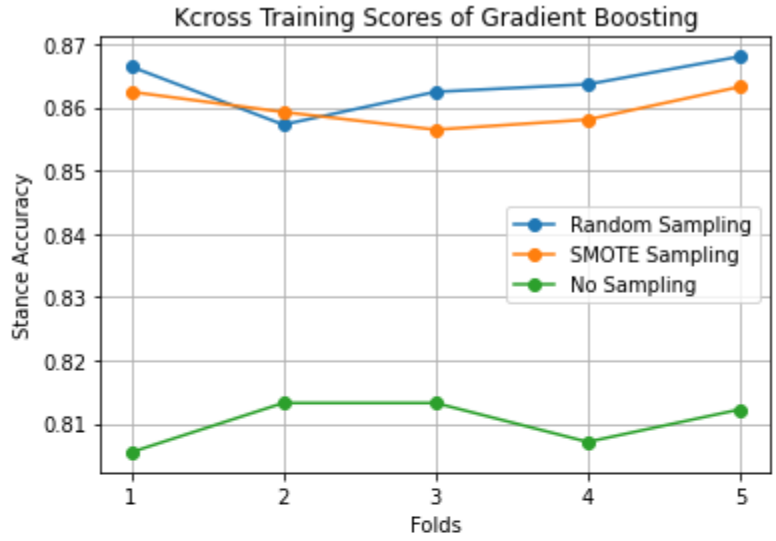
SUPPORT VECTOR MACHINE:

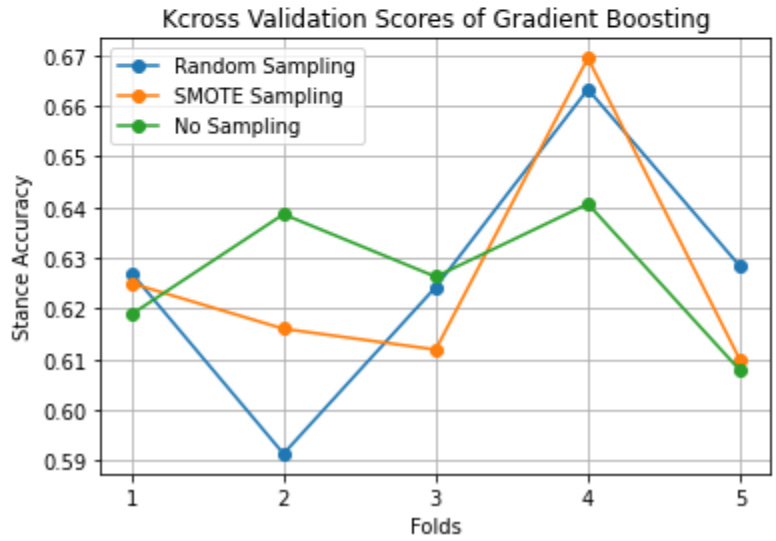
Support vector machines (**SVMs**, also **support vector networks**[1]) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir Vapnik with colleagues. SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik (1982, 1995) and Chervonenkis (1974). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. SVM maps training examples to points in space so as to maximize the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Output:-

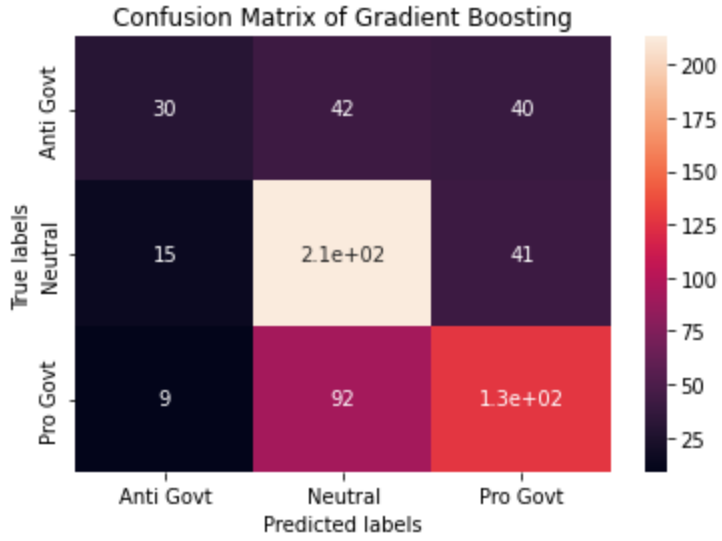
Gradient boosting classifier:

```
Random sampling scores
AVERAGE TRAINING SCORE 0.8635566188197767
AVERAGE TESTING SCORE 0.6268472077288181
smote sampling scores
AVERAGE TRAINING SCORE 0.8598883572567783
AVERAGE TESTING SCORE 0.6264373716632443
Normal scores
AVERAGE TRAINING SCORE 0.8102417076940419
AVERAGE TESTING SCORE 0.6264398963207325
```



