A Comparative Study on Machine Learning Algorithms for Predicting Sentiment Analysis of Movie Reviews

Ms. Shraddha Rajesh Rai [1], Dr Ramesh Solanki[2]

[1] *Department of MCA & Vivekanand Education Society's Institute of technology, Mumbai, India*

[2] *Department of MCA & Vivekanand Education Society's Institute of technology, Mumbai, India*

---***---

Abstract - The aim of this study is to compare of the most popular models available for Sentimental analysis, which is a very popular framework for analyzing the movie reviews. Choosing an analysis model is necessary and can be difficult given the surplus of choices for Sentimental analysis, as it is used more than one model at a time to take advantage hide disadvantage of some models. The selection of a good Analysis model will provide effortless performance of models in the system to deliver the best result. In this paper, we will see the various aspects of these six models for analyzing accuracy of model. The comparison between the six model will be done based on various parameters that can help analyzer decide which model will be better suited for different aspects.

Key Words: Sentiment Analysis, Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Linear SVC, Bernoulli NB, K Neighbors Classifier.

1.INTRODUCTION

Sentiment analysis is the analysis of emotions and opinions from any form of text. Sentiment analysis is also termed as opinion mining. Sentiment analysis of the data is very useful to express the opinion of the mass or group or any individual. Sentiment is often framed as a binary distinction (positive vs. negative), but it can also be a more fine-grained, like identifying the specific emotion an author is expressing (like fear, joy or anger).

Sentiment analysis is used for many applications. Some examples of applications for sentiment analysis include: Shifts in sentiment on social media have been shown to correlate with shifts in the stock market. It is an integral part of market research and customer service approach. It can be useful to quickly summarize some qualities of text, especially if we have so much text that a human reader cannot analyze all of it.

There are following phases of Sentiment Analysis:

Pre-Processing Phase: The data is first cleaned to reduce noise.

Feature Extraction: A token is given to the keywords andthis token is now put under analysis.

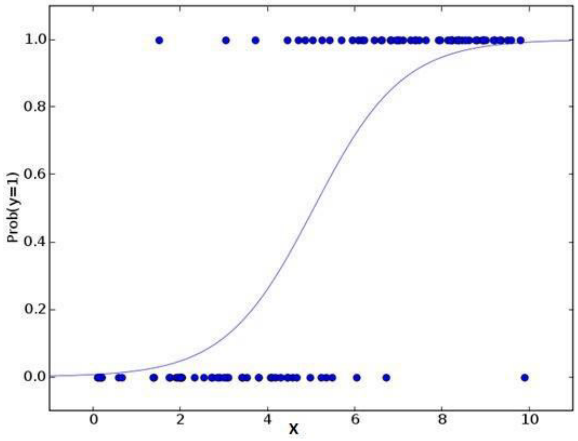
Classification Phase: Based on different algorithms these keywords are put under certain category.

Regression Model:

Linear and Logistic regressions are usually the first algorithms people learn in data science. Due to their popularity, a lot of analysts even end up thinking that they are the only form of regressions. The ones who are slightly more involved think that they are the most important among all forms of regression analysis.

Logistic Regression:

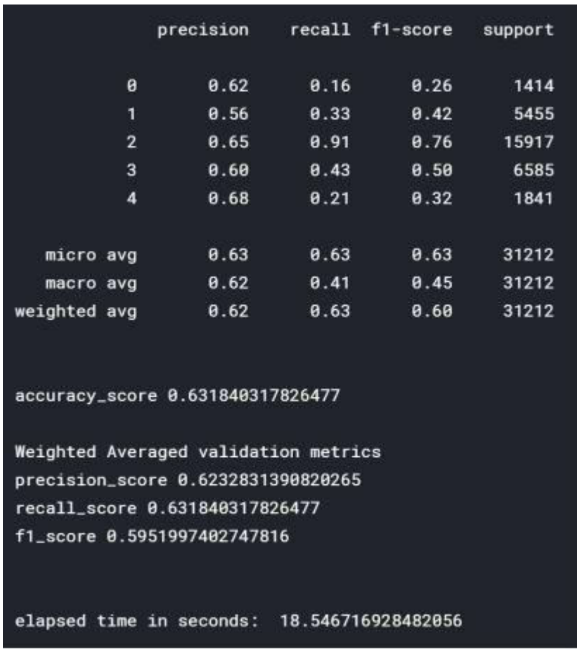
Logistic regression is used to find the probability of event=Success and event=Failure. Logistic Regression is categorized as a classification algorithm. Mostly, it is used to predict a binary outcome (like 0 / 1, False / True, No / Yes, Wrong / Right) when a set of independent variables is given.



