IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Analyzing Political Opinions and Prediction of Election Result of The Indian Election Using Data Mining Approach

[1]Ashweeni A. Kulkarni, [2]Mukta G. Dhopeshwarkar, [3]Pratik S. Jaiswal
[1]M.Tech Research Scholar, [2]Assistant Professor & Research Guide, [3]M.Phil Research Scholar
[1]Department of Computer Science and IT,
[1]Dr. Babasaheb Ambedkar Marathwada University, Aurangabad 431004, India

Abstract: Sentiment Analysis is considered in a category of natural language processing and machine learning. Sentiment analysis has been a popular field for records and scientists. It is a technique of calculating sentiment of a particular statement or sentence for Political review, Movie review, social media like twitter review, hotel review and categorizes them as positive, negative and neutral. Election is conducted to find the public opinion, where candidate choose by group of people using votes and many methods to predict result, Although many agencies and media companies conduct pre poll survey and expert views to predict result of election. We use twitter data to predict outcome of election by collecting twitter data and analyze it to predict the outcome of the election by analyzing sentiment of twitter data about the candidates. We used Machine learning and lexicon based approach to find emotions in tweets and predict sentiment score. We performed data (text) mining on Political and Election based generated tweets. We utilized Dictionary based approach, Naïve Bayes, Support Vector Machine and Decision Tree algorithm to build our classifier and classified the test data as positive, negative and neutral. We also utilized comparative study between classifier for better accuracy result. We identified the sentiment of Twitter users towards each of the considered Indian political leaders and national political parties. We begin to use the case study by selecting 3 National Parties attend in 17th Lok Sabha Election of Indian General Election. We indicate that Naive Bayes can perform anticipation and classification processes with high accuracy in compared with two other algorithms to anticipate participation.

Index Terms **- Sentiment Analysis, Naïve Bayes, Support Vector Machine, Decision Tree, Sentiment Analysis with Python, SentiWordNet dictionary, Indian Election.**

I. INTRODUCTION

Elections are of utmost importance in every Democratic country. As we all know, "DEMOCRACY IS FOR THE PEOPLE,BY THE PEOPLE ,OF THE PEOPLE", democracy is defined as a government of the people, for the people and by the people. All the powers in democracy are in the hand of people. Election is the process of voting to choose someone to be their political leader or the representative in government. People gives equal right for every citizen of the nation to select his leader. For every citizen, Vote is prominently considered in election and people are very concerned towards the winner of their choice, so they express their views on exit polls or opinion poll on results before election.

An opinion poll has existed since the early 19th century[1]. Currently, there are many scientifically proven statistical models to forecast an election[2]. But sometimes, even in the developed countries, the polls failed to accurately predict the election outcomes. Past research shows that several failed polls result such as in the 2004 European elections in Portugal, the 1992 British General Elections, 2007 French presidential elections, the 1998 Quebec Elections the 2006 Italian General Elections, the 2002 and the 2008 Primary Elections in the States.[3]

Latterly, it is observed that traditional polls may fail to make an accurate prediction. Then the scientific community has turned its interest in analyzing web data, such as blog posts or social networks users activity as an alternative way to predict election outcomes, hopefully more accurate.

Social media is a part of our daily lives from some years now. People increasingly tend to express their opinions via social media platforms. On a daily basis, data generated from social media is large volume of data. The question arrives for the future discussion is that can we use these data in order to detect trends, preferences, patterns and predict outcomes of future event? Social media is used for research and more specifically Twitter is used.. Twitter is considered one of the most successful social media. The community of the popular platform counts more than 328.000.000 active users at the moment [4]. Twitter is a microblogging web service that was launched in 2006. Now, it has more than 200 million visitors on a monthly basis and 500 million messages daily. The user of twitter can post a message (tweet) up to 140 characters. The message is then displayed at his/her personal page (timeline). Originally, tweets were intended to post status updates of the user, but these days, tweets can be about every imaginable topic. Based on the research in [5], rather than posting about the user's current status, conversation and endorsement of content are more popular. [5] The advantages of using tweets as a data source are as follows; first, the number of tweets is very huge and they are available to the public. Second, tweets contain the opinion of people including their political view.