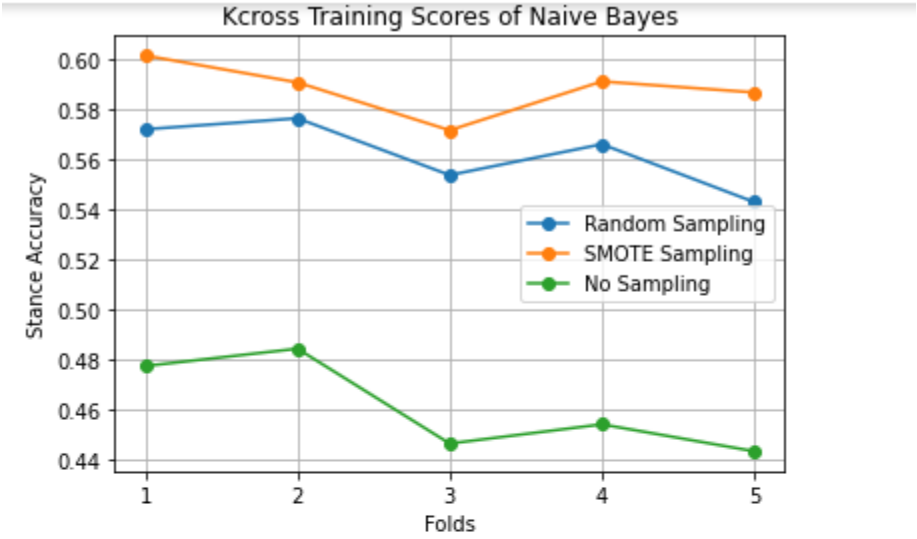


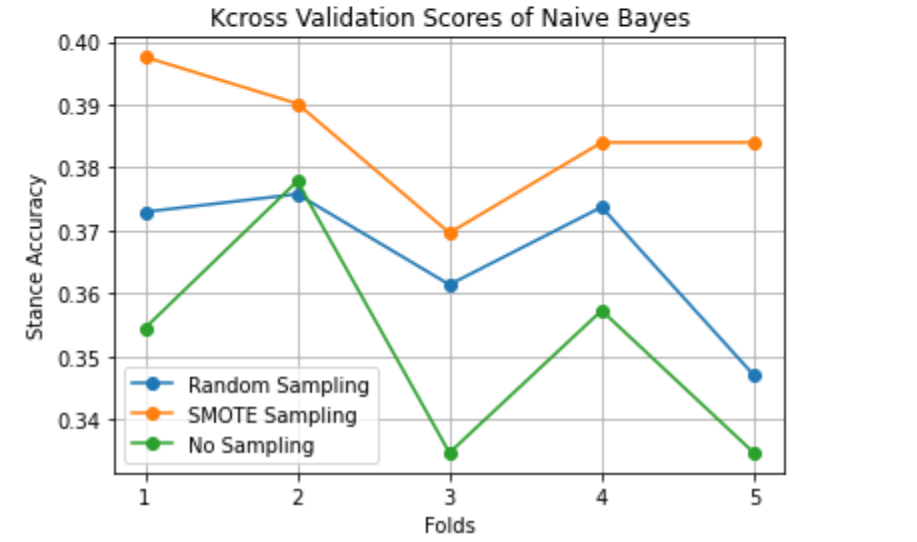
```
Y_value is STANCE
Training accuracy 0.8308702791461412
Training accuracy 0.6075533661740559
```



