US Elections Tweets 2020 DATA ANALYSIS

Seemal Tausif
Department of Computer Science LUMS
Lahore, Pakistan
24100024@lums.edu.pk

Zara Tahir
Department of Computer Science LUMS
Lahore, Pakistan
21030026@lums.edu.pk

Zoha Hayat
Department of Computer Science LUMS
Lahore, Pakistan
24100010@lums.edu.pk

***Abstract*—The aim of this project is provide complete data analysis for the US Election Tweets 2020 dataset.**
***Index Terms*—Data Mining**

I. Introduction

We are analysing the US Election Tweets from the year 2020 to find correlations between different components of the tweets made. Every component analysed provides some information about the popularity of the candidates for the elections. To make the data usable for exploratory data analysis, the data is first preprocessed to remove null values or to replace them with the mode of the data. Afterwards, a thorough data analysis is carried out to find relations of the candidates by carrying out location wise analysis, sentiment analysis and other exploratory data analysis.

II. Over View of Data set

The given dataset contains about 1,747,805 tweets of US election 2020. This dataset was generated by Twitter API. Data of two candidates, Trump and Biden is provided for this project. The fields that are collected are as follow:

- collected at: Date and time of tweet creation
- Tweet id: Unique ID of the tweet
- tweet: Full tweet text
- retweet count: Number of retweets
- likes: Number of likes
- source: Utility used to post tweet
- user id: User ID of tweet creator
- user name: Username of tweet creator
- user description: Description of self by tweet creator
- user screen name: Screen name of tweet creator
- user join date : join date of tweet creator
- user followers count: Followers count on tweet creator
- user location: Location given on tweet creator's profile
- long: Longitude parsed from user location
- lat: : Latitude parsed from user location
- city: City parsed from user location
- country: country parsed from user location
- continent: continent parsed from user location
- state: State parsed from user location
- state code: State code parsed from user location
- collected at: Date and time tweet data was mined from twitter
- Candidate: Tweet for the candidate

III. Prepossessing

Before carrying out data analysis, it is important to visualize the data, deal with missing values, remove unwanted columns, impute data, clean data entries and extract useful information. The detailed description of the prepossessing performed in this phase is discussed in this section.

A. *Data Visualization for Missing values*

The first and foremost step of data analysis is finding and handling missing values in the data set. This will help to identify columns that require imputation or removal of rows or columns based on percentage of data. On analyzing the data the percentage of missing values against each column is given below.

- collected at: 0%
- Tweet id: 0%
- tweet: 0%
- retweet count: 0%
- likes: 0%
- source: 0.09%
- user id: 0%
- user name: 0.0019%
- **user description: 10**%
- user screen name: 0%
- user join date : 0%
- user followers count: 0%
- **user location: 30**%
- **long: 54**%
- **lat: 54**%
- **city: 76**%
- **country: 54**%
- **continent: 54**%
- **state: 67**%
- **state code: 69**%
- collected at: 0%
- Candidate: 0%

Fig 1 below shows visualization of missing values before prepossessing.

B. *Handling Missing values*

As each column that has missing values has a different type so using a single type of technique for handling missing values or removing all the rows for missing values will not

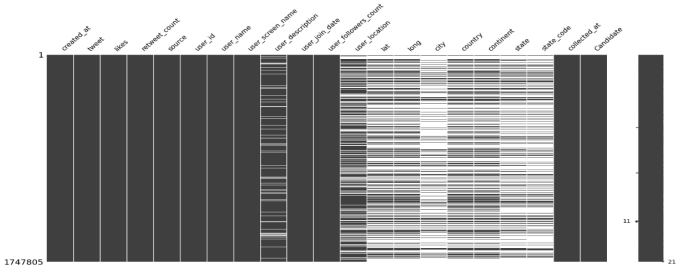

Fig. 1. Missing Values visualization before prepossessing

work for this data set. We had to handle missing values in each column separately using different techniques. The details of handling missing values for each type of section is discussed below.

*1) **Source Name**:* Source Name represents the source from where the tweet was collected. There are 1037 unique source names, of which **Twitter Web App** is most common so all null values in the column are replaced with it. As the percentage of missing values is only 0.09%, we could have dropped the rows as well.

*2) **User Name**:* Although, the number of nulls in User Name are very few but instead of removing these rows, we found a User Name against each User Id. Then this User Name was used in place of each null value in a row against that Id. If there were still null values after this, we filled this column by using User Screen Name.