II. DATABASE INFORMATION

A. Sentiment Analysis.

Sentiment analysis is a text mining technique that uses machine learning and natural language processing (NLP) to automatically analyze text for the sentiment of the writer in positive, negative and neutral [6]. Sentiment analysis allows to organizing text like customer feedback or product reviews or political leaders review, first by category (Features, Shipping, Customer Service, Politics etc.), and then mining text for sentiment so you can see which categories are positive or negative or neutral.

It allows you to analyze thousands of online reviews or social media comments in just minutes. However, before performing any kind of sentiment analysis, you'll need to break down comments, paragraphs, or documents, into smaller fragments of text. Suppose Political Party Opinion, for example, 'I like BJP' sentiment towards this statement is positive, 'But it seems really slow as comparing to other parties' sentiment towards this statement is negative, 'I am ok with BJP, lets see what they can do in future' sentiment towards this statement is neutral. Like this it also contains multiple ideas or opinions, to analyzing the overall sentiment of reviews, tweets, documents, and so on.

B. Sentiment Analysis on Twitter

The goal of Twitter sentiment analysis is to classify tweets into three categories: positive sentiment class, negative sentiment class, and neutral sentiment class. Similarly, further calculations and functions, such as the most commonly used phrases, commonplace phrases, most commonly used emoticon, and frequent sentiment in the midst of the data, may be computed.

As previously stated, performing sentiment analysis on Twitter data is difficult.

The following are the reasons behind this:[7]

1. Limited Tweet size: With just 280 characters to work with, succinct statements are constructed, resulting in a limited number of possibilities. It's also tough to examine the numbers because of the use of slang, acronyms, and emoticons in tweets.
2. Slang: These words are distinct from English terminology, and their use might age a method owing to the growth of slangs.
3. Features of Twitter: Hashtags, customer references, emojis, and URLs are all supported. These need to be processed differently than other phrases.
4. User Variety: Customers express their opinions through a variety of tactics, including the employment of unique language in between, as well as the use of repeated words or symbols to convey an emotion. Some people choose to use a sequence of emoticons to represent a visual snapshot, while others prefer to make sarcastic words that appear to be something else but are highly recommended.

In our project, we have used the Twitter dataset. In that, we are fetching tweets regarding Loksabha Election 2019 and the parties which are included in the Loksabha Election 2019. We have used the dataset of 60000 tweets. In that, we used three different datasets of 20000 tweets for each party like datasets for BJP, NCP, and AAP. Tweets are fetching on monthly basis from December 2018 to March 2019. After fetching tweets for every month then merge that data to our existing dataset. Every month we have created a newly updated dataset for each party for getting the more accurate result.

We are fetching tweets using the following keywrds :

- **BJP**

'BJP', 'BhartiyaJanataParty', 'modi', 'NarendraModi

- **NCP**

'congress', 'NCP', 'Gandhi, 'RahulGandhi

- **AAP**

'AAP', 'ArvindKejriwal', 'AamAadamiParty'

Following are the SearchQueries which are used for fetching tweets :

#Congresswins, #BJPwins, #AAPwins, #CongressvsBJP, #RahulvsModi, #Loksabha, #LoksabhaElection, # LoksabhaElection2019, #BJPforIndia, #ModivsGrandAllowance, #BJP2019, #ModiforPM2019, #GeneralElection2019, # GeneralElection, #RahulforPM2019, #AAPwins2019, #kejariwal, #ArvindKejariwal, #BJP, #AAP, #NCP

We are using tweets per query is 100.

While fetching tweets, we are also fetching the location of the user who tweets, the count of total hashtags used in that tweet, the count of users mentioned in that tweet, and also the count of URLs and symbols used in that tweet.

III. METHODOLOGY

When we create dataset it was not in line way , so we need to clean database in an efficient manner. When we done entire database cleaning process, database setup is prepare to perform particular experiment on it. Then we calculate sentiment score for all the three perties and apply classfier on that database. We used Naïve Bayes classifier, Support Vector Machine classifier, Decision Tree Classifier. Above figure shows the entire execution process from row database collection to the final output of the classifier.