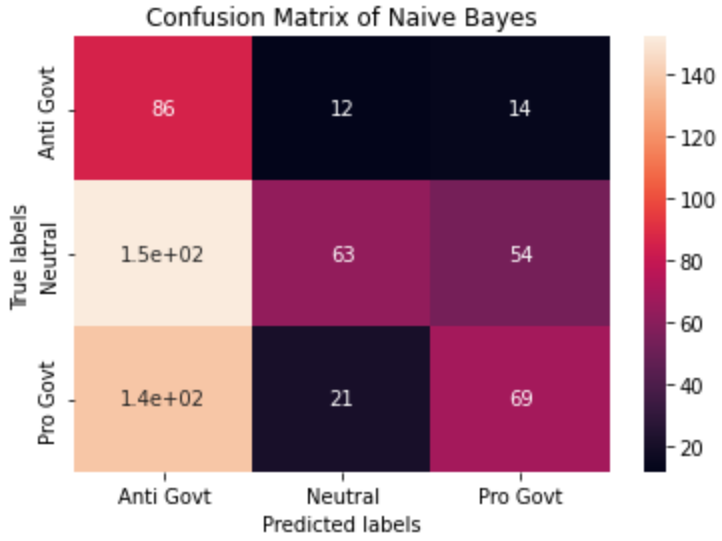
Naive bayes:

```
Random sampling scores
AVERAGE TRAINING SCORE 0.5623604465709728
AVERAGE TESTING SCORE 0.3661712727639951
smote sampling scores
AVERAGE TRAINING SCORE 0.5885167464114833
AVERAGE TESTING SCORE 0.38505234456525395
Normal scores
AVERAGE TRAINING SCORE 0.4611059428148801
AVERAGE TESTING SCORE 0.3518051301040159
```