*3) **User Description**:* The 10 % of nulls in user description were removed by assigning a single user description against each User ID. If there are still nulls, they are replaced with 'unknown' place holder.

*4) **Country, Continent and City**:* The missing values in the country and continent columns were handled in similar ways. Since user screen name are unique with user Id and there can only be one associated country and continent with each user screen name, we replaced the null values of the country and continent with the already existing information about these columns from the same user name. This reduce the nulls from 54.4% to 54.1%. The rest of the nulls are replaced with a place holder 'Geo Data N/A'. With city codes, if null values remained after filling them in using user screen name, we grouped the city based on country and took the mode of which city had the most tweets. We then replaced the null values in the city with this mode value and the remaining nulls are replaced with a placeholder 'Geo Data N/A'.

*5) **State and State Codes**:* A similar process as above was followed to handle the missing values in states and state codes. Since user screen names are unique, any null values in state column were filled in by the value of state for the same user name elsewhere. If null values still remained, we grouped the states based on country and took the mode of the of the number of tweets in each state. We then replaced the null values of state with the mode of the state of that country. This reduces null from 66.67% to 54.3%. The rest of the nulls are replaced with a place holder 'Geo Data N/A'. With state codes, if null values remained after filling them in using user screen name, we grouped the state codes based on state and took the mode of which state code had the most tweets. We then replaced the null values in the state codes with this mode value and the remaining nulls are replaced with a placeholder 'Geo Data N/A'.

*6) **Latitude and Longitude**:* The missing values for these columns were more than 50%. This meant that removing these null values would ultimately result in removing all the entries of the column. Hence, we did not remove them because the coordinates for latitude and longitude were needed later for plotting a geo-location map of the location of tweets.

*7) **Data Visualization after Filling Missing values**:* The data visualization after filling missing value is follow:

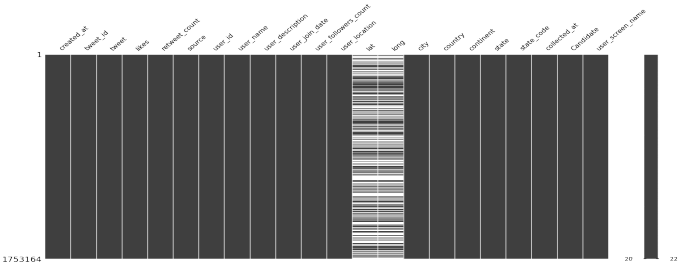

Fig. 2. Missing Values visualization after prepossessing

C. Transformation and Data cleaning

Various transformation and data cleaning techniques are applied to the columns to remove noise from the data. Also, some columns were removed as they were either redundant or play no role in data analysis.

*1) **Noise Removal in User Screen Name**:* While analysing the data it was found that for a single User Id more than one user screen name were used. The user might have changed the name over time. So to make user screen name and user id consistent, only one user name is assigned to each user based on User Id.

*2) **Extracting Information from user description column**:* On closely observing this column, we found that there were some rows containing user description as 'account is temporarily unavailable'. Which means that these accounts have been de-activated. So in order to analyse tweets and popularity of in-active users, a column is added named as **in-active** which is true if the description has the above

provided string else it is marked as false. We have also extracted split form of date to make timeline based anaylsis.

*3) **Extracting hash tags and '@' from the tweets**:* Each tweet has number of hash tags and mentions in the tweets which provides useful information regarding popular hashtags of this time, so these were extracted in different column named as "hash_tags" and "at".

*4) **Dropping the Columns**:* The user location, collected at, user join date, user description, user id, user name are the columns that are removed due to the reason provided below.

- **user id:** As we have defined one to one mapping of User Screen and User Id, this means that both of them represent the same information, hence we are keeping only one of them.
- **user name:** The user screen name uniquely identifies a user, hence keeping two names makes no sense.
- **user description:** The only useful information user description provides is whether the account is active or not which we have already extracted as a separate column. Thus, there is no need to keep this column.
- **user join date:** We could have used user join date for analysis, but that analysis would have been more general and not specific to the domain, like number of followers based on joining date etc. Such an analysis would not contribute in finding the correlation between the election campaign and the tweet contents hence we removed this column.
- **user location:** This column mostly contains noise, besides this we used other location columns to analyse geographical information of tweets, so we removed it.
- **collected at:** The date at which the tweets were collected doesn't provide any significant information about the data except that the data was collected in 2020 which is evident for the project under consideration. Hence this column was removed.