Following is the flow chart of the experiment

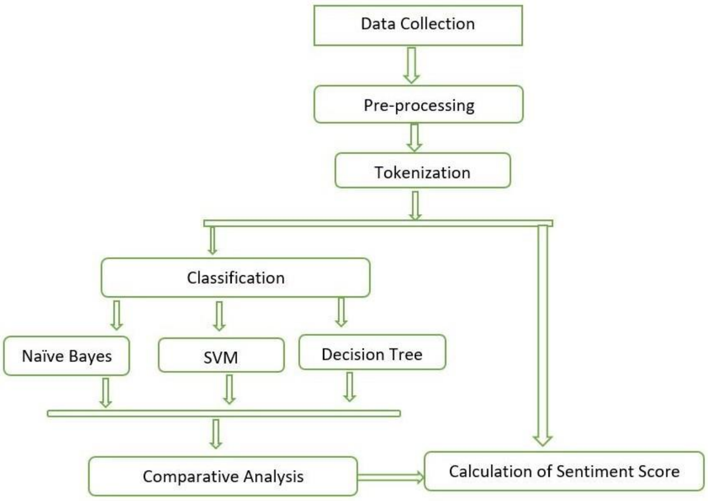


Figure : Flow Chart

IV. EXPERIMENT

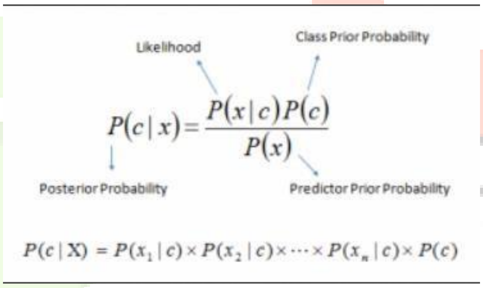
A. Naive Bayes

Naive Bayes is the easiest and fastest classification algorithm for a large chunk of data. In various applications such as junk mail filtering, textual content classification, sentiment analysis, and recommendation systems, Naive Bayes classifier is used successfully. It makes use of the Bayes chance theorem for unknown classification prediction.

The Naive Bayes classification approach is a simple and effective classification task in computer learning. The use of Bayes' theorem with a robust independence assumption between the facets is the foundation for naive Bayes classification. When used for textual information analysis, such as Natural Language Processing, the Naive Bayes classification yields true results.

Naive Bayes model is easy to construct and mainly beneficial for very giant records sets. Along with simplicity, Naive Bayes is recognised tooutperform even rather state-of-the-art classification methods.

Bayes theorem offers a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation under :



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

$$P(c\mid X)=P(x_1\mid c)\times P(x_2\mid c)\times\cdots\times P(x_n\mid c)\times P(c)$$

Above,

- P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

Naïve Bayes Algorithm flow :

i. first, we have created a training and testing dataset for each party.
ii. Creating pickle file of that dataset
iii. Data Preprocessing
iv. Sentiment Analysis using Lexicon-based Method
v. Naive Bayes with TF-IDF on original text data
vi. Naive Bayes with TF-IDF on pre-processed text data achieved very minimal accuracy improvement
vii. Finally, we got precision, Recall, F1Score, Support and Accuracy of our dataset

B. Support Vector Machine

A universal learner is the Support Vector Machine. Both the input and output formats for the Support Vector Machine have been established. The input is vector space, and the output is either positive or negative. The document's text is unsuitable for learning. These texts are formatted in a structured manner. The text is converted into a format that the machine learning system can understand. The texts' scores are computed, and the results are then fed into the Support Vector Machine. The Support Vector Machine has been proven to be one of the most powerful text categorization learning systems.

However, text classification can occasionally result in an error. A text classifier comparison is required to determine which is superior across texts. In this scenario, the performance metric is employed.

