**Abstract**

Social media have received more attention nowadays. Public and private opinions about a wide variety of subjects are expressed and spread continually via numerous social media. Twitter is one of the social media that is gaining popularity. Twitter offers organizations a fast and effective way to analyze customers’ perspectives towards the critical to success in the marketplace. Developing a program for sentiment analysis is an approach to be used to computationally measure customers’ perceptions. The past studies of sentiment classification are not very conclusive about which features and supervised classification algorithms are good for designing accurate and efficient sentiment classification system. We propose to combine many feature extraction techniques to design more accurate sentiment classification system. In this project, we have performed supervised learning on an dataset which contians the opinion of users about the USA Elections and tweets regarding both candidates. The data was first preprocessed to remove unwanted information and classified using four different machine learning algorithms *(Decision Tree, K Nearest Neighbour, Logistic Regression, Random Forest*). This paper presents empirical comparison of four supervised classification algorithms. The tweets and data regarding them are stored and visualized using various graphs and plots. Prediction for the election result of each and every state in USA is done using tweets location.

**Keywords:** Sentiment analysis, Twitter, Python, Data Pre Processing, Decision Tree, K Nearest Neighbour, Logistic Regression, Random Forest, Comparision, Visualisation

**Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **List of Figures**……………………………………………….……………………… | | | IV |
| **List of Tables**………………………………………………….……………………. | | | V |
| **Nomenclature**………………………………………………………………………. | | | VI |
| **1** | **Introduction**……………………………………………………………………... | | 1 |
|  | 1.1 | Background……...…………………………………………………….…… | 1 |
|  | 1.2 | Motivation………………………………………………………….………. | 2 |
|  | 1.3 | Scope of the project ………………………………………………………. | 2 |
|  | 1.4 | Brief description of project undertaken……………………………………. | 3 |
|  | 1.5 | Organization of the report………………………………………………….. | 3 |
| **2** | **Literature Survey**………………………………………………………………. | | 4 |
| **3** | **Project design** …………………………………………………………………... | | 7 |
|  | 3.1 | Introduction………………………………………………………………... | 7 |
|  | 3.2 | Problem statement…………..……………………………………………... | 7 |
|  | 3.3 | Block diagram / system diagram…………………………………………... | 8 |
|  |  | 3.3.1 Process………………………………………………………….......... | 8 |
|  |  | 3.3.2 Methodology………….…………………………………………….. | 9 |
|  | 3.4 | Objectives………………….………………………………………………. | 9 |
|  | 3.5 | Data Collection……………………………………………………………... | 10 |
|  |  | * + 1. Train Data……………………………………………………… | 10 |
|  |  | * + 1. Test Data……………………………………………………… | 10 |
|  | 3.6 | Data Preprocessing ………………………………………………………… | 10 |
|  | 3.7 | Supervised learning…………..………………………..…………………… | 11 |
|  |  | 3.7.1 Decision Tree……..……………..…………………………………… | 11 |
|  |  | 3.7.2 K Nearest Neighbour…………………………………………………. | 12 |
|  |  | 3.7.3 Random Forest………………………………………………………... | 13 |
|  |  | 3.7.4 Logistic Regression…………………………………………………… | 13 |
|  | 3.8 | Result Comparators………………………………………………………….. | 14 |
| **4** | **Implementation and experimentation**………………………………………….. | | 15 |
|  | 4.1 | Hardware and software performance evaluation……….…………………… | 15 |
|  | 4.2 | Decision Tree Implementation Steps……………………………………….. | 15 |
|  | 4.3 | KNN classifier Implementation Steps …,…………………………………... | 18 |
|  | 4.4 | Random Forest Implementation Steps…………….………………………… | 22 |
|  | 4.5 | Logistic Regression Implementation Steps………………………………….. | 24 |
|  | 4.6 | Selection of classifier………………………………………………………… | 26 |
|  | 4.7 | Prediction of Election Result……………………………………………….... | 26 |
|  | 4.8 | Results and Discussion……………………………………………………….. | 32 |
| **5** | **Conclusions and scope for further work**…….………….………..……………… | | 36 |
|  | 5.1 | Conclusions………………………………………………………………..… | 36 |
|  | 5.2 | Scope for further work………………………………………………….…… | 36 |
|  |  |  |  |
| **Bibliography**…………………………………………………………………………. | | | 37 |
| **Appendix A**………………………………………………………………………….. | | | 39 |
| **Appendix B**………………………………………………………………………..…. | | | 41 |
| **Appendix C**……………………………………………………………………………. | | | 43 |
| **Acknowledgements**…………………………………………………………………… | | | 45 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| 3.1 | Block Diagram of Stages for Twitter Sentiment Analysis……………………… | 9 |
| 3.2 | Block Diagram for Methodology of Sentiment Analysis…….………………… | 10 |
| 3.3 | Block Diagram for the Training and Prediction Model.………………………… | 10 |
| 4.1 | Parameters Used in the Standard Decision Tree Classifier……………………… | 16 |
| 4.2 | Model, Train and Test Accuracy………………………………………………… | 17 |
| 4.3 | Accuracy for criterion ‘gini’ and ‘entropy’……………………………………… | 17 |
| 4.4 | Computation of max\_depth for both criterion…………………………………… | 17 |
| 4.5 | max\_depth value v/s Accuracy…………………………………………………. | 18 |
| 4.6 | Parameters of the DT Classifier Model………………………………………… | 18 |
| 4.7 | Best Parameters Obtained After Tuning of DT Model………………………… | 19 |
| 4.8 | Train and Test Accuracy for the Tuned DT Classifier Model…………………… | 19 |
| 4.9 | Pie Chart for the Confusion Matrix of KNN Classifier………………………….. | 20 |
| 4.10 | Computation of Accuracies for Different K value……………………………… | 21 |
| 4.11 | K value v/s Accuracy for k values 1-20………………………………………… | 21 |
| 4.12 | K value v/s Accuracy for k values 20-50………………………………………. | 22 |
| 4.13 | Check for Overfitting…………………………………………………………… | 22 |
| 4.14 | Traing and Test Accuracy for Different Values of K…………………………… | 23 |
| 4.15 | Training and Test Accuracy for Tuned KNN Model……………………………. | 23 |
| 4.16 | Cross Validation Score for Random Forest Classifier…………………………… | 23 |
| 4.17 | Parameters of Random Forest Classifier Model………………………..………. | 24 |
| 4.18 | Parameters Used in the Standard Logistic Regression Model………..…………. | 24 |
| 4.20 | Parameters of the Logistic Regression…………….……………………………. | 24 |
| 4.21 | Tuning of parameters for Logistic Regression …………………………………. | 25 |
| 4.22 | Best Parameters for the Logistic Regression Model……………….……………. | 25 |
| 4.23 | Train and Test Score for Tune Logistic Regression Model……………..………. | 25 |
| 4.24 | Number of tweets for both candidate for each state(i)………………………….. | 27 |
| 4.25 | Number of tweets for both candidate for each state(ii)…………………………. | 28 |
| 4.26 | Tweets for Joe Biden and Donald Trump statewise(i)………………………….. | 29 |
| 4.27 | Tweets for Joe Biden and Donald Trump statewise(ii)…………………………. | 29 |
| 4.28 | Predicted Judgment for all states………………………………………………... | 30 |
| 4.29 | Distribution of Predicted Judgement Data………………………………………. | 31 |
| 4.30 | States under Somewhat Democratic and Insufficient Data …………………….. | 31 |
| 4.31 | States Under Somewhat Republican and Strongly Democratic ………………… | 32 |
| 4.32 | States Under Strongly Republican……………………………………………….. | 32 |
| 4.33 | USA 2020 Results……………………………………………………………….. | 33 |
| 4.34 | Actual Results Statewise……………….…….………………………………….. | 34 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| 4.1 | Hardware and Software Used……………………………………………….…... | 15 |
| 4.2 | Comparision of Result Parameters……………………………………………… | 26 |
| 4.3 | Comparison of Actual and Predicted Results for ‘Strongly Democratic’………. | 39 |
| 4.4 | Comparison of Actual and Predicted Results for ‘Somewhat Democratic’…….. | 39 |
| 4.5 | Comparison of Actual and Predicted Results for ‘Strongly Republic’……….. | 40 |
| 4.6 | Comparison of Actual and Predicted Results for ‘Somewhat Republic’……….. | 40 |
|  |  |  |
|  |  |  |

**Nomenclature**

|  |  |
| --- | --- |
| FP | False positive |
| FN | False negative |
| F1 | Evaluation parameter combining precision and recall |
| K | Number of nearest neighbours(training documents) with which distance computation is to be performed |
| NFt | Number of documents containing term t |
| TP | True positive |
| TN | True Negative |

**Chapter 1**

**Introduction**

*This chapter presents a brief idea about this project where the aim is to provide a quick understanding of the various concepts used in the twitter sentiment analysis. It also gives a succinct account about the motivation behind the project and its scope.*

N n

**1.1 Background**

Within the past few years, there has been a colossal development within the utilize of microblogging stages such as Twitter. Impelled by that development, companies, and media organizations are progressively looking for ways to mine Twitter for data approximately what individuals think and feel about almost their items and services. Millions of individuals antisocial organize locales to precise their feelings, conclusions and uncover their day-by-day lives. Be that as it may, individuals type in anything such as social exercises or any comment on items. Through the online communities give an intelligently gathering where customers illuminate and impact others. Such as social media to promote or talk specifically to clients for interfacing with the customer's viewpoint of items and administrations.

Sentiment analysis is the analysis of feelings in text, videos or images. The purpose is to pick up an diagram of the more extensive open supposition behind certain topics. Accurately, it may be a opinion of public. Social media destinations are used to connect with other individuals and to remain well aware with news and current occasions. Social media has become a stage to individuals to voice their suppositions. This kind of data shapes a premise for assessing , rate the performance of not as it were any motion picture but around other items and to know almost whether it'll be a victory or not. This sort of endless data on these locales can utilize for promoting and social thinks about. Hence, estimation investigation has wide applications and incorporate feeling mining.

**1.2 Motivation**

Being amazingly inquisitive about being related with Machine Learning, the final year project was an incredible opportunity to donate us the time to learn and affirm our interest in machine learning. The idea that we will make predictions,analysis and give the machines power to memorize by themselves is both capable and boundless in terms of possibilities. Machine Learning is used almost everywhere. That’s why we chose to conduct this project.

**1.3 Scope of the Project**

The scope of Twitter Sentiment Analysis is is extensive. The capacity to extricate sentiment knowledge from tweets may be a miracle that's being broadly embraced by organizations over the world. Emotion on social media have its wide effect on various industries. Different scope of Twitter Assumption Examination is as takes after:

* Digital Marketing.
* Improving Customer Experience (CX).
* Monitor various sentiments in real-time and over-time.
* Making marketing/sales strategies based on the response from consumers.
* Identify various problems, threats before they cause major problems.
* Eradicate hate speech, trolling, online bullying, etc.
* Identify anti-nationalist, extreme-nationalist comments which might hurt the sentiments of the locals.
* Identifying fake news and automatically blocking/reporting them.

**1.4 Brief Description of Project**

In this project we have classified tweets from twitter into ‘positive’ or ‘negative’ sentiment by building different machine learning model based on probabilities to understand the emotions of people towards ‘Donald Trump’ and ‘Joe Biden’ for US Elections 2020. Various preprocessing techniques are used to clean the data to get optimum results and analyses of the results is done based on hashtag, location, sentiment of tweets etc.

**1.5 Organization of the Report**

Chapter 1: This chapter presents a brief outline explaining what twitter sentiment analysis is and how it can extract insights from the tweets across the world.

Chapter 2: This Chapter explains different methods used for sentiment analysis by different researchers and the resultant outputs and conclusion drawn from those papers.

Chapter 3: This Chapter explains problem statement, methodology used to implement this project along with its objectives.

Chapter 4: This Chapter presents comparative study of different machine learning algorithms, analysis and final results of the project.