Important Points:

- It is widely used for classification problems
- Logistic regression doesn't require linear relationship between dependent and independent variables. It can handle various types of relationships because it applies a non-linear log transformation to the predicted odds ratio
- To avoid overfitting and underfitting, we should include all significant variables. A good approach to ensure this practice is to use a step wise method to estimate the logistic regression
- It requires large sample sizes because maximum likelihood estimates are less powerful at low sample sizes than ordinary least square
- If the dependent variable is multi class then it is known as Multinomial Logistic regression.

Model Accuracy:



```
              precision    recall  f1-score   support

           0       0.62      0.16      0.26      1414
           1       0.56      0.33      0.42      5455
           2       0.65      0.91      0.76     15917
           3       0.60      0.43      0.50      6585
           4       0.68      0.21      0.32      1841

   micro avg       0.63      0.63      0.63     31212
   macro avg       0.62      0.41      0.45     31212
weighted avg       0.62      0.63      0.60     31212


accuracy_score 0.631840317826477

Weighted Averaged validation metrics
precision_score 0.6232831390820265
recall_score 0.631840317826477
f1_score 0.5951997402747816


elapsed time in seconds:  18.546716928482056
```

Result:

Figure 1 shows that: Accuracy Score, Weighted Average, Precision score, Recall Score & f1 Score, elapsed Time in sec of Dataset.

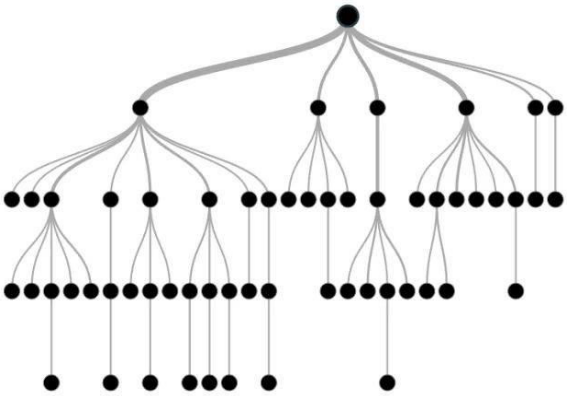
In which Logistic Regression is highly Accurate with less elapsed time.

CART Models:

Decision Tree Classifier:

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map nonlinear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression).

How does it work: Decision tree is a type of supervised learning algorithm (having a predefined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.

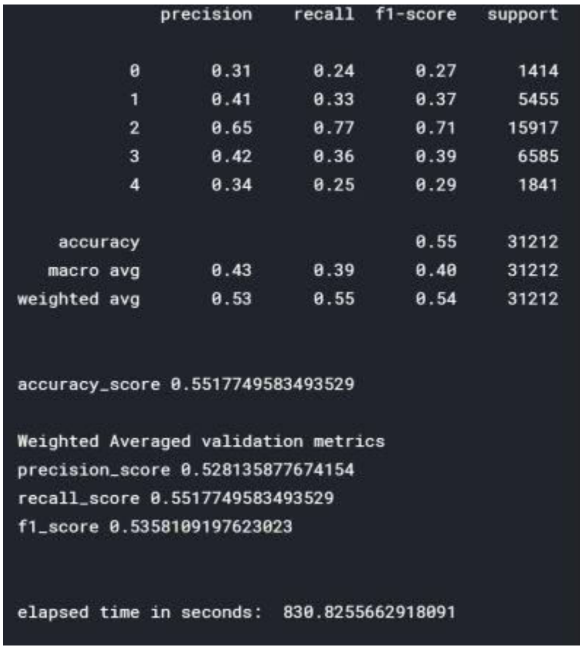


Important Terminology related to Decision Trees:

1. **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:** It is a process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
4. **Leaf/ Terminal Node:** Nodes do not split is called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
6. **Branch / Sub-Tree:** A subsection of entire tree is called branch or sub-tree.