*5) **Changing Type of Attribute**:* On observing the data types of columns we found that instead of Integer and float type attributes, all the attributes have type object. The string type object can be easily interpreted with object type as well. So only 'created_at' is type casted to date-time type.

*6) **Removing Duplicates**:* While considering all the attribute together there were no duplicates rows, but when only tweets were consider there are 240600 duplicates. These are due to retweets. In overall evaluation we have not removed them but on sentiment based analysis, these are removed as a tweet can be assigned to both Biden and Trump as seen in the data.

D. Cleaning tweets

On observing the tweets, we found that there was much noise in the tweet, like emojis, urls, hashtags and mentions which makes the tweet dirty. An example of dirty tweet is given below.

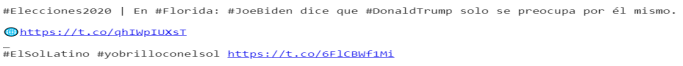

Fig. 3. Dirty Tweet

Before we move on to the actual sentiment classification of tweets, there is some data cleaning to be done. As a matter of fact, this step is critical, as it ultimately effects the learning process and performance of the model. Following are the prepossessing steps done in order to clean the tweets. A clean tweet can be seen in Fig 4.

- All the hashtags are removed from the data.
- All the '@' symbols are removed from the tweet.
- All the punctuation's are removed. Although they may help in sentiment classification, but we have kept the task simple.
- All the html tags are removed.
- All the text is converted to lower case.

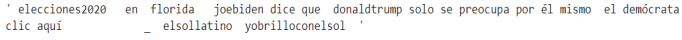

Fig. 4. Clean Tweet

E. Sentiment analysis of tweets

To analyse tweets and derive correlations between different components of the dataset, we carried out sentiment based analysis instead of simple analysis. To perform this task, we imported the sentiment analyzer built in library, created sentiments and analysed tweets based on the results.

IV. Correlation Between Attributes

We plotted the correlation between different columns of the dataset by making a correlation matrix. The matrix shows that a visible correlation is between all location specific columns. It can also be seen that the number of likes, followers and retweets are correlated with each other as well. All attributes that identify a user (user screen name, id etc) are also correlated with each other. The matrix can be seen in Fig 5.

V. Exploratory Data Analysis (EDA)

In this section, the exploratory data analysis of the data is discussed in details. We have analyzed the data based on inactive user, candidates, locations, sentiment source etc.

A. Analysis of tweets of In-active users

Before looking at the tweets with respect to other attribute, we analyze the data on the basis of inactive users to see if the in-active users are among the top people to tweet. First we see number of active and inactive user. The table below shows the statistics.

The table shows that there are only four inactive users. Further exploring the number of tweets these 4 users made, the data shows that there were only 21 tweets by them where as total number of tweets by active users are 1747784. Further exploring number of followers, likes and retweet counts, the

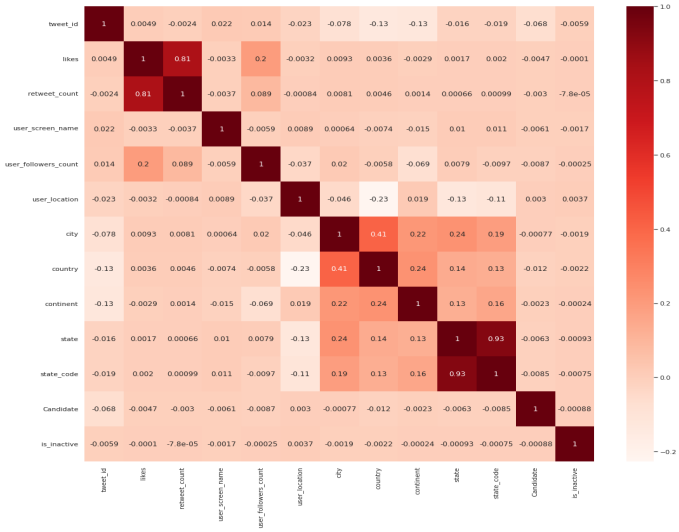

Fig. 5. Correlation Matrix between attributes

TABLE I
NUMBER OF INACTIVE AND ACTIVE USER

Type	Number of user
Active	483203
In- Active	4

results do not show any significant importance of tweets made by these user.