Chapter 5: This Chapter presents the conclusion and the future scope of the project.

*Summary of this chapter……*

*We explained the concept of twitter analysis and how machine learning has boundless application in all fields and with the help of sentiment analysis, we can get insights of the tweets across the world.*

*In the next chapter, we shall understand the different machine learning algorithms used by different researchers and their respective outputs and conclusions.*

N n

**Chapter 2**

**Literature Survey**

*This chapter presents some of the scholarly and research works in the field of Data analysis and Data Mining on analyzing sentiments from data and preparing prediction model accordindly.*

[1] Explains different phases in which twitter data is classified based on sentiments. Real life twitter data was extracted in the form of hashtags. Preprocessing of data is done using Tokenization and Stemming. Later, these preprocessed tweets are classified into positive, negative and neutral sentiments. Visualization was done using histogram plots and generating pie charts of classifications made based on sentiment score.

[2] Built a model that classified tweets into different sentiments. Experimentation was done using different models. Tree based model used tweets as tree. The feature based model uses many different highlights and the unigram model uses thousands of highlights. Comparing all the models, Tree based model worked well compared to other two models.

[3] Tweepy was used to collect data from twitter. They classified the tweets into different sentiments. Various preprocessing techniques were used and redundant data was filtered using Chi-square technique. Finally, Naïve Bayes Classifier was used to classify the sentiments into positive and negative.

[4] Naïve Bayes classifiers were used to detect polarity of tweets. Variations of Credulous Bayes classifiers were assembled to be specific Pattern (prepared to arrange tweets as sure, negative, and nonpartisan), and Parallel (utilizes an extremity vocabulary and characterizes as certain and negative. Impartial tweets dismissed). The highlights considered by classifiers were Lemmas (things, action words, descriptors, and intensifiers), Extremity Vocabularies, and Multiword from various sources, and Valence Shifters.

[5] Naive Bayes, Maximum Entropy, and Support Vector Machines were used as base classifiers and two different feature sets to classify the sentiments into different polarity. For better conclusion and better accuracy different methods like the fixed blend, weighted mix, and Meta-classifier are applied.

[6] Distant Supervision was used for twitter sentiment analysis. Train data consist of noisy data such as emoticons. Different ML algorithms like Naïve Bayes, Maximum Entropy, and Support Vector Machine was used for classification. Unigram, Bigram and POS were used as features. Support Vector Machine and unigram combination outperformed the other combinations of models and features.

[7] Hashtags in tweets were used along with punctuation, single words, n-grams, and patterns as different features for sentiment classification. K-Nearest Neighbor algorithm was used to assign sentiments to the training and the testing dataset.

[8] Firehouse API was used to extract real time twitter data. Machine Learning based algorithms like Multinomial Naive Bayes, Stochastic Gradient Descent, and the Hoeffding Tree were used to analyze the tweets. GD-based models were comparatively better than others when right learning rate is used.

[9] Machine Learning algorithms like SVM, Naïve Bayes, Logistic Regression, and Random Forest classifiers are used to classify the tweets into different sentiments. Six different preprocessing methodologies are applied to the dataset, the precision and F1 score of these classifiers are improved by preprocessing strategies. The Naïve Bayes and Random Forest classifier are touchier than other models to preprocessing techniques.

[10] ML algorithms like Naivee Bayes, SVM and Maximum Enropy are used to classify the tweets into different sentiments and accuracy can be significantly improved when wordnet is used along with these ML algorithms.

[11] Used a bag-of-words method for sentiment analysis in which the relationships between words were not at all considered and a document is represented as just a collection of words. To determine the sentiment for the whole document, the sentiments of every word were determined and those values are united with some aggregation functions.

[12] Created Twitter corpus using Twitter API to gather data that includes emojis. Naive Bayes Classifier along with features like Ngram and POs was used to classify the tweets into various sentiments. The Model did not achieve better accuracy as the train data contained the tweets that only includes emojis.

[13] Two Machine Learning algorithms like Naive Bayes bigram model, and a Maximum Entropy model was used to classify the tweets into different sentiments. They tracked down that the Naive Bayes classifiers worked far superior to the Maximum Entropy model.

[14] ML algorithms like Naïve Bayes Classifier, Maximum Entropy and SVM are used to classify the tweets into various sentiments. Feature Selection technique played the major role as it increased the overall accuracy of the model. In most of the cases Naïve Bayes gave better accuracy compared to others but in some cases even Maximum Entropy model was successful to classify the tweets.

*Summary of this chapter……*

*We understood different pre processing techniques like tokenization, stemming, removal of URLs etc can be used to optimise our data set. Different machine learning algorithms like Decision Tree, KNN, SVM, Logistic Regression, Maximum Entropy and Naïve Bayes can be used to classify the tweets into different sentiments that is positive, negative and neutral. In some cases Naïve Bayes gave better accuracy while in others Logistic Regression and Maximum Entropy did wonders. The Bayes and RandomForest Classifier are more sensitive than Logistic Regression and SVM after applying pre processing techniques.*

*In the next chapter, we shall understand the different machine learning algorithms used to implement the project along with its advantages and disadvantages.*

N n

**Chapter 3**

**Project Design**

*This chapter presents the problem statement, the objectives and the methodology followed to implement the project.*

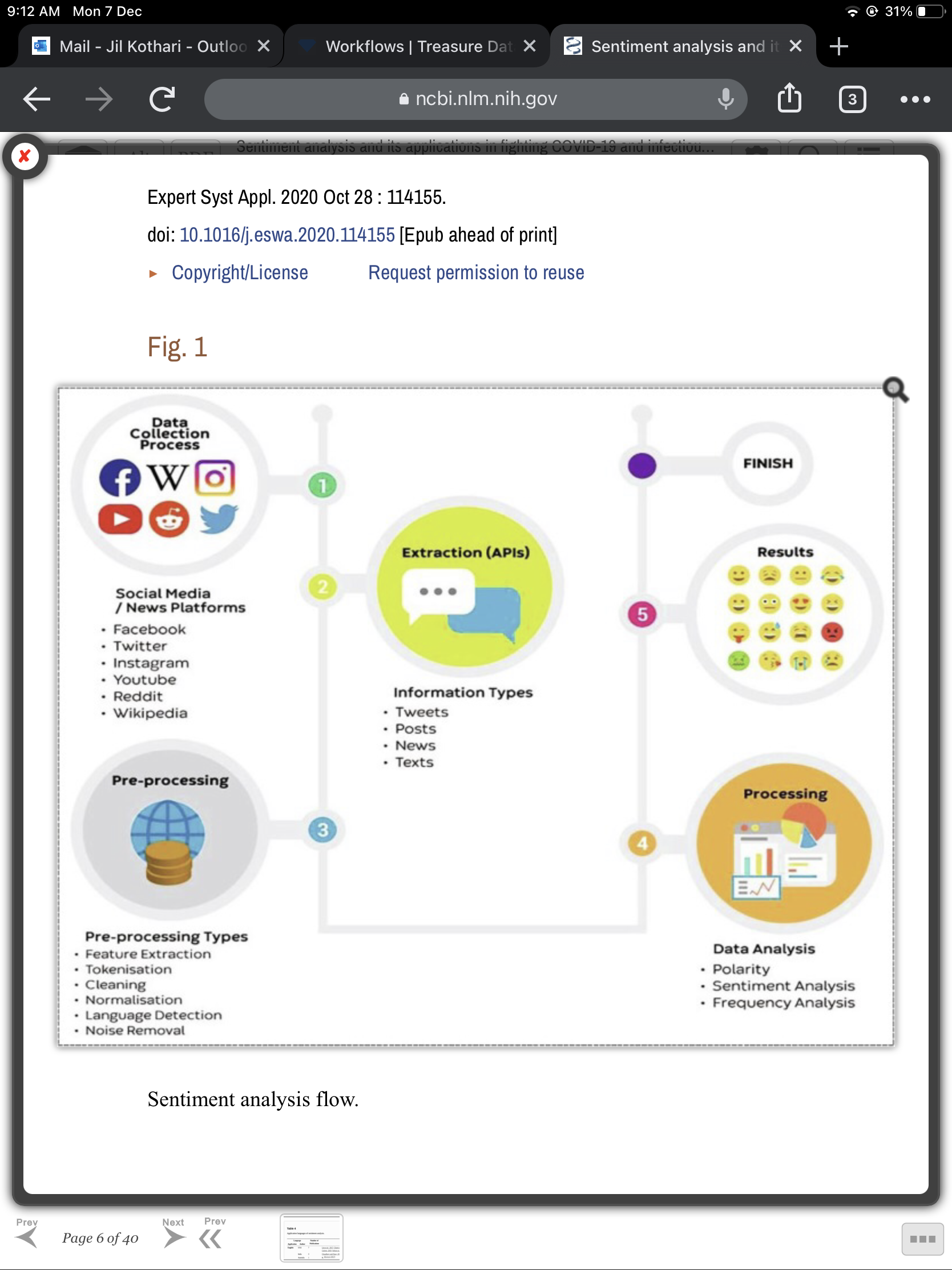
**3.1 Introduction**

Amid decisions individuals go through the audits to create it straightforward to choose or get an thought on whom to select or elect the proper candidate within the races. Assumption investigation, permits client to know the circumstance of what the audit are made with respect to particular item or progressing things instead of understanding profound contemplations of the analyst. In this extend we have collected twitter information before the USA 2020 Races which has been utilized as test information. To begin with, we have preprocessed the data and after that performed preparing on four diverse machine learning models and after tuning hyperparameters we chosen the finest show and made expectations on test information. This extend is isolated into four parts: information collection, preprocessing, estimation investigation, and information representation.

**3.2 Problem Statement**

Our aim of this project is to build a program to perform Twitter Sentiment Analysis using machine learning algorithms and. which filters the tweets based on the subject and location, classify them and provide us a general analysis of what the people are feeling about the USA 2020 Elections. We have also predicted winner for each and every state in USA based on the data collected from various location.

**3.3 Block Diagram**

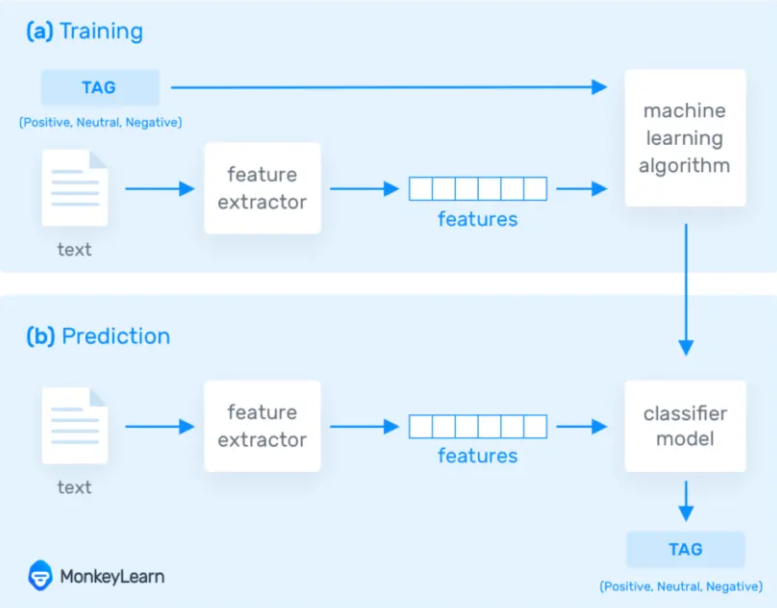


**Fig.3.1** Block Diagram of Stages for Twitter Sentiment Analysis

**Methodology**



**Fig.3.2** Block Diagram for the Methodlogy of Sentiment Analysis



**Fig.3.3** Block Diagram for the Training and Prediction Method

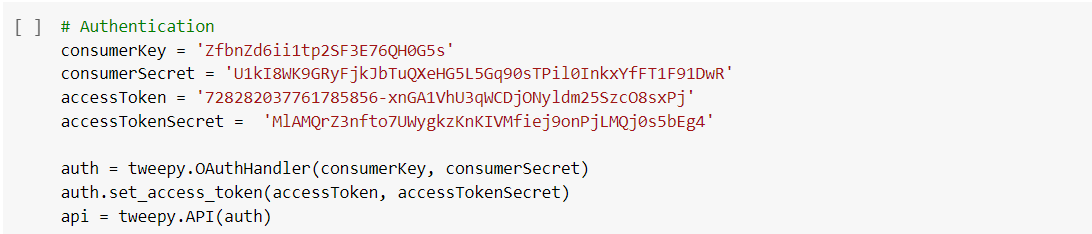
**3.4 Objectives**

The main objectives of the project are:

* Pre Processing the data.
* Implementing the four machine learning models (Decision Tree, X, Logistic Regression, and Random Forest,K Nearest Neighbour).
* Selecting the best model by comparing accuracies, precision, f1 score and recall.
* Preparing test data by collecting real time tweets with the help of Twitter API.
* Perform sentiment analysis on test data and analyse the result.