7. **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

Model Accuracy:



```
              precision    recall  f1-score   support

           0       0.31      0.24      0.27      1414
           1       0.41      0.33      0.37      5455
           2       0.65      0.77      0.71     15917
           3       0.42      0.36      0.39      6585
           4       0.34      0.25      0.29      1841

    accuracy                           0.55     31212
   macro avg       0.43      0.39      0.40     31212
weighted avg       0.53      0.55      0.54     31212


accuracy_score 0.5517749583493529

Weighted Averaged validation metrics
precision_score 0.528135877674154
recall_score 0.5517749583493529
f1_score 0.5358109197623023


elapsed time in seconds:  830.8255662918091
```

Result:

Figure 2 shows that: Accuracy Score, Weighted Average, Precision score, Recall Score & f1 Score, elapsed Time in sec of Dataset.
In which Decision Tree Classifier is less Accurate with very high elapsed time.

Bagging Trees:

Random forest classifier:

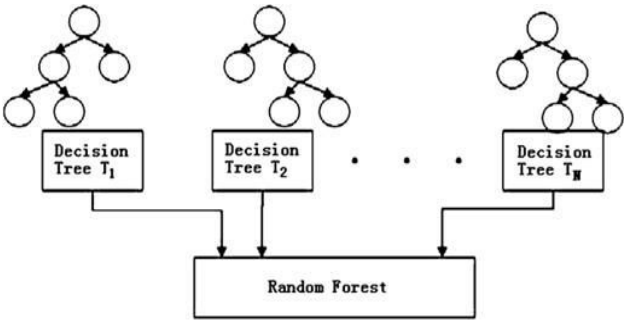
Random forest algorithm is an ensemble classification algorithm. Ensemble classifier means a group of classifiers. Instead of using only one classifier to predict the target, In ensemble, we use multiple classifiers to predict the target.

In case of random forest, these ensemble classifiers are the randomly created decision trees. Each decision tree is a single classifier and the target prediction is based on the majority voting method.

The majority voting concept is same as the political voting. Each person votes per one political party out all the political parties participating in elections. In the same way, every classifier will vote to one target class out of all the target classes.

To declare the election results. The votes will calculate and the party which got the greatest number of votes treated as the election winner. In the same way, the target class which got the greatest number of votes considered as the final predicted target class.
Before we go further it's better to spend some time on the below articles to understand how the random forest algorithm works.
How does the algorithm work? It works in four steps:

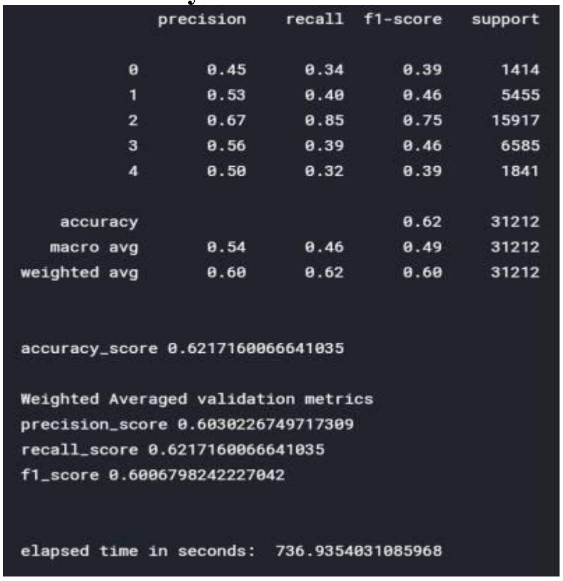
1. Select random samples from a given dataset.
2. Construct a decision tree for each sample and get a prediction result from each decision tree.
3. Perform a vote for each predicted result.
4. Select the prediction result with the most votes as the final prediction.



Random Forests vs Decision Trees

- Random forests is a set of multiple decision trees.
- Deep decision trees may suffer from overfitting, but random forests prevents overfitting by creating trees on random subsets.
- Decision trees are computationally faster.
- Random forests is difficult to interpret, while a decision tree is easily interpretable and can be converted to rules.