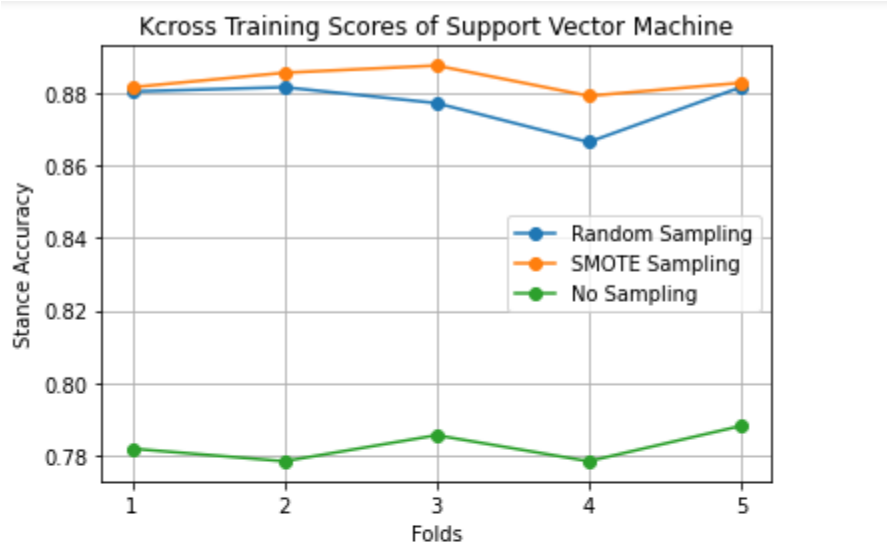


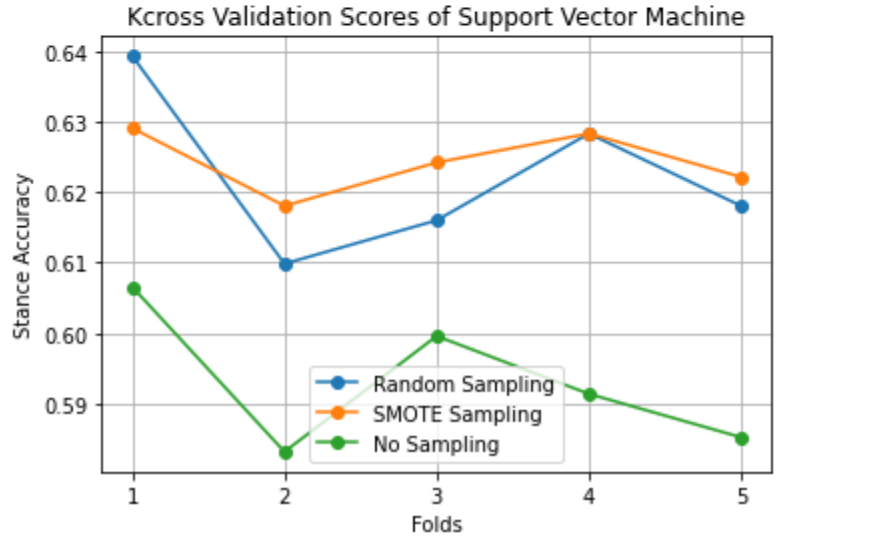
```
Y_value is STANCE
Training accuracy 0.4783798576902025
Training accuracy 0.3579638752052545
```



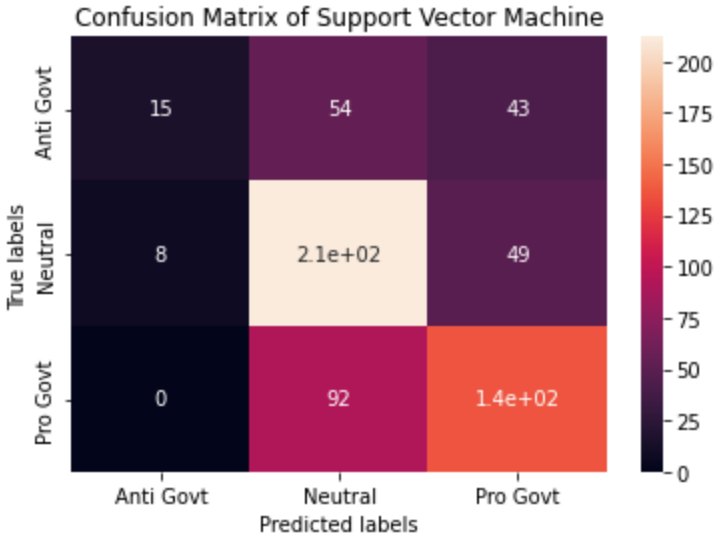
Support Vector machine:

```
Random sampling scores
AVERAGE TRAINING SCORE 0.8774322169059012
AVERAGE TESTING SCORE 0.6223247046150739
smote sampling scores
AVERAGE TRAINING SCORE 0.8833333333333334
AVERAGE TESTING SCORE 0.6243823004679031
Normal scores
AVERAGE TRAINING SCORE 0.7824301516177937
AVERAGE TESTING SCORE 0.5931800585720538
```