**3.5 Data Collection**

1. **Test Data**
2. The user’s tweets from Twitter are collected the client inputs text in the form of hashtags. It is the first step in sentiment analysis. Utilizing Twitter API, we gathered the Twitter data. The OAuth is gave the keys. Consumer Key, Consumer Secret, Access Token, and Access Token Secret for twitter application and perform Handshake protocol. After which, the certificate is downloaded and a PIN is generated for the application to access tweets.



1. **Train Data**

The train data comprises of user tweets with its respective sentiment. It consists of around 100,000 tweets.

**3.6 Data Preprocessing**

We used Python programming language to implement pre-processing steps Twitter Dataset.

1. **Tokenization**

Tokenization basically refers to splitting up a larger body of text into smaller lines, words or even creating words for a Non-English language. The various tokenization functions in built into nltk module are used for data processing.

1. **Stemming and Lemmatization**

Stemming and Lemmatization generate root form of the inflected words. The difference is that stem might not be an actual word whereas, lemma is an actual word. Stemming process is faster as it follows an algorithm with steps to perform on the words whereas, Lemmatization uses WordNet Corpus which makes it slower than stemming.

1. **CountVectorizer**

CountVectorizer is used to convert a collection of text documents to a vector of counts. It also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for the text.

1. **TFIDF Transformer**

TF-IDF stands for Term Frequency- Inverse Document Frequency and is a statistic that aims to better define how important a word is for the document. This is performed by looking at how many times a word appears into a document.

**3.7 Supervised Learning**

Supervised learning uses a training set to show models to yield the required output. This training dataset includes inputs and proper outputs, which permit the model to be told over time. The algorithmic program measures its accuracy through the loss operate, adjusting till the error has been sufficiently decreased. Varied algorithms and computation techniques area unit employed in supervised machine learning processes. We've used following algorithms for classification:

1. **Decision Tree**

Decision tree is a powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. Decision trees perform classification without requiring much computation. Decision trees are able to handle both continuous and categorical variables.

Advantages of the Decision Tree:

* It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
* It can be very useful for solving decision-related problems.
* There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree:

* The decision tree contains lots of layers, which makes it complex.
* It may have an over fitting issue, which can be resolved using the Random Forest algorithm.
* For more class labels, the computational complexity of the decision tree may increase.

1. **K Nearest Neighbour**

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm that can be used for both classifications as well as regression predictive problems. K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.

Advantages of KNN Algorithm:

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

1. **Random Forest**

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Advantages of Random Forest:

* It overcomes the problem of over fitting by averaging or combining the results of different decision trees.
* Random forests work well for a large range of data items than a single decision tree does.
* Random forest has less variance then single decision tree.

Disadvantages of Random Forest:

* It is less intuitive in case when we have a large collection of decision trees.
* The prediction process using random forests is very time-consuming in comparison with other algorithms.

1. **Logistic Regression**

Logistic Regression is a classification algorithm, used to find probability of event success and event failure. It supports categorizing data into discrete classes by studying the relationship from a given set of labelled data. It learns a linear relationship from the given dataset and then introduces a non-linearity in the form of the sigmoid function.

Advantages of Logistic Regression:

* It is easy to implement, interpret and very efficient to train.
* It is very fast at classifying unknown records.
* It can interpret model coefficients as an indicators of feature importance.

Disadvantages of Logistic Regression:

* If the number of observations is less than the number of features, Logistic Regression should not be used, otherwise, it may lead to over fitting.
* Nonlinear problems can’t be solved with it because it has a linear decision surface.
* It is tough to obtain complex relationships using Logistic Regression. More powerful and compact algorithms such as neural networks can easily outperform this algorithm.
  1. **Result Comparators**

1. **Accuracy**

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observations to total observations.

1. **Precision**

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

1. **Recall (Sensitivity)**

Recall is the ratio of correctly predicted positive observations to all the observations in the actual class.

1. **F1 Score**

F1 Score is the weighted average of the Precision value and the Recall value.

*Summary of this chapter……*

*We understood different pre-processing techniques like tokenization, stemming, removal of URLs etc and different supervised machine learning algorithm like Decision Tree. K Nearest Neighbor, Random Forest and Logistic Regression. KNN can be very effective when the training data is large, Random Forest is used over decision tree because it is very advanced and can reduce the problem of over fitting. Logistic Regression is easy to implement, interpret and easy to train but it is tough to obtain complex relationship. We also learnt some result comparators such as Accuracy, Precision, Recall and F1 Score that can be used to analyse and compare our results for the different machine learning algorithms.*

*In the next chapter, we shall understand the different implementation steps and the results obtained using different machine learning algorithms.*

N n

**Chapter 4**

**Implementation and Experimentation**

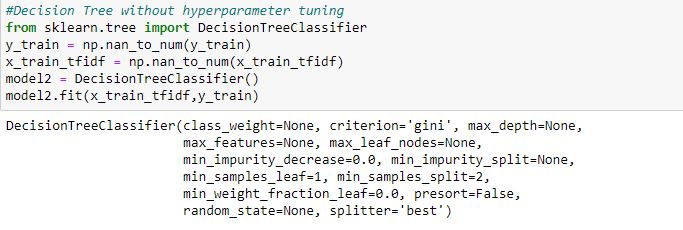
*This chapter presents the implementation steps and how the most accurate results were obtained. Four models were implemented the best one was selected for testing. After predicting the sentiment tweets were filtered based on location and hashtag from which we predicted result for each state.*

**4.1 Hardware and Software Used**

|  |  |
| --- | --- |
| HARDWARE | SOFTWARE |
| **Windows 10** | **Python 3.0 or above** |
| **Nvidia 940 MX Graphic Card** | **Anaconda Framework and Jupyter Notebook and Google Colab** |
| **RAM: 8GB**  **Hard disk: 1 TB** | **Twitter API** |

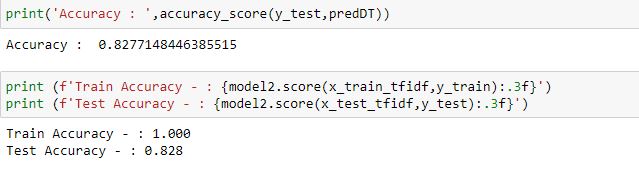
**Table 4.1** Hardware and Software Used

**4.2 Decision Tree Classifier Implementation Steps**

****

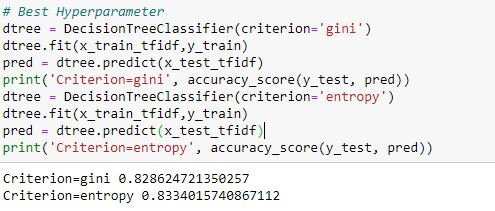
**Fig.4.1** Parameters Used in the Standard Decision Tree Classifier

For Decision Tree Classifier, we obtained an accuracy of 82.7%.



**Fig**.**4.2** Model, Train and Test Accuracy

Here, as we can see that Train accuracy is 1 which leads to the over fitting. To avoid over fitting, tuning of parameters is required that can reduce the train accuracy and increase the test accuracy.



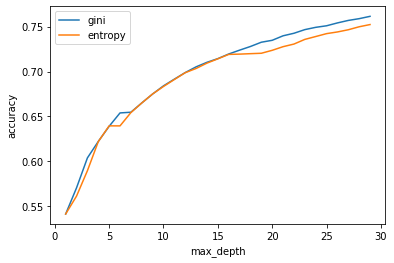
**Fig**.**4.3** Accuracy for criterion ‘gini’ and ‘entropy’

As criterion entropy gave little better accuracy than criterion gini, we shall use entropy as our parameter to increase accuracy.



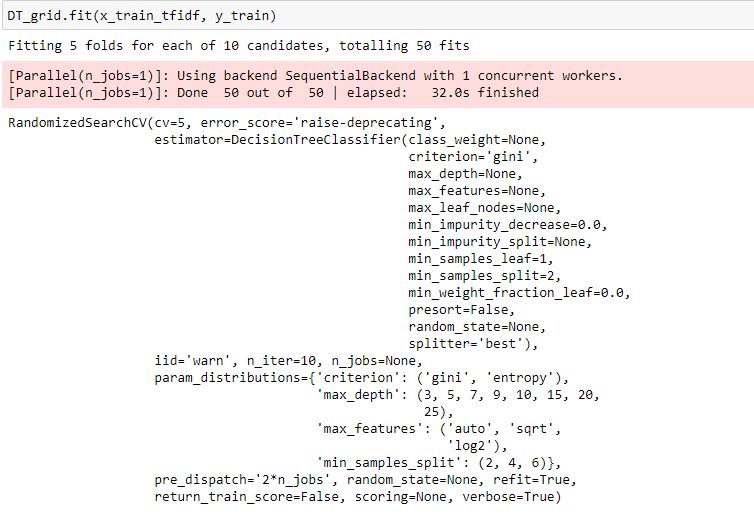
**Fig**.**4.4** Computation of max\_depth for both criterion

Let us find the best max\_depth value for criterion ‘gini’ as well as criterion ‘entropy’.

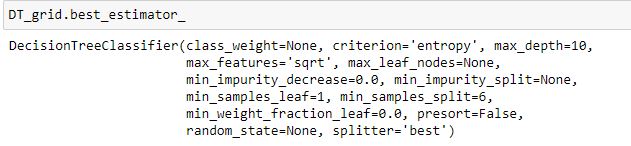


**Fig.4.5** max\_depth value v/s Accuracy

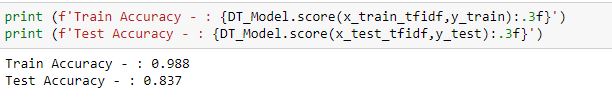
Using RandomizedSearchCV, we can alter the parameters to avoid overfitting of the data and also to increase the accuracy of the model.



**Fig.4.6** Parameters of the DT Classifier model



**Fig**.**4.7** Best Parameters obtained after Tuning of Decision Tree Classifier



**Fig.4.8** Test and Train Accuracy for the tuned DT Classifier Model

Accuracy achieved = 83.7% Recall = 0.823

Precision = 0.853 F1\_Score = 0.837

Confusion Matrix:

[[9316 1602]

[2000 9063]]

Train Accuracy has reduced from 1 to 0,98 which limits the problem of over fitting and there has been slight increment in the Test Accuracy. The overall accuracy of the tuned model is 83.7% which does not show great improvement compared to the accuracy of untuned model. Therefore, to get the desired results and to improve the accuracy of the model, Random Forest Classifier is used which is more advanced and better accuracy compared to Decision Tree Classifier.