Support vector machine Algorithm flow :

i. first, we have created a training and testing dataset for each party.
ii. Creating pickle file of that dataset
iii. Data Preprocessing
iv. Sentiment Analysis using Lexicon-based Method
v. SVM with TF-IDF on original text data
vi. SVM with TF-IDF on pre-processed text data - achieved very minimal accuracy improvement
vii. Finally, we got precision, Recall, F1Score, Support, and Accuracy of our dataset

C. Decision Tree

The most powerful and widely used tool for categorization and prediction is the decision tree. A decision tree is a flowchart-like tree structure in which each internal node represents an attribute test, each branch reflects the test's conclusion, and each leaf node (terminal node) stores a class label. As decision trees are supervised algorithms, they must be trained using annotated data.

As a result, the main notion is the same as for any text classification: given a set of documents (for example, TFIDF vectors) and their labels, the algorithm will determine how strongly each word connects with each label.

Decision Tree Algorithm flow :

i. first, we have created a training and testing dataset for each party.
ii. Creating pickle file of that dataset
iii. Data Preprocessing
iv. Sentiment Analysis using Lexicon-based Method
v. Decision Tree with TF-IDF on original text data
vi. Decision Tree with TF-IDF on pre-processed text data achieved very minimal accuracy improvement
vii. Finally, we got precision, Recall, F1Score, Support, and Accuracy of our dataset.

We consider the following evaluation measures in order to compute the overall performance of the system.

	Positives	negatives
positives	True Positive(tp)	False Positive(fp)
negatives	False Positive(fp)	False Negative(fn)

1. **Precision:** Precision is defined as portion of true positive predicted instances among all positive predicted instances.
Precision = $tp/tp + fp$
2. **Recall:** Recall is calculated as portion of true positive predicted instances against all actual positive instances.
Recall = tp /tp + fn
3. **Accuracy:** Accuracy basically is the portion of true predicted instances against all predicted instances.
Accuracy = $tp+tn/tp+tn+fp+fn$
4. **F-measure:** F-measure is the combination of Presicion and Recall and is calculated as:
F-Measure = 2∗Precision∗Recall/Recall + Precision

Preprocessing data :

- Convert every tweets to lower case
- Remove Twitter username
- Remove punctuations, numbers and special characters
- Convert more than 2 letter repetitions to 2 letter (example (wooooooow --> woow))
- Remove extra spaces
- Remove URLs
- Emoji analysis
- Handle contractions words " can't " >> " can not "
" won't " >> " will not " " should't " >> " should not "
- Remove Stop word

The following pie chart shows the locations of the tweets that we downloaded from Twitter.

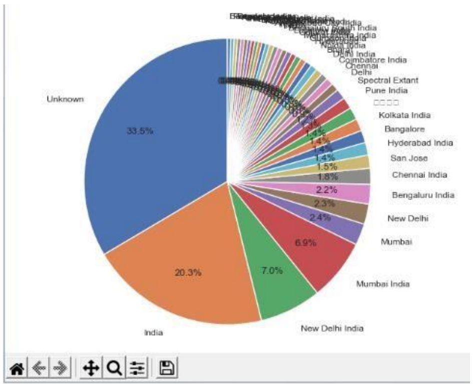


Figure: Tweets fetch for BJP from different location

4.1 BJP

The following bar chart shows the sentiment score and polarity of tweets for BJP. In that bar chart indicate nearly about 35% of Positive tweets, 21% of Negative tweets, 24% of Neutral tweets. The neutral intensity of the tweets indicates some of the users were neutral about the BJP they are neither positive nor negative.

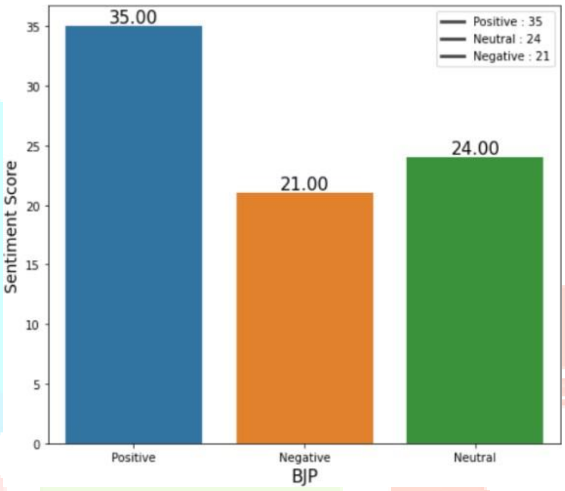


Figure: Sentiment Score for BJP

This is the word cloud for the positive tweets used for BJP. This word cloud we showing from our dataset.