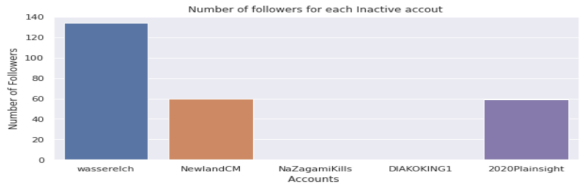

Fig. 6. Followers of Inactive users

B. *Candidate wise analysis of tweets*

The detailed candidate wise analysis of tweets is discussed in this section. There are tweets of two candidates; Trump and Biden. The distribution of tweets for both candidate is shown in the figure 7. The figure shows that the number of tweets

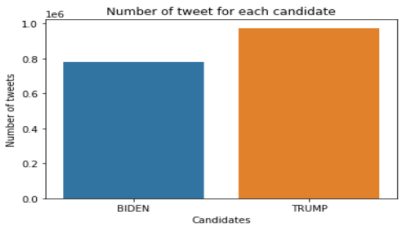

Fig. 7. Number of tweets per Candidate

for Trump is higher then that of Biden. The time line analysis of the candidate is also done (Fig 8).

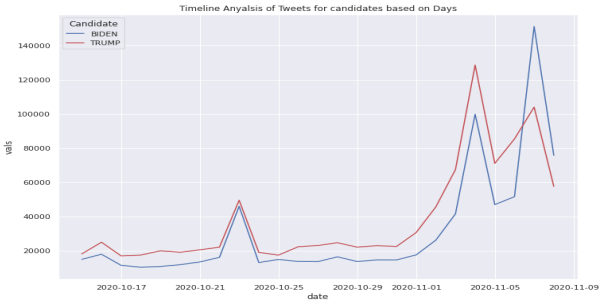

Fig. 8. Timeline Analysis on candidate

The timeline shows that number of tweets of trump are always higher then Biden expect between Nov 7, 2020 to Nov 11, 2020 where a peak can be seen for Biden. The candidate wise analysis is also done in combination with other attributes.

C. *Sentiment Wise Analysis of Tweets*

Besides observing number of tweets per candidate, it is important to see if the popularity of the candidates is negative in most of the tweets. This might help us know the real popularity of a candidate. After analyzing candidates in isolation we have also analyzed them on the basis of sentiments of tweets. The initial analysis of sentiment is as shown in the figure 9.

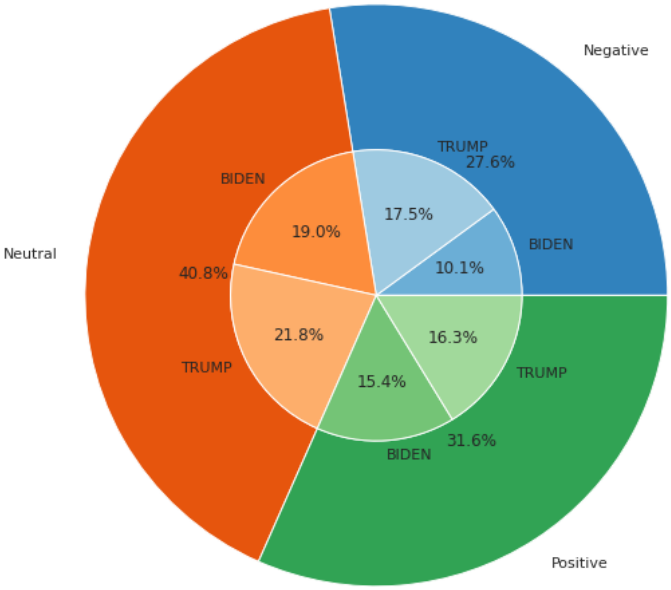

Fig. 9. Number of Tweets form each source of each Candidate

The figure shows that there are 40% of neutral tweets, 31% of positive tweets and 28% of negative tweets. The figure shows that for each sentiment of tweet Trump has more tweets then Biden. But the number of negative tweets of Trump are much higher than Biden.

The timeline based analysis shows that the negative tweets of Biden are lowest through out the timeline. The neutral tweets of Trump dominated overall but after 6 Nov the Neutral tweets for Biden shows a spike.