**4.3 K Nearest Neighbour Classifier Implementation Steps**

We implemented the Kneighbor Classifier wherin we did tuning of various parameter to get highest accuracy. Without parameter tuning we got following results:

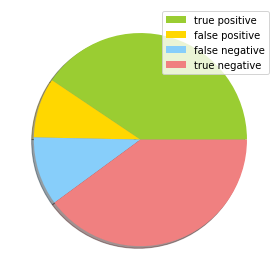
Accuracy achieved = 80.51% Recall = 0.795

Precision = 0.817 F1\_score = 0.806

Confusion Matrix:

[[8921 1997]

[2287 8776]]



**Fig**.**4.9** Pie Chart for the Confusion Matrix of KNN Classifier

The most important task in implementing the KNN algorithm was finding an optimal value of K. We did it by :

1. **By Cross-validation:**

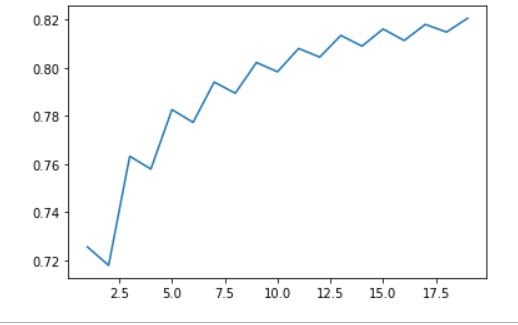
Divided the training data into 5 parts out of which 4 parts was training data and 1 part was validation data.

Performed cross validation and got the accuracy for different values of k ranging from (1-50).

Only odd values of k were taken as even number of neighbours can lead to miscalculations.

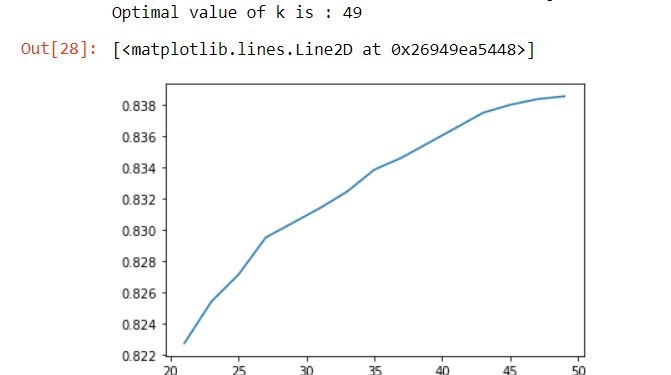
**Fig.4.10** Computation of accuracies for different K value

From k values 1-20 ( X axis= k\_values, y-axis= Cross validation scores)



**Fig.4.11** K value v/s Accuracy for k values 1-20

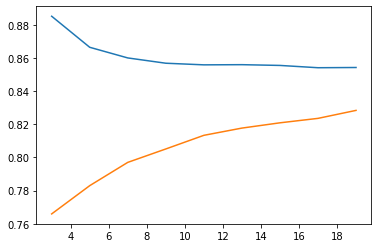
From k values 20 - 50( x axis =k\_values, y-axis= accuracy)



**Fig.4.12** K value v/s Accuracy for k values 20-50

As we see the accuracy is increasing as the value of k increases. Our dataset is too large so we cannot traverse through entire dataset.

1. **Check for Overfitting**
2. After testing various values of k on training set and testing set we see that, we get different accuries on both set for same value of k.



**Fig.4.14** Training and Test dataset accuracy for different values of K

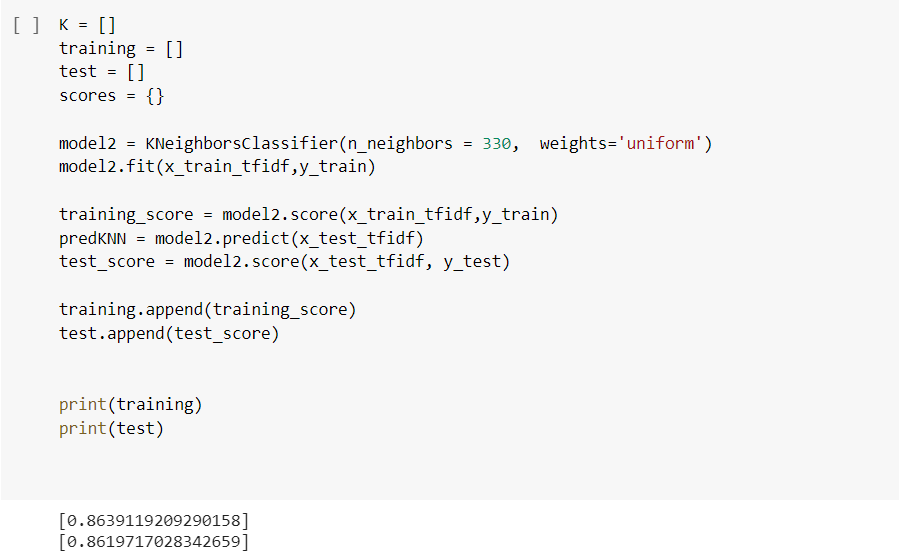
The blue line is training dataset accuracy and yellow line is test dataset accuracy.

1. **Square Root of N (N= number of sampes)**

As our training dataset consists of 90000 samples we cannot check accuracy for each an every value of k.

As k value increases the training accuracy becomes constant around 86% and test dataset accuracy increases constantly.

So by taking k as square root of number of samples we tried to find accuracy on training as well as test dataset and the test dataset attained accuracy of 86%.



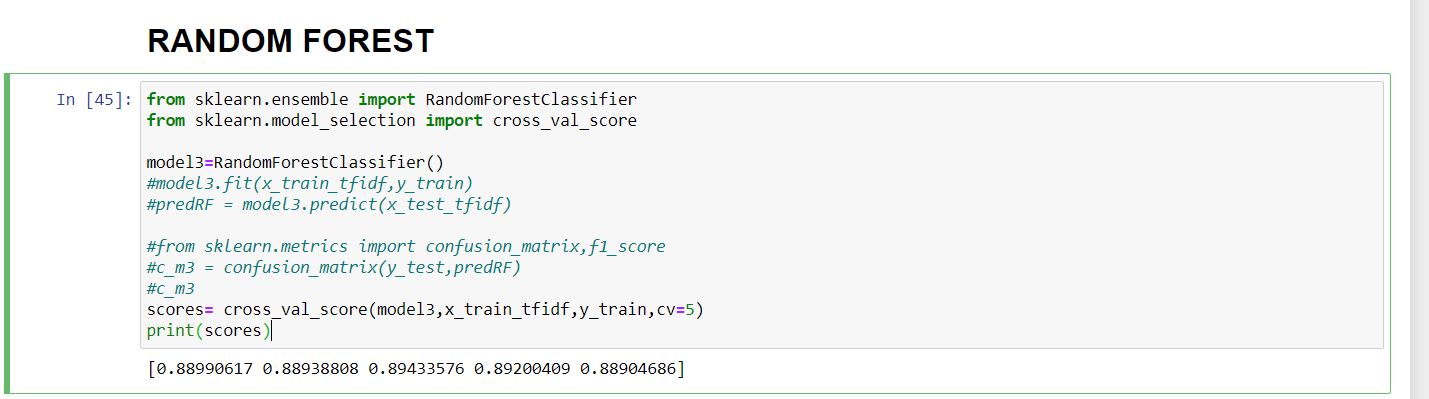
**Fig.4.15** Training and Test Accuracy for tuned KNN Model

Accuracy:[0.8620171966698512] recall = 0.8354462690376925

precision = 0.8993405385601758 F1\_score = 0.8662167526796348

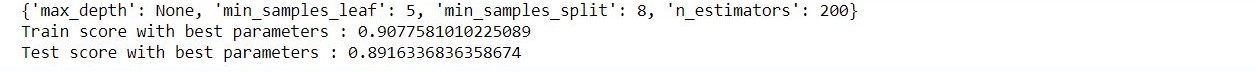
**4.4 Random Forest Classifier Implementation Steps**

Random Forest is a modified version of the Decision Tree. We have implemented RandomForestClassifier from sklearn library. Firstly we performed cross validation to check for overfitting and the results are as shown below.



**Fig.4.16** Cross Validation Score for Random Forest Classifier

Then we tuned the hyperparameters based on our dataset using GridSearchCV. With the help of GridsearchCV we gave a list of parameters and it gave us the best parameters based on the accuracy of model on our dataset.



**Fig.4.18** Train and Test Score for tuned Random Forest Classifier Model

Confusion Matrix:

[[9841 1077]

[1285 9778]]

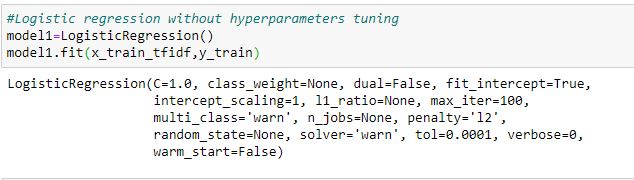
recall = 0.8845047636167536

precision = 0.9013555596263052

F1\_score = 0.892850662311740

**4.5 Logistic Regression Classifier Implementation Steps**

We implemented Logistic Regression Classifier without hyper parameters tuning as well as with hyper parameters tuning.



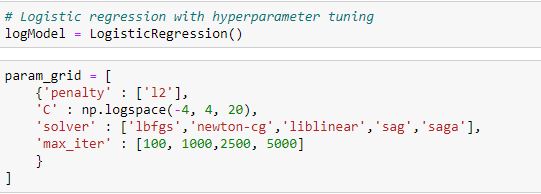
**Fig.4.19** Standard Parameters of Logistic Regression model

Without hyper parameter tuning, we achieved an accuracy of 90.8% and the confusion matrix for the same is

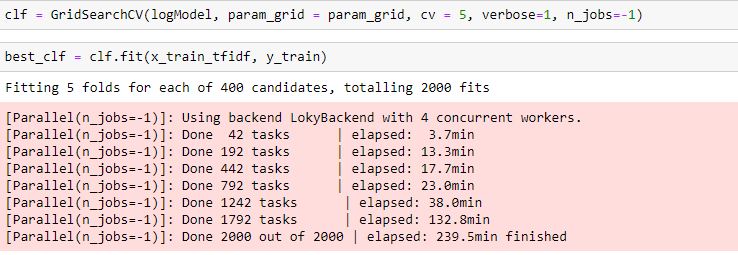
[[9923 995] Recall = 0.907 Precision = 0.908

[1021 10042]] F1\_score = 0.907

For hyperparameters tuning, we have created a dictionary with all the parameters of the Sklearn Logistic Regression Model and then using GridSearchCV, we extracted all the optimum parameters that can be used to increase the accuracy of the model.