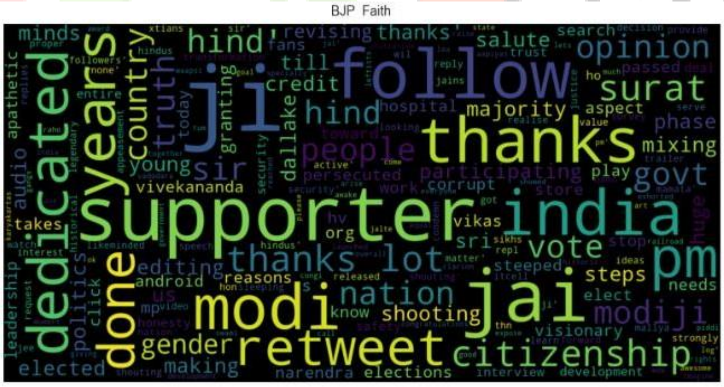


Figure : Positive word cloud for BJP

This is the word cloud for the Negative tweets used for BJP. This is the word cloud we showing from our dataset which indicates negative tweets.

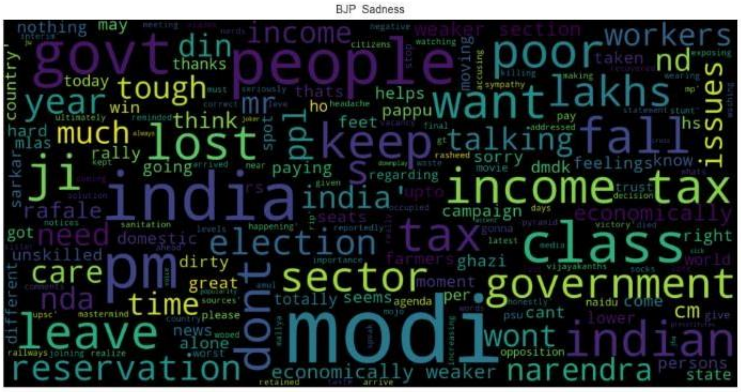


Figure : Negative word cloud for BJP

The following graph shows retweets detection, we are detecting tweets that are already used in our experiment and then removing that from our dataset. This will help to reduce the redundancy of the tweets and get a more accurate result.

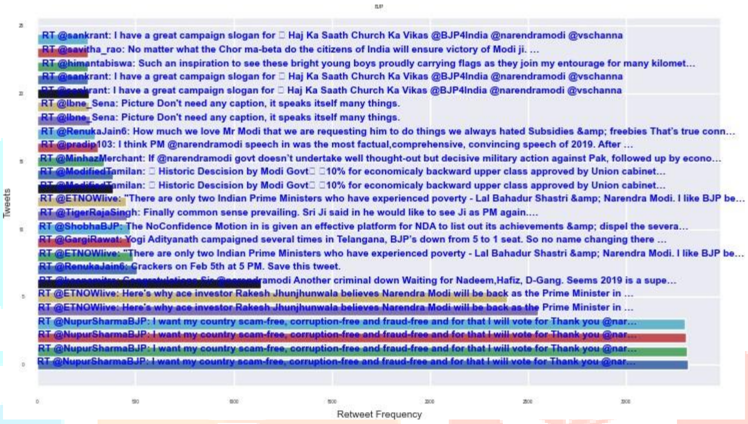


Figure : Retweets detection for BJP

Table 1 : Results using Naïve Bayes for BJP

	Precision	Recall	F1-Score	Support
positive	0.53	0.77	0.63	4
negative	0.56	0.75	0.64	9
Accuracy			0.87	100
Weighted avg	0.55	0.76	0.64	9