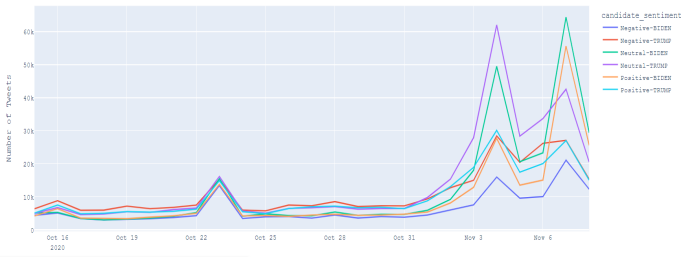

Fig. 10. Sentiment wise distribution of Data

D. *Sources of Tweets*

The sources from where the tweet are collected are analyzed in detail. Based on the number of tweets per source, the top most sources where the tweets are posted are

- Twitter Web App
- Twitter for iPhone
- Twitter for Android
- Twitter for iPad
- TweetDeck
- Instagram

The tweets generated by each source are displayed in the figure 11:

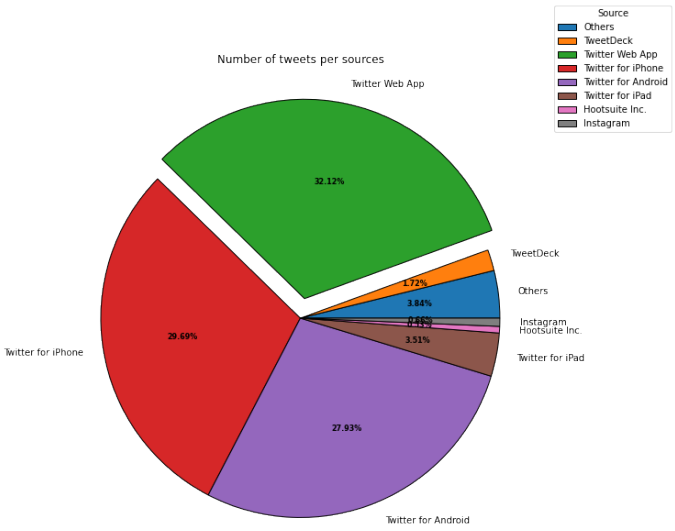

Fig. 11. Tweets form each source

The figure shows that most tweets are posted on Twitter web app. The others in the sources is used to display percentage of sources which have less then 0.5% of tweets. Further exploring the number of tweets of each candidates (Fig 12) in top 6 show that the number of tweets for trump from each source are higher then that of Biden.

The sources was also analyzed by sentiments as well. The analysis shows that all the sources have most tweets as neutral, followed by positive tweets and then negative tweets. The positive tweets on 'Twitter Web Page' and 'Twitter for Iphone' are more then 'Twitter for Android' which is the top source for posting tweets.

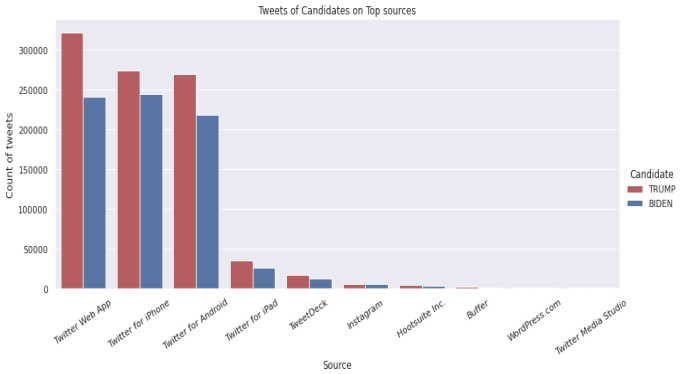

Fig. 12. Number of Tweets form each source of each Candidate

Further we also visualized the sentiment wise tweets on each source based on candidates as shown in the Fig 14. The plots shows that negative tweets for Trump on top sources are more then that of Bedien. Also, for both the Candidates, the number of positive and neutral tweets are higher then negative tweets, which was the trend observed in the previous plots as well.

