Predicting the outcome of an Election Using data of Social Media

Mayank Aggarwal (2014152)
Student of
Graphic Era University
566/6, Bell Road, Society Area,
Clement Town, Dehradun,
Uttarakhand 248002
Email: mayankanshuaggarwal26@gmail.com

In recent years a quick and sudden burst in the users of social media has opened a door or multiple research problems. Elections are a hot topic all over the world and all the political parties have started using social media heavily for their political gain. In this project we have collected Election tweets for a period of 51 days , used NLP to understand the opinions of public and predicted the outcome of Delhi Election.

1 Motivation

In many articles the Deep Learning architectures and NLP tasks are treated. These articles use a technical vocabulary, sometimes understanding can be difficult. At this point the question arises: How do they work? Why have they achieved so much success? These have achieved great success in recent years because there is no need to make feature engineering. In Natural Language Processing (NLP) several DL architectures have been proposed to solve many tasks, from speech recognition to analysis. Many classic NLP tasks, such as speech recognition (PoS) and Named Entity Recognition (NER) tags, can be solved as a sequence labeling problem.

2 Tasks

In this section, known tasks of the NLP will be dealt with SVM. The tasks that we will deal with are: Hashtag Extraction , User Mentions Extraction , sentiment classification tasks and finally converting opinions into vote for the predicting the result of election.

3 Methodology

First collected tweets of Delhi Election 2020 during the time period of January and February date wise. And store all the essential information that twitter will provide in a tweet into a csv file.
We mainly focused on the text field , language field (in which language tweet is tweeted) , geo coordinates (if present) and its retweet details..
After dealing with above task created a python script to create a list of languages that are used in tweets.. The main reason of creating the list of languages to count in how many languages public can give their opinion and which language is highly used. According to that list date wise separated only tweets which are written in English because frequency of English is very high as compare to other languages.
After separating all the tweets according to our need created a python script to collect all the hashtags and mentions with their frequency so that we can clearly identify what are the trending hashtags and user mentions
Then manually created three lists, First list contain all the hashtag and user mentions related to AAP, second for BJP and third for INC. Now for the final opinion extraction and converting opinions into vote separate tweets into 3 different sections. First section will contain only those tweets in which keywords present related to AAP political party only. In similar way second section will contain BJP tweets and third section with INC tweets
Finally we converted all the positive opinions into votes using NLP for the election result prediction.

3.1 SVM

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.

3.1.1 How Does SVM Work?

The basics of Support Vector Machines and how it works are best understood with a simple example. Let's imagine we have two tags: red and blue, and our data has two features: x and y. We want a classifier that, given a pair of (x,y) coordinates, outputs if it's either red or blue. We plot our already labeled training data on a plane:

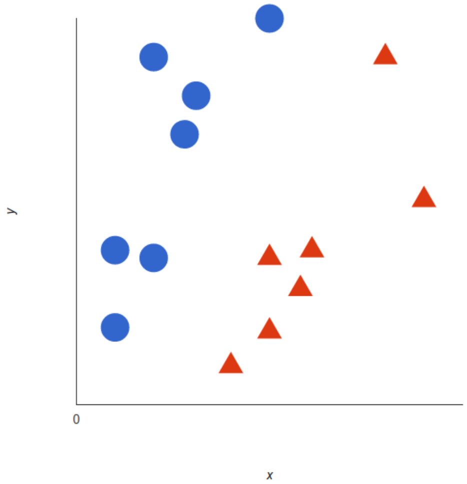


Fig. 1: Our labeled data

A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it's simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red.