Model Accuracy:



```
              precision    recall  f1-score   support

           0       0.45      0.34      0.39      1414
           1       0.53      0.40      0.46      5455
           2       0.67      0.85      0.75     15917
           3       0.56      0.39      0.46      6585
           4       0.50      0.32      0.39      1841

    accuracy                           0.62     31212
   macro avg       0.54      0.46      0.49     31212
weighted avg       0.60      0.62      0.60     31212


accuracy_score 0.6217160066641035

Weighted Averaged validation metrics
precision_score 0.6030226749717309
recall_score 0.6217160066641035
f1_score 0.6006798242227042


elapsed time in seconds:  736.9354031085968
```

Result:

Figure 3 shows that: Accuracy Score, Weighted Average, Precision score, Recall Score & f1 Score, elapsed Time in sec of Dataset.

In which Random Forest Classifier is Accurate with high elapsed time.

Linear SVC:

A linear support vector machine is a type of binary classifier which tries to separate a data set into two categories with a maximum-margin hyper-plane. For each $x \in F$ a general feature vector set, we have an associated binary classification $y \in \{-1, 1\}$. We define our training set,

$$T = \{(x_i, y_i) : x_i \in F, y_i \in \{-1, 1\}\}$$

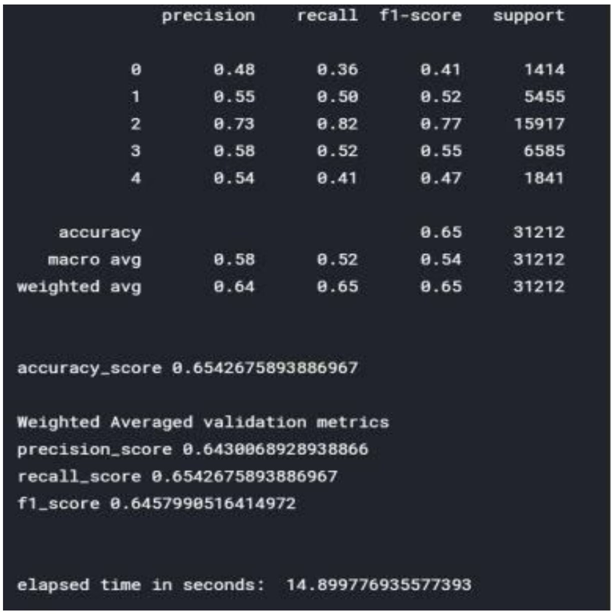
Then a hyperplane which satisfies the points of $x \in T$ is defined as $w \cdot x - b = 0$

We want all data points to fall outside of the margin of the hyperplane, so we impose the condition

$y_i \cdot (w\ Xi - b) \geq 1 \forall (x_i, y_i) \in T$ So the problem becomes under the above constraint.

arg min 1 $\|w\|2(w, b)2$ In this approach, we create five x vs. all linear SVM classifiers where x 0, . . . , 4 This essentially gives a ”reject-accept” type classifier where if we predict a sentence to have output value 1 then it has sentiment x and otherwise it does not. Then we simply predict the sentiment of each sentence with all five classifiers to produce a final sentiment.

Model Accuracy:



```
              precision    recall  f1-score   support

           0       0.48      0.36      0.41      1414
           1       0.55      0.50      0.52      5455
           2       0.73      0.82      0.77     15917
           3       0.58      0.52      0.55      6585
           4       0.54      0.41      0.47      1841

    accuracy                           0.65     31212
   macro avg       0.58      0.52      0.54     31212
weighted avg       0.64      0.65      0.65     31212


accuracy_score 0.6542675893886967

Weighted Averaged validation metrics
precision_score 0.6430068928938866
recall_score 0.6542675893886967
f1_score 0.6457990516414972


elapsed time in seconds:  14.899776935577393
```

Result: Figure 4 shows that: Accuracy Score, Weighted Average, Precision score, Recall Score & f1 Score, elapsed Time in sec of Dataset.
In which Linear SVC Classifier is highly Accurate with less elapsed time.