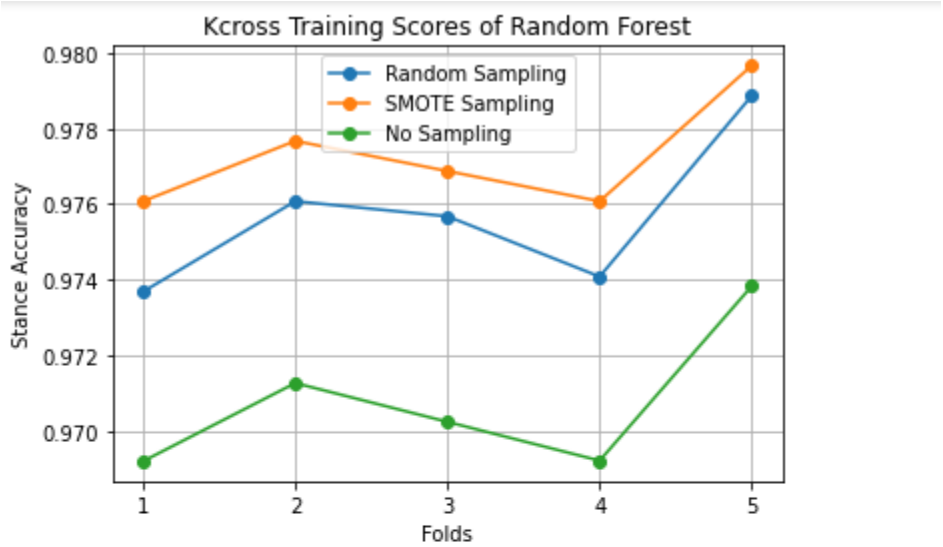


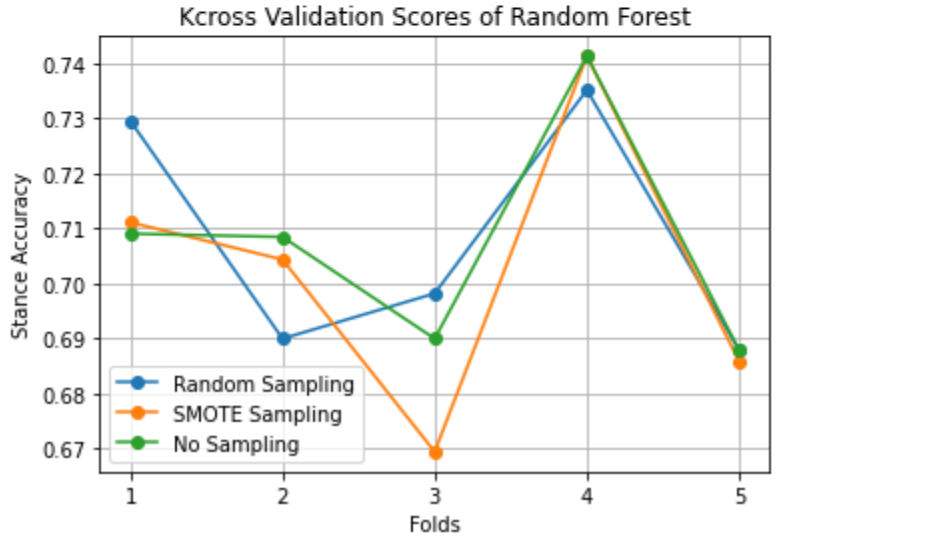
```
Y_value is STANCE
Training accuracy 0.7952928297755884
Training accuracy 0.5960591133004927
```



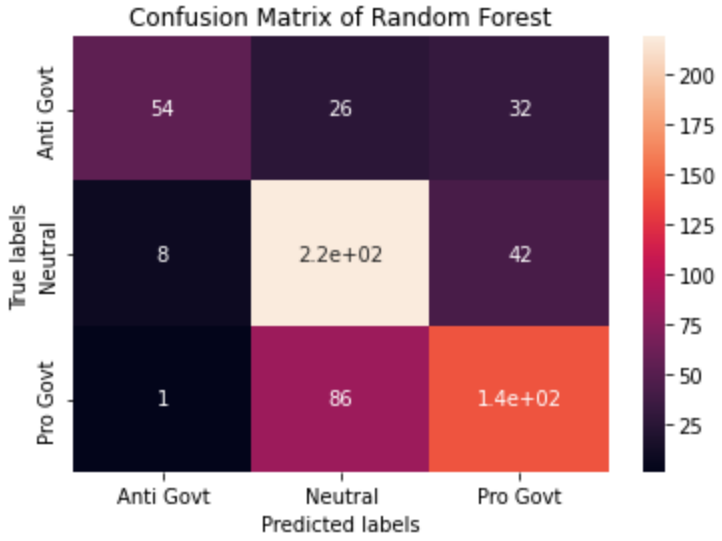
Random Forest:

```
Random sampling scores
AVERAGE TRAINING SCORE 0.9756778309409888
AVERAGE TESTING SCORE 0.7081192984818394
smote sampling scores
AVERAGE TRAINING SCORE 0.9772727272727273
AVERAGE TESTING SCORE 0.7023773858013262
Normal scores
AVERAGE TRAINING SCORE 0.9707510722605075
AVERAGE TESTING SCORE 0.7073063587706601
```