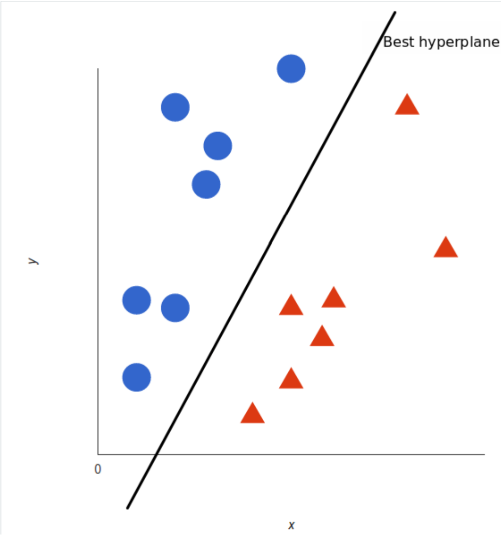


Fig. 2: In 2D, the best hyperplane is simply a line

But, what exactly is the best hyperplane? For SVM, it's the one that maximizes the margins from both tags. In other words: the hyperplane (remember it's a line in this case) whose distance to the nearest element of each tag is the largest.

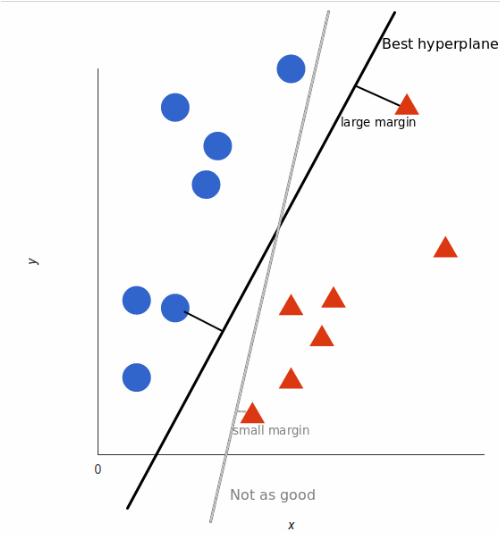


Fig. 3: Not all hyperplanes are created equal

3.1.2 How in NLP?

So, we can classify vectors in multidimensional space. Great! Now, we want to apply this algorithm for text classification, and the first thing we need is a way to transform a piece of text into a vector of numbers so we can run SVM with them. In other words, which features do we have to use in order to classify texts using SVM?
This means that we treat a text as a bag of words, and for every word that appears in that bag we have a feature. The value of that feature will be how frequent that word is in the text.

This method boils down to just counting how many times every word appears in a text and dividing it by the total number of words. So in the sentence “All monkeys are primates but not all primates are monkeys” the word monkeys has a frequency of 2/10 = 0.2, and the word but has a frequency of 1/10 = 0.1 .
For a more advanced alternative for calculating frequencies, we can also use TF-IDF.

Now that we've done that, every text in our dataset is represented as a vector with thousands (or tens of thousands) of dimensions, every one representing the frequency of one of the words of the text. Perfect! This is what we feed to SVM for training. We can improve this by using preprocessing techniques, like stemming, removing stopwords, and using n-grams.

3.2 What's New

We have gone through many past work based on Election Forcasting, so what we concluded is while downloading tweets with the help of Twitter API keys it will give tweets according to any query,hashtags , user mentions and so on. But there is one major disadvantage of such procedure in Election Forcasting. Lets have a look with an example. If I will download tweets which contains hashtags narendraModi so the tweets which are downloaded will contain narendraModi but it will contain some more hashtags also like ArvindKejriwal and during cleaning process there will be a loss of these hashtags and it may be a possibility that tweet is for ArvindKejriwal but we consider it for narendraModi. For the election narendraModi and ArvindKejriwal both are competitors.

To overcome this disadvantage we collected all the hashtags and user mentions , counted the frequencies and distribute these hashtags and user mentions belonging to their respective political parties. And we separated only those tweets which contain only one of the three political parties related mentions and hashtags . We tried to mimic target based analysis by doing above process so that we can correctly find which political party is targeted in a tweet ,then find sentiments and converted all positive tweets into votes for the political party.

4 Conclusion

After successfully applying SVM with an accuracy of 82 percent , We converted all the positive opinions into votes for the respective parties and plotted bar graphs to show on which day which political party is on top.