Table 2 : Results using SVM for BJP

	Precision	Recall	F1-Score	Support
positive	0.76	0.77	0.63	3
negative	0.75	0.75	0.64	8
Accuracy			0.79	100
Weighted avg	0.75	0.75	0.75	8

Table 3 : Results using Decision Tree for BJP

	Precision	Recall	F1-Score	Support
positive	0.42	0.53	0.47	3
negative	0.40	0.22	0.27	4
Accuracy			0.64	100
Weighted avg	0.41	0.33	0.39	3

4.2 NCP

The following bar chart shows the sentiment score and polarity of tweets for NCP. In that bar chart indicate nearly about 26% of Positive tweets, 35% of Negative tweets, 31% of Neutral tweets. The neutral intensity of the tweets indicates some of the users were neutral about the NCP they are neither positive nor negative.

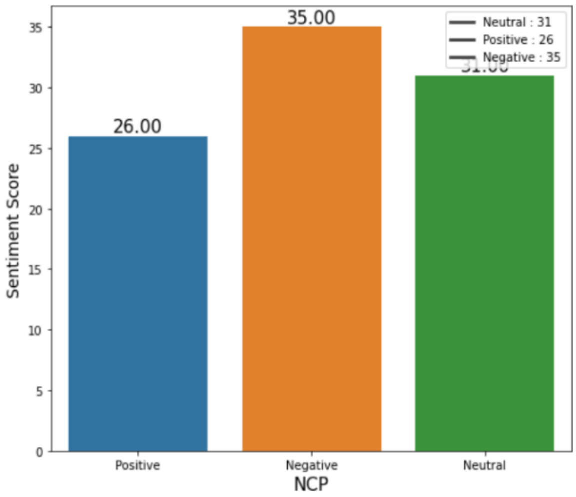


Figure: Sentiment Score for NCP

Table 4 : Results using Naïve Bayes for NCP

	Precision	Recall	F1-Score	Support
positive	0.42	0.55	0.48	5
negative	0.50	0.43	0.46	8
Accuracy			0.81	100
Weighted avg	0.46	0.51	0.47	6

Table 5 : Results using SVM for NCP

	Precision	Recall	F1-Score	Support
positive	0.41	0.76	0.53	5
negative	0.62	0.60	0.61	8
Accuracy			0.70	100
Weighted avg	0.52	0.65	0.55	7

Table 6 : Results using Decision Tree for NCP

	Precision	Recall	F1-Score	Support
positive	0.34	0.66	0.45	5
negative	0.33	0.12	0.19	1
Accuracy			0.64	100
Weighted avg	0.30	0.41	0.34	3

4.3 AAP

The following bar chart shows the sentiment score and polarity of tweets for AAP. In that bar chart indicate nearly about 24% of Positive tweets, 28% of Negative tweets, 31% of Neutral tweets. The neutral intensity of the tweets indicates some of the users were neutral about the AAP they are neither positive nor negative.

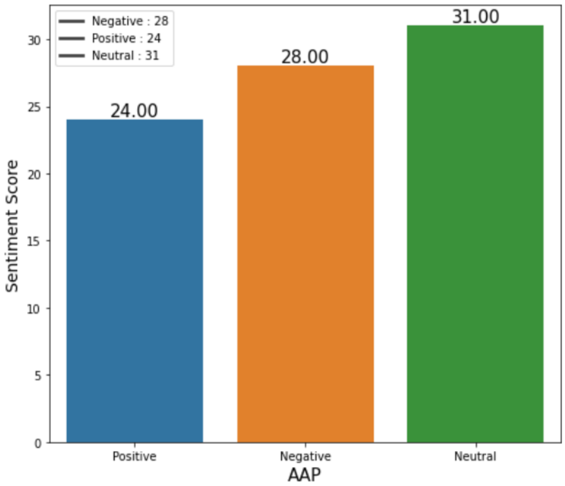


Figure: Sentiment Score for AAP

Table 7 : Results using Naïve Bayes for AAP

	Precision	Recall	F1-Score	Support
positive	0.47	0.72	0.57	3
negative	0.50	0.42	0.46	5
Accuracy			0.72	100
Weighted avg	0.45	0.63	0.50	5