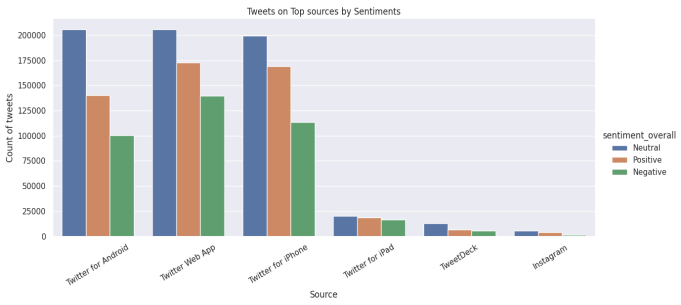

Fig. 13. Number of Tweets form each source by sentiments

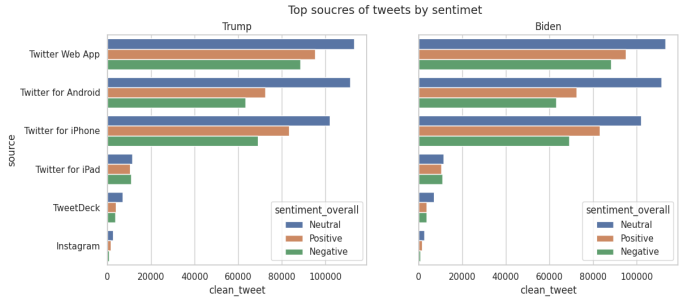

Fig. 14. Sentiment wise tweets on each Source based on Candidates

E. *Location wise analysis of tweets*

In order to do the location-wise analysis of tweets, we first plotted the number of tweets by each country and discovered that the highest number of tweets was from the USA which is understandable since the elections are in the USA. We

then counted the number of tweets for each candidate based on the countries and plotted this information as well. We also performed sentiment wised analysis of tweets based on location. We then repeat this process for the cities and the states. In both the analysis of cities and states, Trump had a higher number of tweets. The difference was especially large in states like California, England, and Florida.

*1) **Country wise analysis of tweets**:* Fig 15 shows country wise analysis of tweets for each candidate. As the data has many rows with missing data about country, that is why 'Geo Data N/A' appeared as a column with most tweets for both Candidate. In almost all the countries number of tweets for Trump is more then that of Beiden except India.

Further sentiment wise analysis on country for each candidate

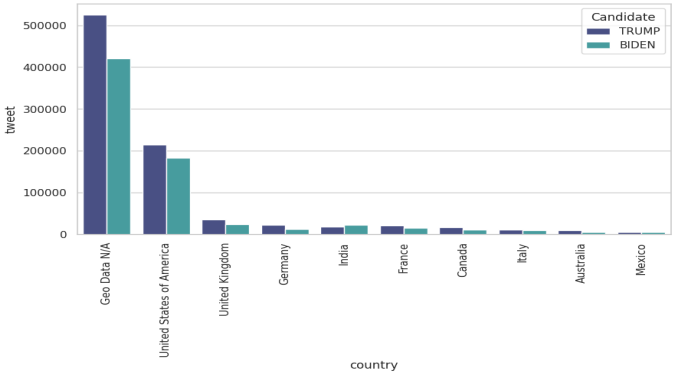

Fig. 15. Country wise visualization of tweet count

is also performed as shown in Fig 16. The plots for both candidates shows that USA and UK have more positive tweets then other two categories. Where as in France, for both the candidates neutral tweets were more. In Germany, neutral and negative tweets for Trump are available in same ratio where as Biden shows popularity in India as well.

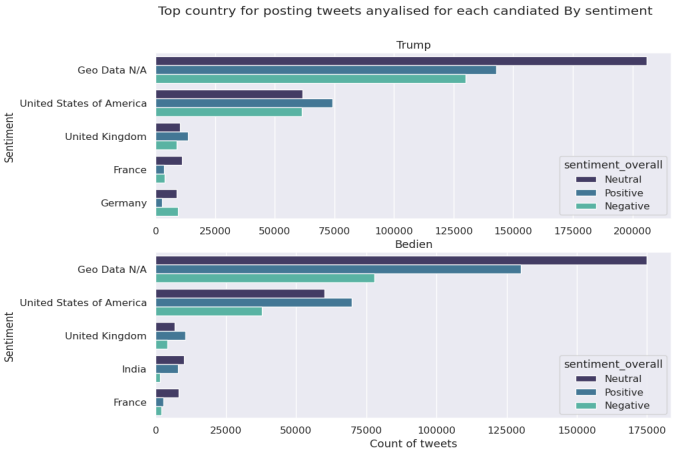

Fig. 16. Country wise visualization of tweet count

*2) **City wise analysis of tweets**:* The city wise analysis of tweets for each candidate is also performed. Fig 17 displays plot of city wise analysis of plot. As the data has many rows with missing data about city, that is why 'Geo Data N/A' appeared as a column with most tweets for both Candidate. In almost all the countries number of tweets for Trump is more then that of Beiden except New Delhi, a city in India.