Naive Bayes Classifier:

Naive Bayes model is easy & simple to build and is highly useful for classifying text in very large data sets. Along with its simplicity, it is also known to perform even better than highly advanced & sophisticated classification algorithms. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c).

$$P(c|x) = \frac{P(x|c)\ P(c)}{P(x)}$$

There are 3 types of models are available for Naïve Bayes classifier in scikit-learn library:

Gaussian: It is used as a classifier by assuming that the features will follow a normal distribution.

Multinomial: It is used in case of disjoint counts. Now consider Bernoulli trials which is a step ahead and count how frequently letter occurs in the statement instead of letter occurring in the statement i.e., you can assume it as taking the number of times outcome number xp is observed over the q trials.

Table 1: Sample Reviews

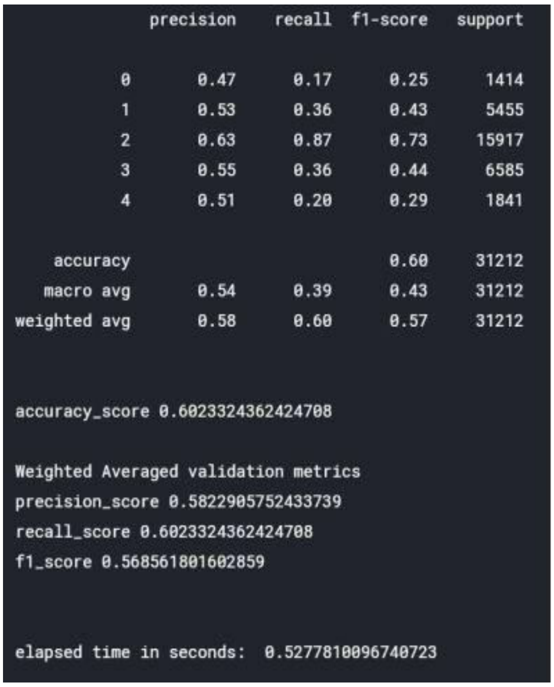
Sentiment	*Movie Reviews*
Positive	*The Puppet King is an amazing movie, don't take me wrong.* *The people who are very close to me know "To what extent I love the movie Zorokin.*
Negative	*It's completely waste of time and money both to watch Zorokin.* *Had an interesting discussion with one of my colleagues at work about how the movie "James and his Destiny" sucks.*

BernoulliNB

Bernoulli naive Bayes is similar to the multinomial naive bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering and recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier.

Model Accuracy



```
              precision    recall  f1-score   support

           0       0.47      0.17      0.25      1414
           1       0.53      0.36      0.43      5455
           2       0.63      0.87      0.73     15917
           3       0.55      0.36      0.44      6585
           4       0.51      0.20      0.29      1841

    accuracy                           0.60     31212
   macro avg       0.54      0.39      0.43     31212
weighted avg       0.58      0.60      0.57     31212


accuracy_score 0.6023324362424708

Weighted Averaged validation metrics
precision_score 0.5822905752433739
recall_score 0.6023324362424708
f1_score 0.568561801602859


elapsed time in seconds:  0.5277810096740723
```

Result:

Figure 5 shows that: Accuracy Score, Weighted Average, Precision score, Recall Score & f1 Score, elapsed Time in sec of Dataset.
In which Bernoulli NB Classifier is Accurate with very less elapsed time.

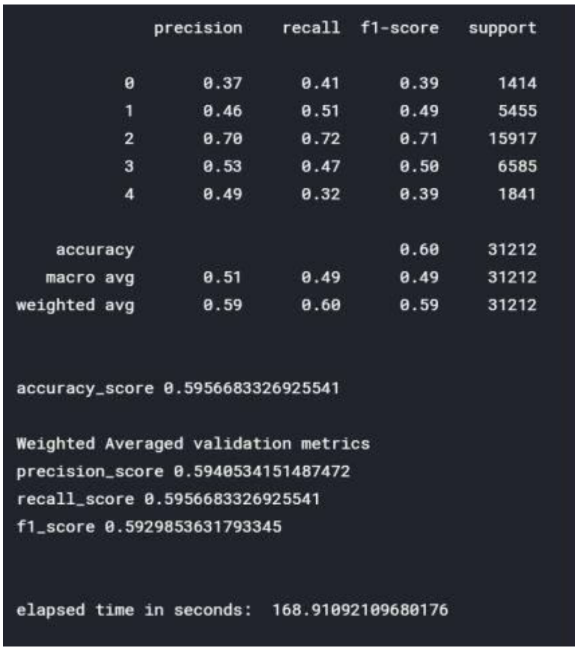
Lazy Classifiers :