**Fig.4.20** Parameters of Logistic Regression



**Fig.4.21** Tuning of parameters for Logistic Regression



**Fig.4.22** Best Parameters for Logistic Regression

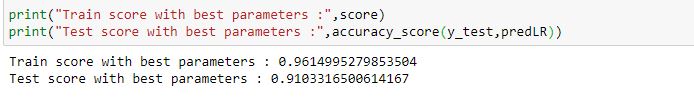
Accuracy achieved = 96.1% Recall = 0.907

Precision = 0.912 F1\_score = 0.909

Confusion Matrix:

[[9923 995]

[1021 10042]]



**Fig.4.23** Train and Test Score for tuned Logistic Regression Model

**4.6 Selection of Classifier**

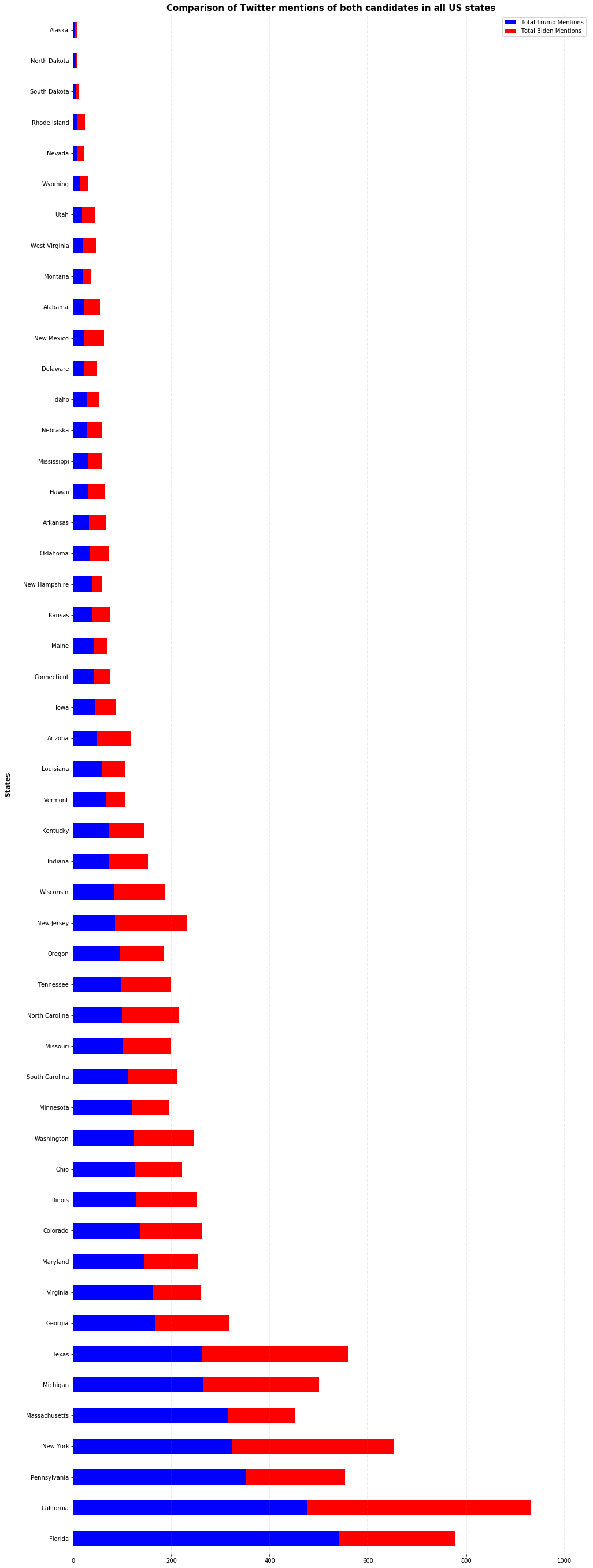
We can see that out of all four algorithm we are getting best accuracy in the Logistic Regression Model. Apart from accuracy the precision, f1score and recall are comparatively better in the Logistic Regression Algorithm. So we will use it on the test data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters | Decision Tree | K Nearest Neighbour | Random Forest | Logistic Regression |
| Accuracy | 83.7 | 86.2 | 89.16 | 96.1 |
| Precision | 85.3 | 89.93 | 90.1 | 91.2 |
| F1 | 83.7 | 86.62 | 89.28 | 90.9 |
| Recall | 82.3 | 83.54 | 88.45 | 90.7 |

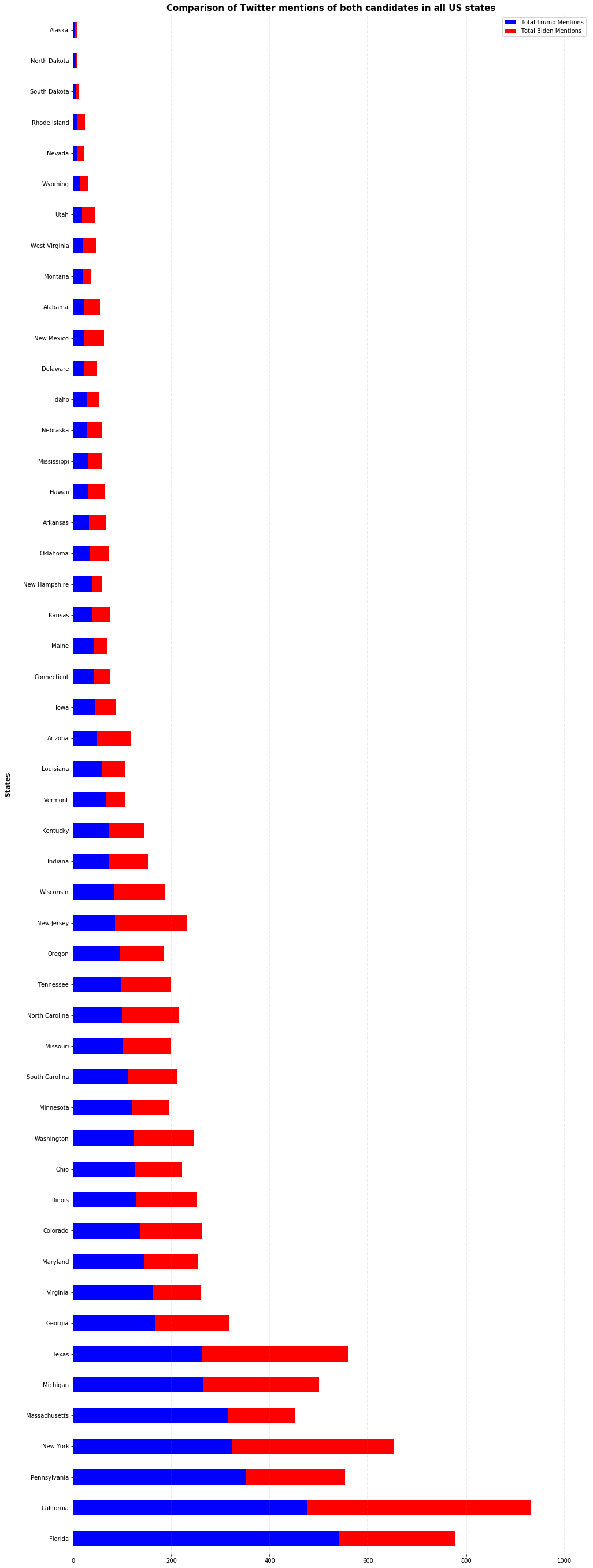
**Table 4.2** Comparision of Result Parameters

**4.7 Prediction of Election Results**

1. We follow the same steps for data preprocessing on our test data. After preprocessing we do the prediction using logistic regression. After prediction comes the analysis part wherein we predict the result for each state based on the location of tweets.
2. For this we separate out tweets of two candidates (Donald Trump and Joe Biden) into two data frames.
3. Then we determine number of tweets of each candidate for each an every state. The output for the same is shown below.

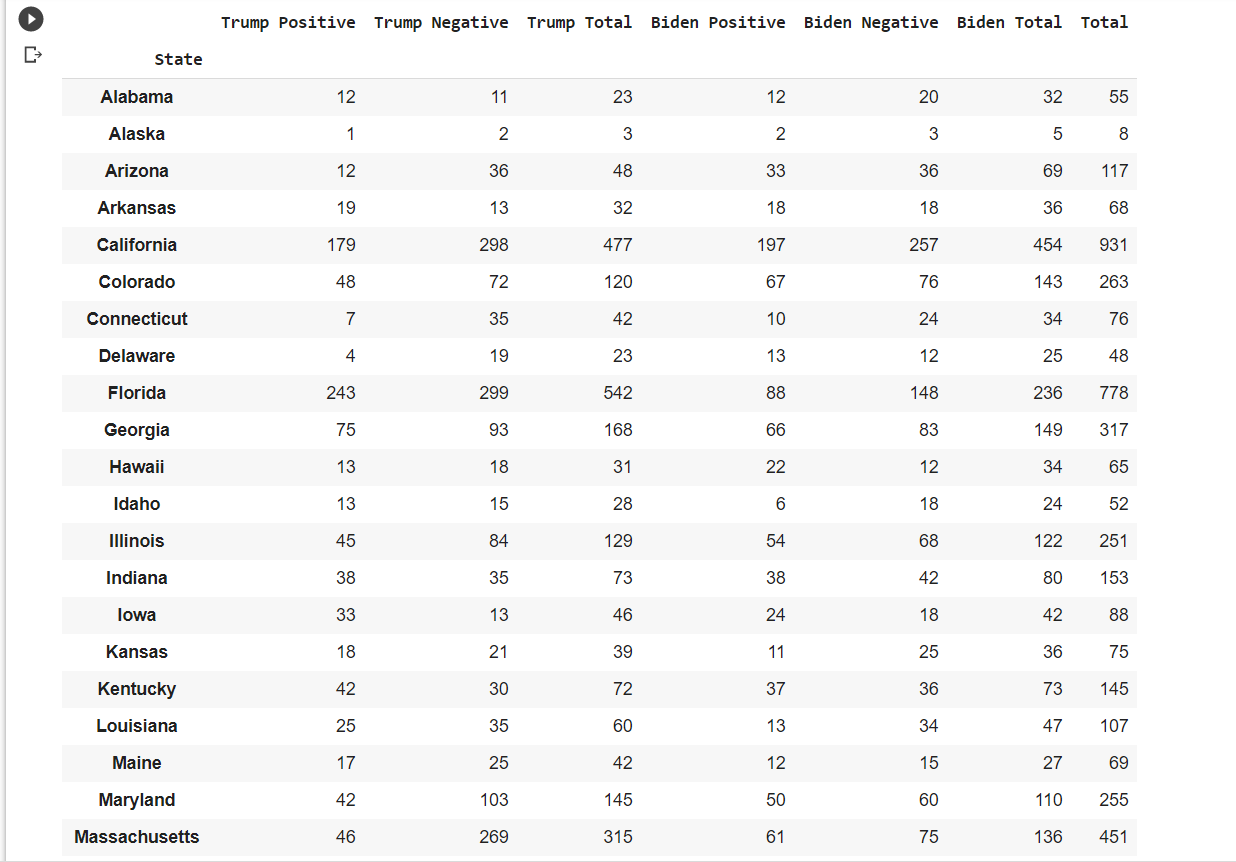


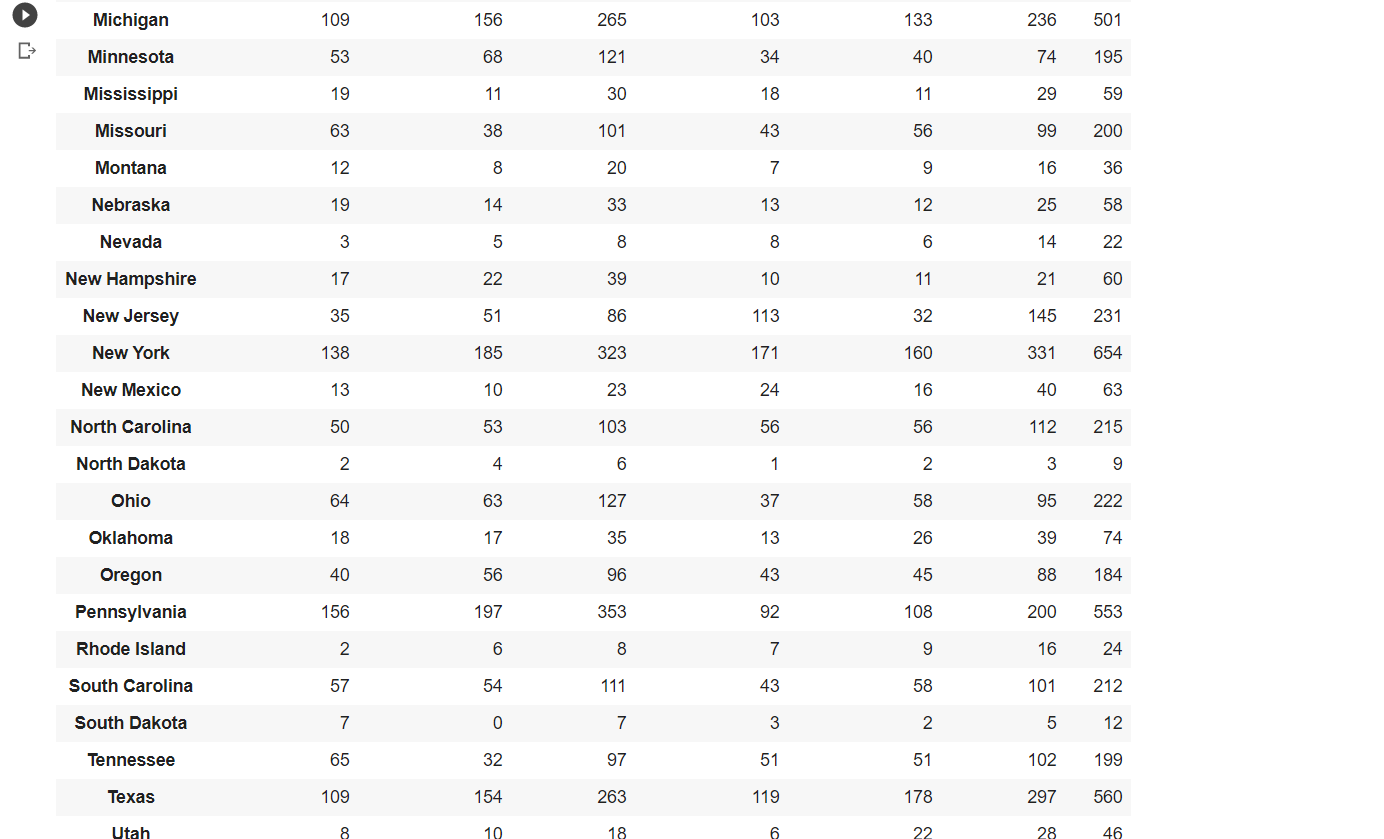
**Fig.4.24** Tweets for Joe Biden and Donald Trump statewise(i)