Table 8 : Results using SVM for AAP

	Precision	Recall	F1-Score	Support
positive	0.43	0.72	0.54	6
negative	0.49	0.55	0.52	5
Accuracy			0.70	100
Weighted avg	0.46	0.66	0.53	6

Table 9 : Results using Decision Tree for AAP

	Precision	Recall	F1-Score	Support
positive	0.48	0.41	0.44	6
negative	0.32	0.67	0.42	3
Accuracy			0.51	100
Weighted avg	0.41	0.52	0.40	4

- **COMPARATIVE STUDY OF DATA MINING CLASSIFIER :**

We are using Naïve Bayes Classifier, Support Vector Machine, and Decision Tree classifier to calculate the result accuracy of Tweets regarding BJP, NCP, AAP.

We got following results:

Using Naïve Bayes Classifier for the tweets for BJP accuracy is 87%, for NCP accuracy is 81%,for AAP accuracy is 72%.

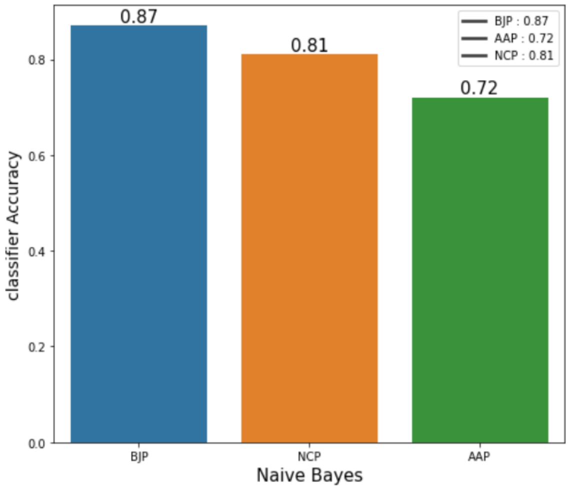


Figure : Naïve Bayes classifier result for BJP, NCP, and AAP

Using Support Vector Machine Classifier for the tweets for BJP accuracy is 79%, for NCP accuracy is 47%,for AAP accuracy is 61%.

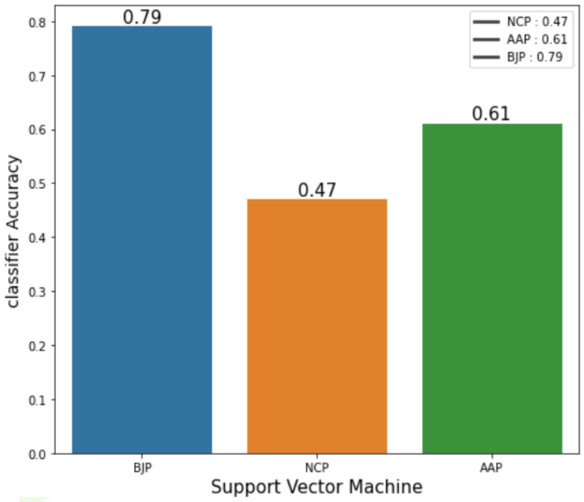


Figure: SVM classifier result for BJP, NCP, and AAP

Using Decision Tree Classifier for the tweets for BJP accuracy is 64%, for NCP accuracy is 65%, for AAP accuracy is 51%.