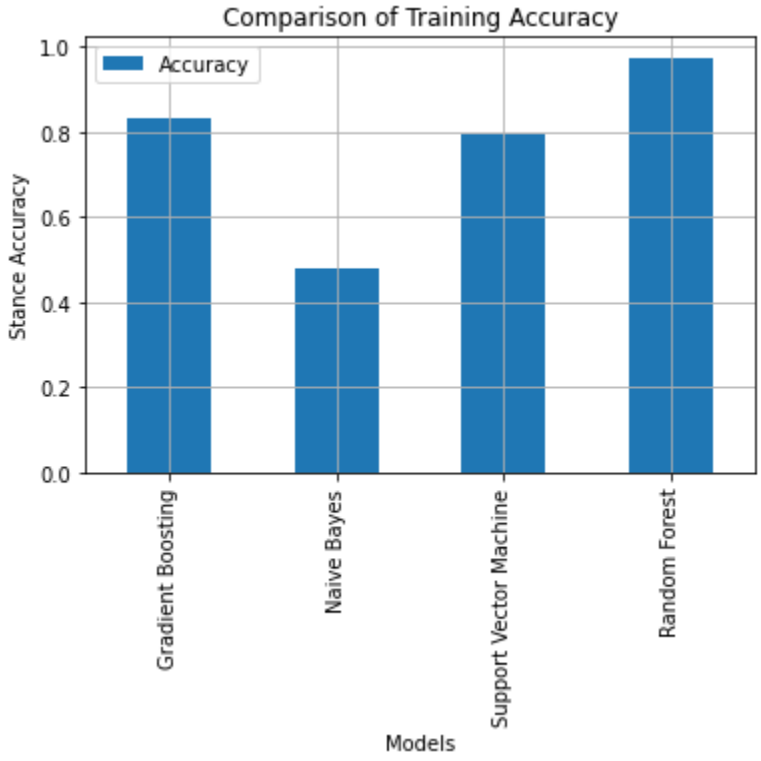


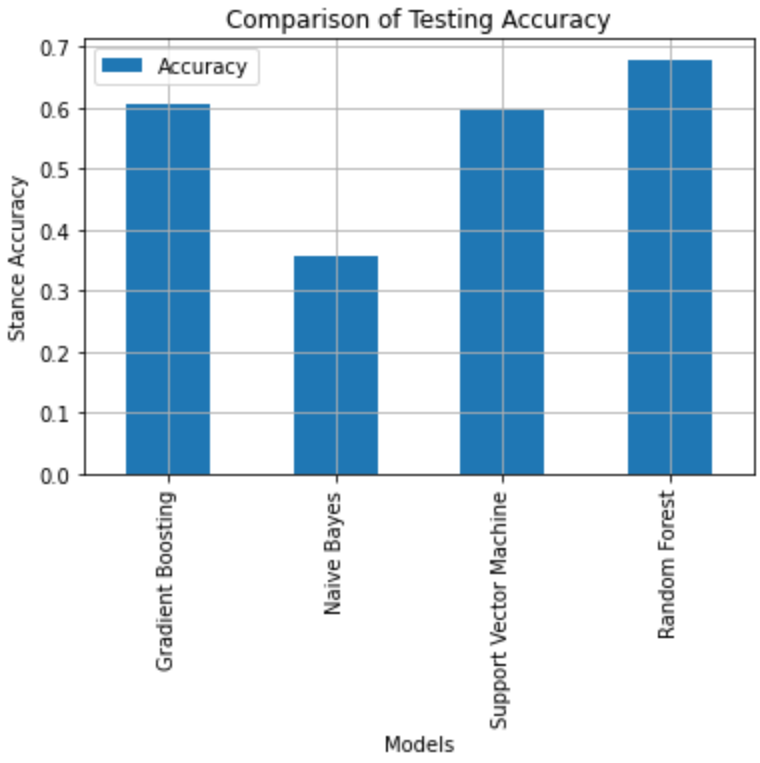


```
Y_value is STANCE
Training accuracy 0.9753694581280788
Training accuracy 0.6798029556650246
```



Accuracy:





Conclusion:

Random Forest classifiers provided highest training and testing accuracy with 97.5% and 70.8% respectively.

Future directions

Handling emotion ranges: We can improve and train our models to handle a range of sentiments. Tweets don't always have positive or negative sentiment. At times they may have no sentiment i.e. neutral. Sentiment can also have gradations like the sentence, This is good, is positive but the sentence, This is extraordinary. is somewhat more positive than the first. We can therefore classify the sentiment in ranges, say from -2 to +2.