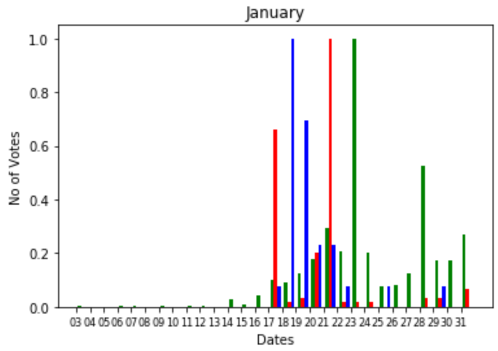


Fig. 4: Vote Variation of month January

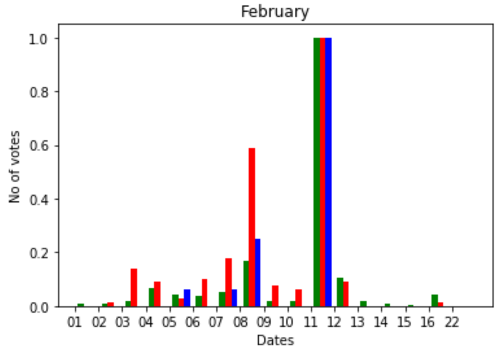
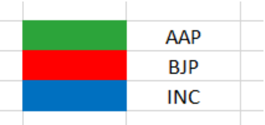
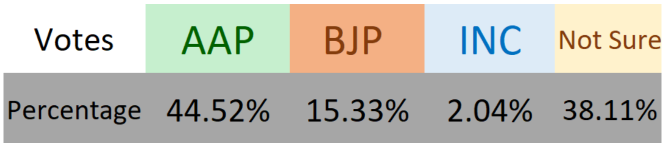


Fig. 5: Vote Variation of month February



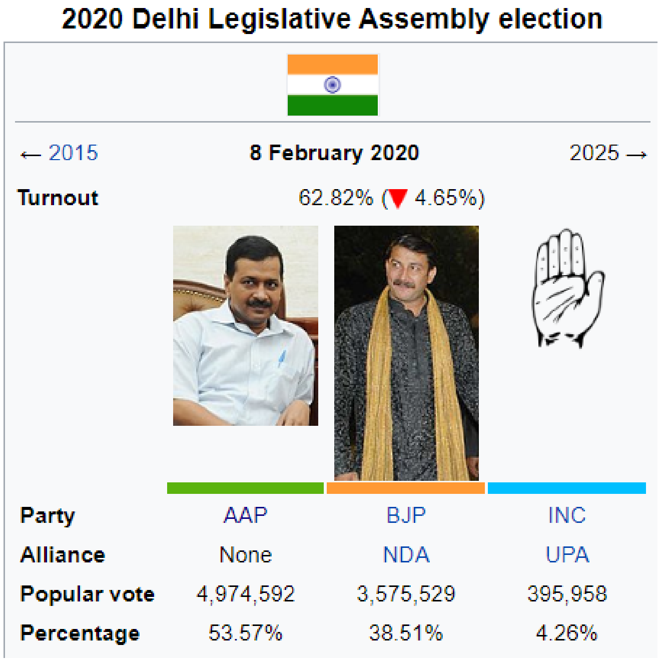
🟩	AAP
🟥	BJP
🟦	INC

And the final predicted result of Delhi 2020 Assembly Election result is -



Votes	AAP	BJP	INC	Not Sure
Percentage	44.52%	15.33%	2.04%	38.11%

And the original result of Delhi 2020 Assembly Election result is -



2020 Delhi Legislative Assembly election

← 2015 | **8 February 2020** | 2025 →

Turnout 62.82% (▼ 4.65%)

Party	AAP	BJP	INC
Alliance	None	NDA	UPA
Popular vote	4,974,592	3,575,529	395,958
Percentage	53.57%	38.51%	4.26%

Here out of total voters 44.52 percent are in favour of AAP , 15.33 percent in favour of BJP and 2.04 percent in favour of INC. There is one more category of not sure. These are those voters about whom we are not sure that they will vote for party whom they mentioned in their tweets. If we distribute 38.11 percent equally to all political parties then also the winner is AAP