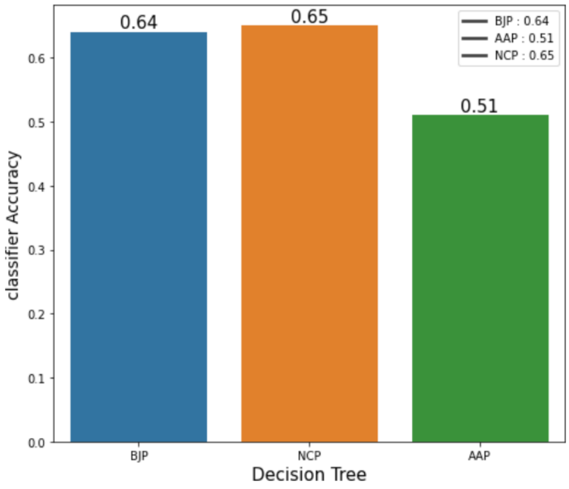


Figure: Decision Tree classifier result for BJP, NCP, and AAP

V. RESULTS AND DISCUSSION

On the basis of the results of our experiment we got the highest accuracy for the Naïve Bayes classifier, Using Naïve Bayes Classifier for the tweets for BJP accuracy is 87%, for NCP accuracy is 81%,for AAP accuracy is 72%.
Using Support Vector Machine Classifier for the tweets for BJP accuracy is 79%, for NCP accuracy is 47%,for AAP accuracy is 61%.
Using Decision Tree Classifier for the tweets for BJP accuracy is 64%, for NCP accuracy is 65%, for AAP accuracy is 51%.
Also we got highest positive sentiment for BJP is 35% , for NCP is 26% , and for AAP is 24%.
So, according to this experiment Naïve Bayes Classifier is better classifier to got accurate result.

VI. CONCLUSION

In our project, we have mainly focused on 2 approaches The first approach involved is the highest positive sentiment score for the party participating in the loksabha election 2019. The second approach is using a data mining algorithm we calculate sentiment score and accuracy from our tweet dataset of three parties and also show comparative about classifiers for which classifier is the better classifier.

Also, among the three classifiers i.e Naïve Bayes classifier, SVM classifier, and Decision Tree classifier, the Naïve Bayes classifier proves to generate a better result than the others.

According to our experiment, we got a total number of positive tweets for BJP to be 36%, congress to be 27%, and AAP to be 25%. As we are calculating the winning in the Lok Sabha election and the major parties contesting are BJP and congress we can clearly see that BJP is winning over the people's hearts and according to that BJP wins the Lok Sabha election.

On the basis of these classifiers, BJP has the highest accuracy in all three classifiers. Hence according to that BJP won the Lok Sabha election.

VII. LIMITATIONS

When we consider the total population we are not considering the people eligible to vote. There are many people whose views might change during the election. There are states with so less population that their vote affecting is negligible in this method. Because the long retweets couldn't be fully recovered, they were represented with "...", which the computer interprets as neutral sentiment. Due to a misuse of semantics, sarcasm was not detected in several statements. The Twitter search API could only obtain data from the last seven days. In other circumstances, using hashtags as a shorthand depiction of the party produced unclear outcomes. It's impossible to attain 100 percent accuracy while analysing tweets.

REFERENCES

[1] Hillygus, D. S. (2011). The evolution of election polling in the United States. Public opinion quarterly, 75(5),, 962-981.

[2] Lewis Beck, M. S. (2005). Election forecasting: principles and practice. The British Journal of Politics & International Relations,7(2), 145-164.

[3] Fumagalli, L. &. (2011). The total survey error paradigm and pre-election polls: The case of the 2006 Italian general elections. ISER Working Paper Series. 2011-29.

[4] "Number of monthly active Twitter users worldwide from 1st quarter 2010 to 1st quarter 2017 (in millions)" Link: https://www.statista.com/statistics/282087/number-of- monthly-active-twitterusers/ [Accessed 2/2/2018].

[5] Pratik Jaiswal, Mukta Dhopeshwarkar, Mangesh Patil, Anupriya Kamble, Gajanand Boywar, Ramesh R. Manza, and Surekha B. Jaiswal, "Identification of Educationally Backward Countries in Primary, Secondary and Tertiary Level Students by Using Different Classification Techniques", Springer Nature Singapore Pte Ltd. 2021 V. S. Rathore et al. (eds.), Rising Threats in Expert Applications and Solutions, Advances in Intelligent Systems and Computing 1187, https://doi.org/10.1007/978-981-15-6014-9_91

[6] Dann, S. (2010). Twitter content classification. First Monday, 15(12).

[7] Bahrainian S.-A., Dengel A., "Sentiment Analysis and Summarization of Twitter Data", 16th IEEE International Conference on Computational Science and Engineering, pp. 227-234, Sydney, Australia, December 2013.