K Neighbors Classifiers

K-nearest neighbors (or k-NN for short) is a simple machine learning algorithm that categorizes an input by using its k nearest neighbors. K-NN is non-parametric, which means that it does not make any assumptions about the probability distribution of the input. This is useful for applications with input properties that are unknown and therefore makes k-NN more robust than algorithms that are parametric. The contrast is that parametric machine learning algorithms tend to produce fewer errors than non-parametric ones, since taking input probabilities into account can influence decision making.

Furthermore, k-NN is a type of lazy learning, which is a learning method that generalizes data in the testing phase, rather than during the training phase. This is contrasted with eager learning, which generalizes data in the training phase rather than the testing phase. A benefit of lazy learning is that it can quickly adapt to changes, since it is not expecting a certain generalized dataset. However, a major downside is that a huge amount of computation occurs during testing (actual use) rather than pre-computation during training.

Model Accuracy :



```
              precision    recall  f1-score   support

           0       0.37      0.41      0.39      1414
           1       0.46      0.51      0.49      5455
           2       0.70      0.72      0.71     15917
           3       0.53      0.47      0.50      6585
           4       0.49      0.32      0.39      1841

    accuracy                           0.60     31212
   macro avg       0.51      0.49      0.49     31212
weighted avg       0.59      0.60      0.59     31212


accuracy_score 0.5956683326925541

Weighted Averaged validation metrics
precision_score 0.5940534151487472
recall_score 0.5956683326925541
f1_score 0.5929853631793345


elapsed time in seconds:  168.91092109680176
```

Result:

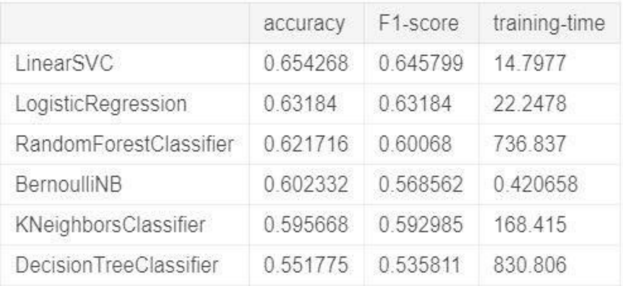
Figure 6 shows that: Accuracy Score, Weighted Average, Precision score, Recall Score & f1 Score, elapsed Time in sec of Dataset.In which Decision Tree Classifier is Accurate with high elapsed time.

Conclusion:

Using more than one model of ML for predicted sentiment of movie review on the basis of accuracy and reliability of each model.

This is what makes choosing the perfect model for analyses review. There are many popular ML methods for analysis, six of which are Regression Models, CART Models, Bagging Trees, SVM Models, Naive Bayes Models and Lazy Classifiers. Different models results different record of accuracy and elapsed time hence, the selection of one can be done based on various parameters.

After evaluating all the models we come to conclusion that **Linear SVC** is the most reliable model which is giving us the best results with an accuracy of 65.4268 and elapsed time 14.7977 sec and Decision Tree Classifier is the less reliable because of an accuracy of 0.551775 and elapsed time 830.806 sec.



	accuracy	F1-score	training-time
LinearSVC	0.654268	0.645799	14.7977
LogisticRegression	0.63184	0.63184	22.2478
RandomForestClassifier	0.621716	0.60068	736.837
BernoulliNB	0.602332	0.568562	0.420658
KNeighborsClassifier	0.595668	0.592985	168.415
DecisionTreeClassifier	0.551775	0.535811	830.806

References:

1. https://www.datacamp.com/community/tutorials /random-forests-classifier-python
2. https://www.analyticsvidhya.com/blog/2016/04/c omplete-tutorial-tree-based-modeling-scratch-in-python/
3. https://towardsdatascience.com/machine-learning-clas sifiers-a5cc4e1b0623
4. https://monkeylearn.com/blog/sentiment-analysis-wit h-python/
5. https://hub.packtpub.com/how-to-perform-sentiment- analysis-using-python-tutorial/