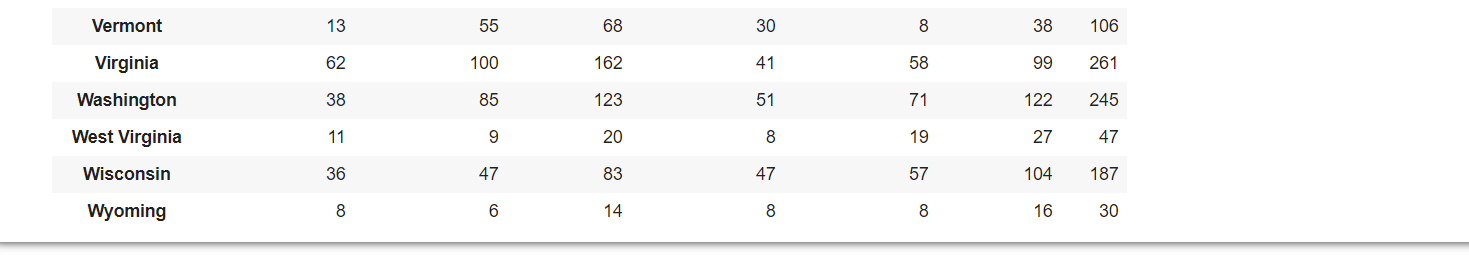


**Fig.4.25** Tweets for Joe Biden and Donald Trump statewise(ii)

1. Some states have more tweets than others so the dataset isn't well disseminated dataset. Presently that we have classified each tweet information positive and negative, we'll will see the number of positive and negative tweets statewise for each candidate.

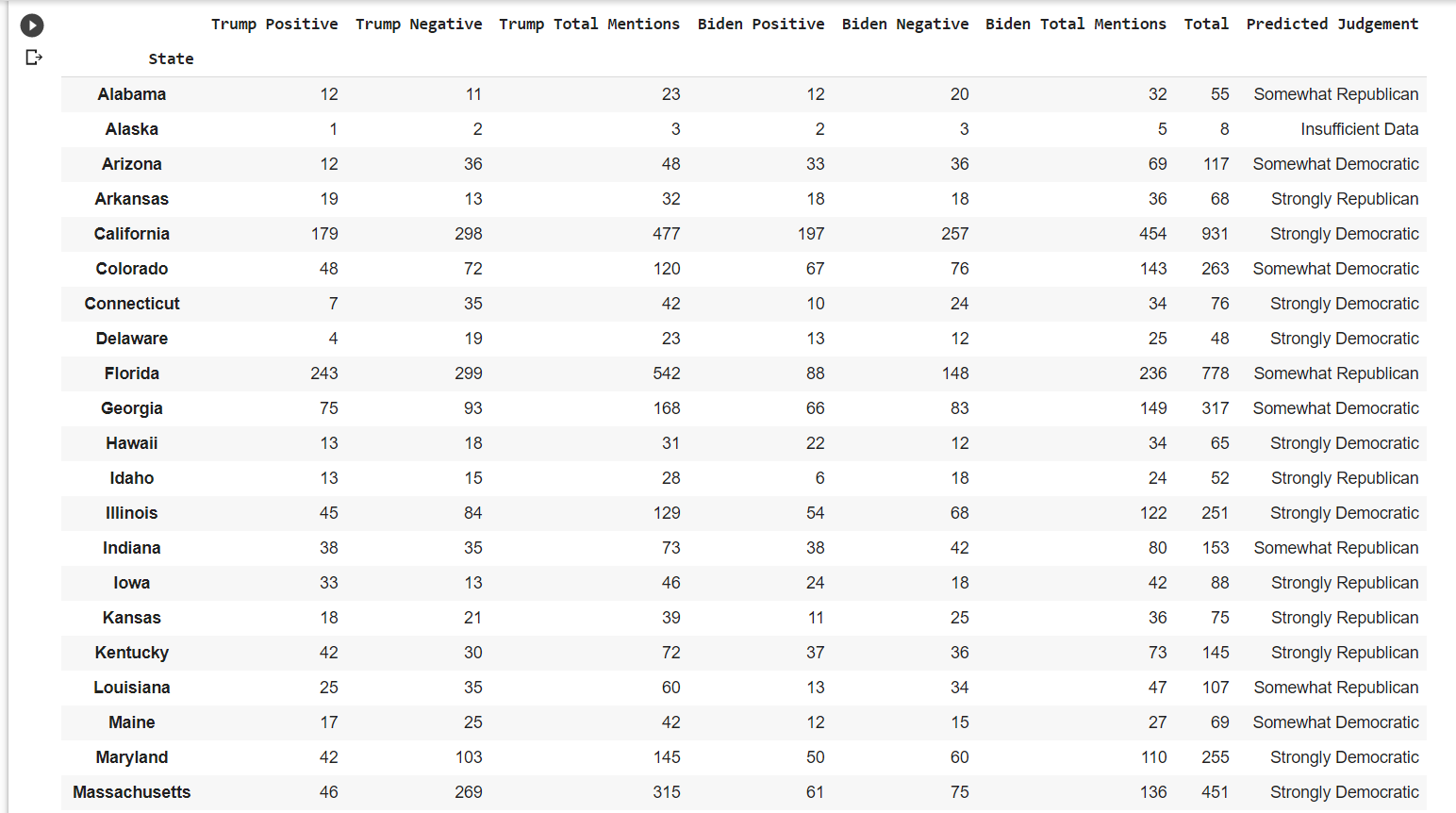
**Fig.4.26** Number of tweets for positive and negative sentiments for both candidate statewise(i)



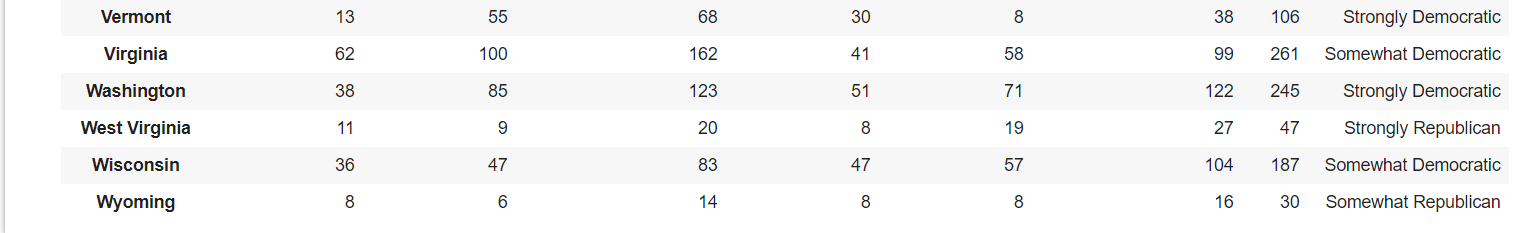


**Fig.4.27** Number of tweets for positive and negative sentiments for both candidate statewise(ii)

1. We use the above analysis to classify whether a state is one of the following. In a particular state if the number of positive tweets for a candidate surpasses the same for other candidate and the number of negative tweets are also less than other the candidate is strongly leading in that state. If the margin between positive tweets is more than the margin for negative tweets then there is mixed public opinion for both candidate for the particular state. For state where there are less than 15 tweets then the data is insufficient.



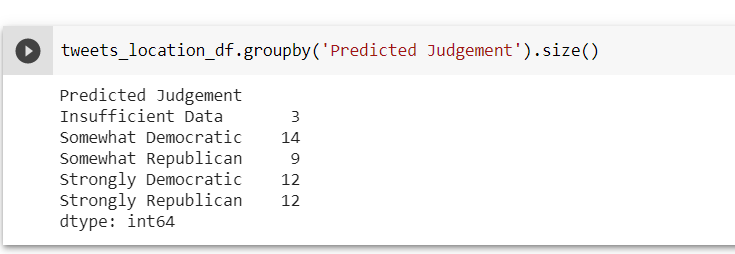




**Fig.4.28** Predicted Judgment for all states

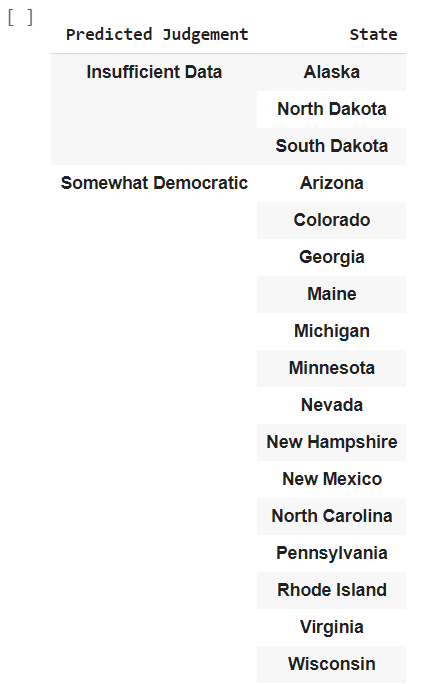
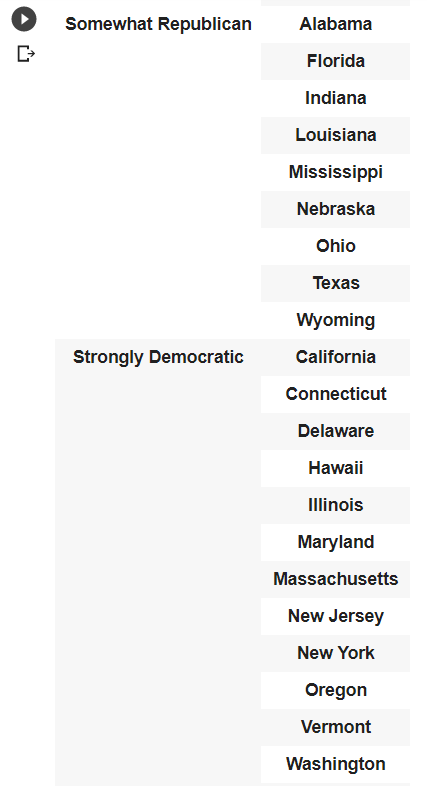
**4.8 Result and Discussion**

We predicted results for each and every state in USA. We got insufficient data for three states but for other states we were able to predict.

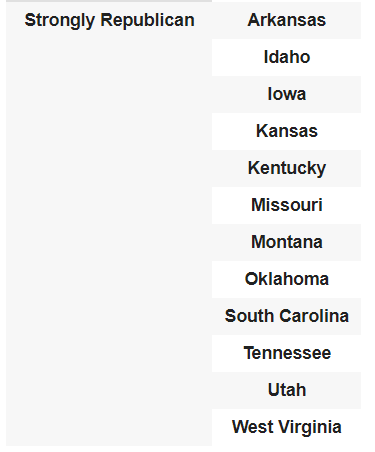


**Fig.4.29** Distribution of Predicted Judgment Data

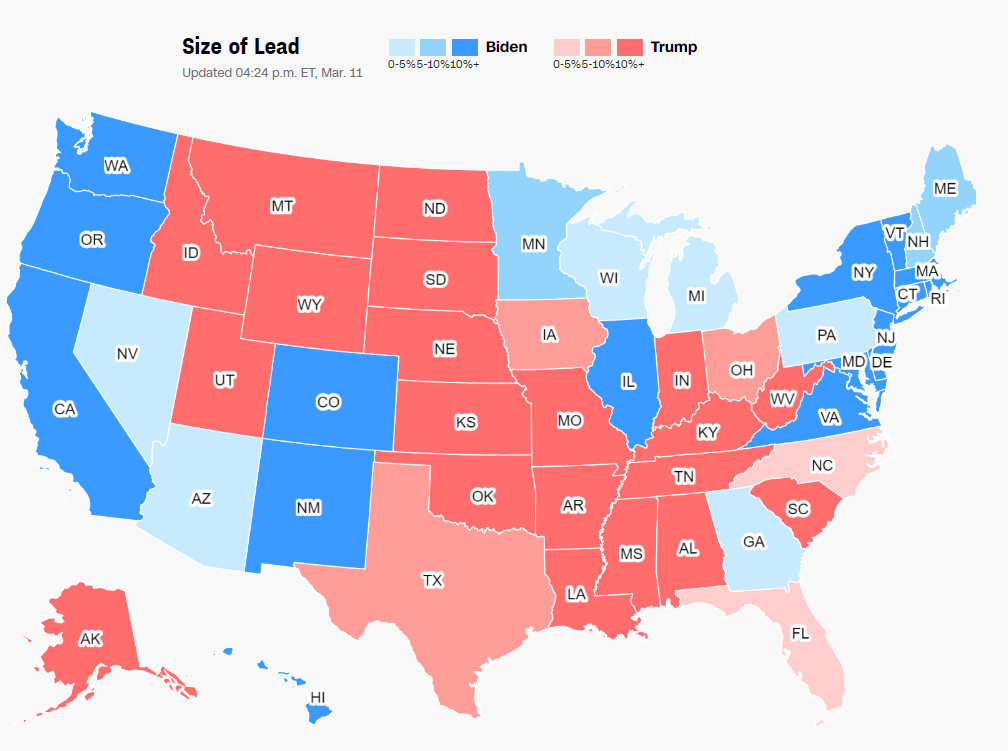
The results we obtained have been shown below

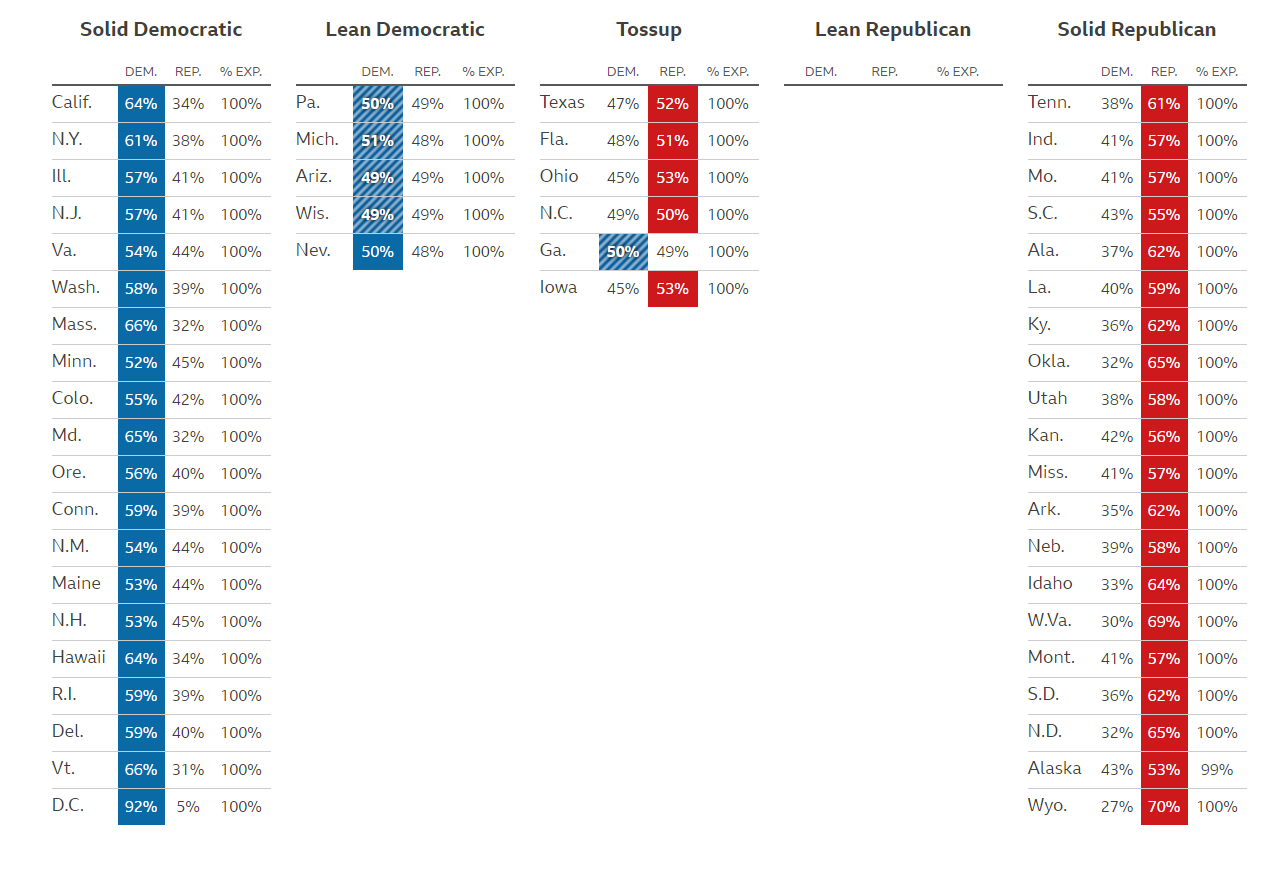
**Fig.4.30** States under Somewhat Democratic **Fig.4.31** States under Somewhat Republican and Insufficient Data Strongly Democratic



**Fig4.32.** States under Strongly Republican



**Fig.4.33** USA 2020 Results



**Fig.4.34** Actual results statewise from NY Times

|  |  |
| --- | --- |
| **Actual Results** | **Predicted Results** |
| California, Illinois, New York, New Jersey, Virginia, Washington, Massachusetts, Minnesota, Colorado, Maryland, Oregon, Connecticut, New Mexico, Maine, New Hampshire, Hawaii, Rhode Island, Delaware, Vermont | California, Illinois, New York, New Jersey, Washington, Massachusetts, Maryland, Oregon, Connecticut, Hawaii, Delaware, Vermont |

**Table 4.3** Comparison of Actual and Predicted Results for ‘Strongly Democratic’.

|  |  |
| --- | --- |
| **Actual Results** | **Predicted Results** |
| Pennsylvania, Michigan, Arizona, Wisconsin, Nevada. | Arizona, Colorado, Georgia, Maine, Michigan, Minnesota, Nevada, New Mexico, New Hampshire, North Carolina, Pennsylvania, Rhode Island, Virginia, Wisconsin. |

**Table 4.4** Comparison of Actual and Predicted Results for ‘Somewhat Democratic’.

|  |  |
| --- | --- |
| **Actual Results** | **Predicted Results** |
| Tennessee, Indiana, Montana, South Carolina, Alaska, Kentucky, Oklahoma, Utah, Kansas, Missouri, Arkansas, Nebraska, Idaho, West Virginia, Montana, North Dakota, South Dakota, Wyoming | Tennessee, Montana, South Carolina, Kentucky, Oklahoma, Utah, Kansas, Missouri, Arkansas, Idaho, West Virginia, Montana, Iowa, South Carolina |

**Table 4.5** Comparison of Actual and Predicted Results for ‘Strongly Republic’.

|  |  |
| --- | --- |
| **Actual Results** | **Predicted Results** |
| NA | Alabama, Florida, Indiana, Nebraska, Ohio, Texas, Wyoming, Louisiana |

**Table 4.6** Comparison of Actual and Predicted Results for ‘Somewhat Republic’.

Looking at the actual results, our model has predicted most of the states accurately with the help tweet sentiments. States where there was a close margin between either candidate came into somewhat democratic or somewhat republic category. Some of those states are :

1. Georgia
2. Arizona
3. Wisconsin
4. Pennsylvania
5. North Carolina
6. Nevada
7. Michigan
8. Florida
9. Texas
10. Minnesota
11. New Hampshire
12. Ohio
13. Maine

Other states where one candidate had a strong lead over the other were categorised into Strongly Democratic or Strongly Republic category. Some of them are:

1. California
2. Oregon
3. Wyoming
4. Utah
5. Virginia
6. New York
7. Iwoa
8. Illinois
9. Oklahoma
10. Washington

*Summary of this chapter…..*

*In this chapter we have implemented for machine learning models-Decision Tree, KNN, Random Forest and Logistic Regression. We have done hyperparameter tuning and selected the best model based on result comparators we have used. Then we did sentiment prediction on test data. After that the sentiments were filtered according to their location and hashtag. Prediction for election results for each state in USA was done.*

*In the next chapter, we shall discuss the conclusion and future scope of the project.*

N n

**Chapter 5**

**Conclusions and scope for further work**

*This chapter presents the final conclusion and the scope of twitter sentiment analysis for the future work.*

**5.1 Conclusions**

The objective to analyse twitter information on a 2020 USA elections with the help on user tweets the same topic is successfully completed. Estimation investigation and supposition mining require point by point information of the relation between twitter and Tweepy. Various libraries like numpy,matplotlib,nltk and pandas are moreover utilized for easier examination and visualization of the results obtained. In this paper we compared the results of four supervised classification algorithms. We inferred that Logistic Regression Algorithm was well performing among the four on our dataset. Tweets were accurately classified into positive and negative and filtered according to its subject and location. Public opinion was predicted for the USA 2020 presidental elections for each state and it was inferred that the democratic party has greater chance of winning than the Republic party. The democratic party is leading in the battleground states with a small margin.

**5.2 Scope for Future Work**

* Extending the project sentiment analysis can be done on similar social-media platforms like Instagram, Facebook, Snapchat, etc can be done.
* Image Sentiment Analysis is another challenging research in this domain.
* Thirdly, dataset should have same number of tweets for all states for both candidates. But within the real world, this can be troublesome to attain in the event that not incomprehensible. Be that as it may, some distribution models should be included for a uniform distribution of states.

*Summary of this chapter……*

*We discussed about Tweepy and different python libraries like Matplotlib, Pandas etc that can be used for data extraction and visualization of tweets. Four different supervised machine learning algorithms were stimulated and among all, Logistic Regression outperformed others. Tweets were perfectly classified into different sentiments and according to its subjects and location. It can also be inferred from the results that Democratic Party has greater chance of winning than the republic party. We also discussed future scope of the project that can improve quality of the project.*

* N n

**BIBLIOGRAPHY**

1.Prakruthi V, Sindhu D and Dr S Anupama, “Real Time Sentiment Analysis of Twitter Posts ” In the Proceedings of 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions. (CSITSS), IEEE Explore, December 2018.

2. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, “Sentiment Analysis of Twitter Data", In Proceedings of the ACL 2011Workshop on Languages in Social Media,2011 , pp. 30-38.

3.Po-Wei Liang, Bi-Ru Dai,”Opinion Mining on Social Media Data”, IEEE 14th International Conference on Mobile Data Management Milan, Italy, June 3-6, 2013, ISBN:978-494673-6068-5.

4. Pablo Gamallo, Marcos Garcia, “Citius: A Naïve Bayes Strategy for Sentiment Analysis on English Tweets”, 8th International Workshop on Semantic Evaluation, Dublin, August 2014.

5. R Xia, C Zong and S Li, “Ensemble of feature sets and classification algorithms for Sentiment Classification”, Information Sciences: an International Journal, Vol. 181.

6. Go, R Bhayani, L huang, “Twitter Sentiment Classification Using Distant Supervision”, Stanford University, Technical Paper 2009.

7. Dmitry Davidov, Ari Rappoport , “Enhanced Sentiment Learning Using Twitter Hashtags and Smileys”, Coling 2010: Poster VolumePages 241, August 2010.

8. Bifet and E Frank, “Sentiment Knowledge Discovery in Twitter Streaming Data”, In Proceedings of the 13th International Conference on Discovery Science, Berlin, Germany: Springer 2010.

9. Zhao Jianqiang and Gui Xiaolin,”Comparison Research on Text Pre Processing Methods on Twitter Sentiment Analysis” , In IEEE Research Journal., 22 February 2017, DOI: 1109/ACCESS.2017.2672677.

10. Rasika Wagh and Payal Punde, “Survey on Sentiment Analysis using Twitter Dataset”, In Proceedings of 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE, October 2018.

11. P. D. Turney, “Thumbs up or thumbs down ?: semantic orientation applied to unsupervised classification of reviews,” In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 2002.

12. A Pak and P. Paaroubek, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”, In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010.

13. R Parikh and M Movassate, “Sentiment Analysis of User Generated Twitter Updates using Various Classification Techniques” CS224N Final Report, 2009.

14. Lokesh Mandloi and Ruchi Patel, “Twitter Sentiment Analysis using Machine Learning Methods”, In Proceedings of 2020International Conference for Emerging Technology (INCET), IEEE, August 2020.

15.https://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

16.https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

17.https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

18.https://scikitlearn.org/stable/modules/generated/sklearn.linear\_model.LogisticRegression.html

19.https://www.tutorialspoint.com/python\_text\_processing/python\_tokenization.html

20.https://www.tutorialspoint.com/python\_data\_science/python\_stemming\_and\_lemmatization.htm

21.https://towardsdatascience.com/machine-learning-gridsearchcv-randomizedsearchcv-d36b89231b10

22. https://docs.tweepy.org

**APPENDIX A**

**Code for Data Pre Processing**

def drop\_features(features,data):

data.drop(features,inplace=True,axis=1)

def process\_tweet():

train\_data['processed\_tweets'] = train\_data['tweet'].str.replace('[^A-Za-z0-9 ]', '')

train\_data['processed\_tweets'] = train\_data['processed\_tweets'].str.replace('@[a-zA-Z0-9]+', '')

import re

re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])")

process\_tweet()

train\_data.head(10)

drop\_features(['id','tweet'],train\_data)

train\_data.head(10)

train\_data.info()

train\_data['processed\_tweets'] = train\_data['processed\_tweets'].astype(str).str.split()

train\_data.head(10)

from nltk.stem.snowball import SnowballStemmer

stemmer = SnowballStemmer("english")

train\_data['processed\_tweets']= train\_data['processed\_tweets'].apply(lambda x: [stemmer.stem(i) for i in x])

train\_data.head(10)

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

nltk.download("stopwords")

stopwords = nltk.corpus.stopwords.words('english')

import string

def process(text):

nopunc = set(char for char in list(text) if char not in string.punctuation)

nopunc = " ".join(nopunc)

return [word for word in nopunc.lower().split() if word.lower() not in stopwords]

train\_data['processed\_tweets'] = train\_data['processed\_tweets'].apply(process)

train\_data.head(10)

def join\_tokens(tweet):

return " ".join(tweet)

train\_data['processed\_tweets'] = train\_data['processed\_tweets'].apply(join\_tokens)

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(train\_data["processed\_tweets"],train\_data["label"], test\_size = 0.2, random\_state = 42)

from sklearn.feature\_extraction.text import CountVectorizer, TfidfTransformer

count\_vect = CountVectorizer(stop\_words='english')

transformer = TfidfTransformer(norm='l2',sublinear\_tf=True)

x\_train\_counts = count\_vect.fit\_transform(x\_train)

x\_train\_tfidf = transformer.fit\_transform(x\_train\_counts)

print(x\_train\_counts.shape)

print(x\_train\_tfidf.shape)

x\_test\_counts = count\_vect.transform(x\_test)

x\_test\_tfidf = transformer.transform(x\_test\_counts)

y\_test = np.nan\_to\_num(y\_test)

x\_test = np.nan\_to\_num(x\_test)

print(x\_test\_counts.shape)

print(x\_test\_tfidf.shape)

**APPENDIX B**

**Libraries Used**

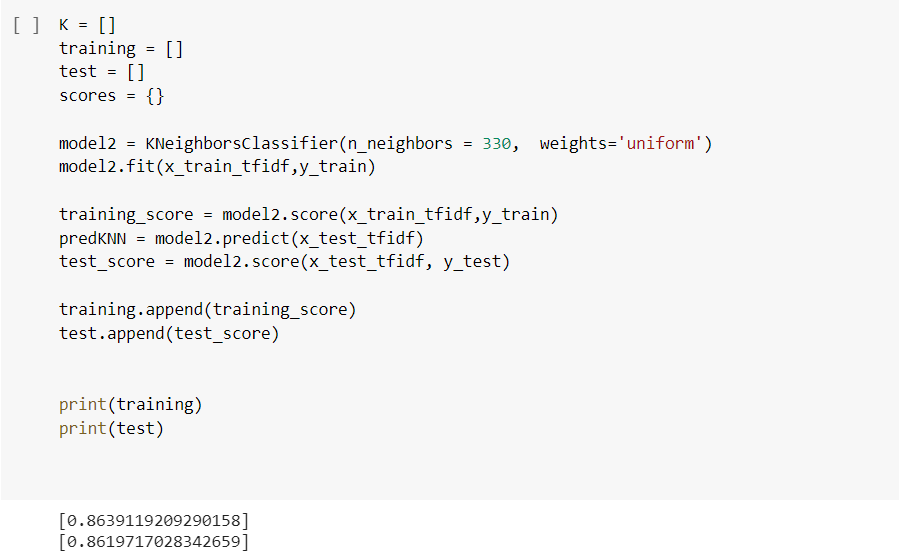
* 1. Pandas- Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of Python programming language.
  2. NumPy- It is an open source module of python which provides fast mathematical computation on arrays and matrices.
  3. Seaborn- Seaborn is a python data visualization library based on matplotlib. It provides a high level interface for drawing attractive and informative statistical graphics.
  4. Re- A regular expression or re specifies a set of strings that matches it, this module let you check if a particular string matches a given regular expression.
  5. SnowballStemmer- It is a stemming algorithm that reduces the word to its word stem that affixes to suffixes and prefixes or to roots of wards known as lemma.
  6. NLTK- NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.
  7. Sklearn.DecisionTreeClassifier- DecisionTreeClassifier is a class capable of performing multi class classification on a dataset.
  8. Sklearn.KNeighborsClassifier- This classifier implements learning based on the k nearest neighbors. The value of k is dependent on data.
  9. Sklearn.LogisticRegression- It is a machine learning classification algorithm that is used to predict the probability of a categorical dependent variable.
  10. Sklearn.RandomForest- 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.

**APPENDIX C**

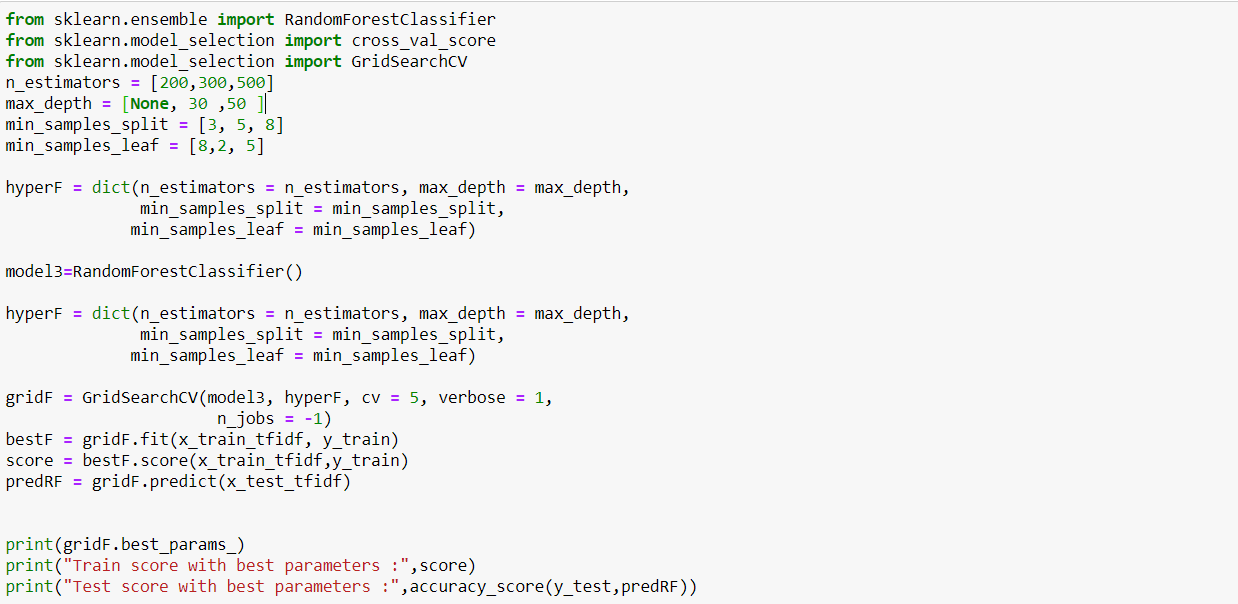
**Decision Tree Code**



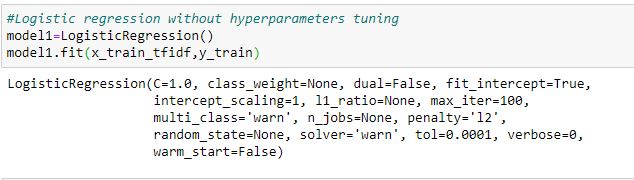
**K Nearest Neighbour Code**

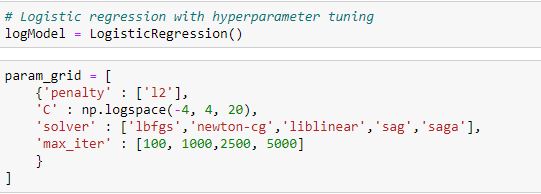


**Random Forest Code**



**Logistic Regression Code**





**ACKNOWLEDGEMENT**

Implementing the project titled ‘Twitter Sentiment Analysis’ was a great learning experience. Right from envisioning the project till building the final project it took a lot of experimentation and iteration, we not only learnt the concepts of Python and Machine Learning, but also learnt a lot about Time Management, Work Distribution, Team Work and Project Management.

We are particularly grateful to our internal project guide Prof. Jayshree Kundargi for her invaluable guidance, constructive insights and expertise that played a vital role in the advancement of the project. She has been a source of inspiration and a pillar of support. We are grateful to Dr. Shubha Pandit, Principal Madam, K.J Somaiya College of Engineering, Mumbai for providing us all facilities for our work, without which it would not have been possible to complete this project.

We would also like to acknowledge the motivation that the Head of our department, Dr. Ameya Naik provided us with, and we are thankful for the same.

We would like to thank the teaching and non-teaching staff that helped us in every possible way and also owe thanks to our friends who co-operated with us, helped us.