Further sentiment wise analysis on country for each candidate

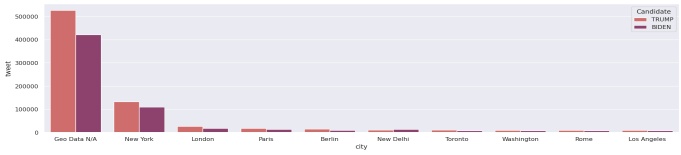

Fig. 17. City wise visualization of tweet count

is also performed as shown in Fig 18. The plots for both candidates shows that New York and London have more positive tweets then other two categories. Where as in Paris, for both the candidates neutral tweets were more. In Berlin, neutral and negative tweets for Trump are available in same ratio where as Biden shows popularity in New Dehli in neutral and Positive in as well.

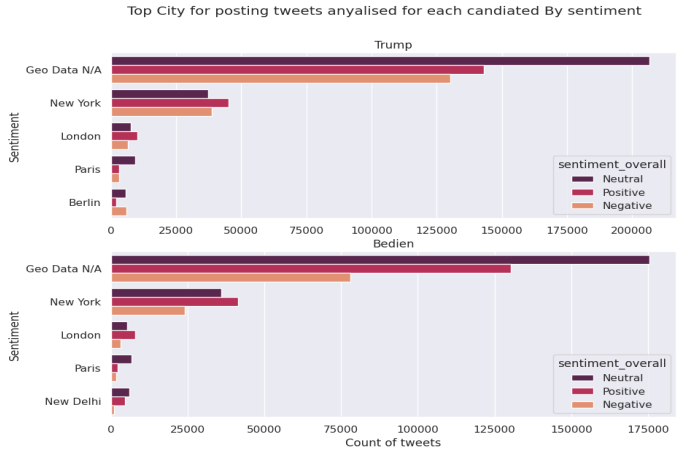

Fig. 18. City wise visualization of tweet count by Sentiment

*3) **State wise analysis of tweets**:* Further location wise analysis of tweets is also performed for states. Fig 19 shows of state wise analysis of candidate. As the data has many rows with missing data about city, that is why 'Geo Data N/A' appeared as a column with most tweets for both Candidate. In almost all the states number of tweets for Trump is more then that of Beiden. Most tweets are posted in California.

Further sentiment wise analysis on state for each candidate

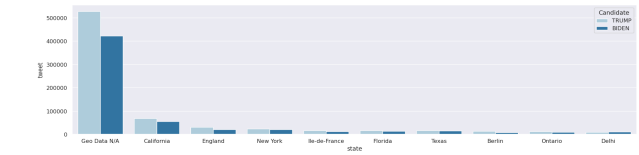

Fig. 19. State wise visualization of tweet count

is also performed as shown in Fig 21. The plots for both

candidates shows that California, England and New york have more positive tweets then other two categories. In New york and England have more negative tweets for Trump then that of Bedien. This is the on the bases of ratio of tweets of each sentiment.

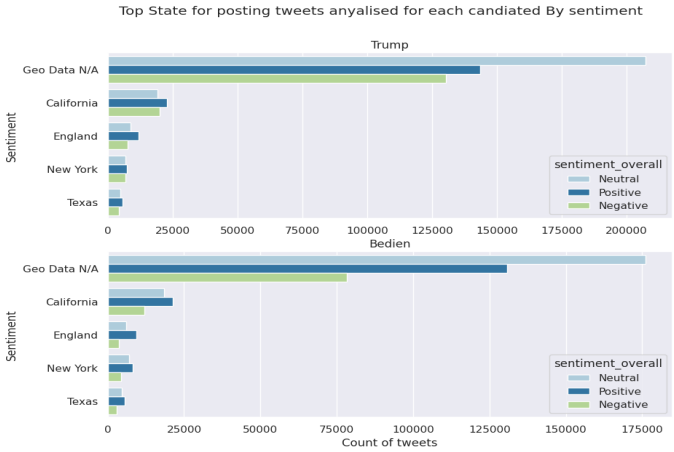

Fig. 20. State wise visualization of tweet count

F. *Analysis of data by Hashtag and Top words*

As the content of the tweet, i.e hastags and words of tweets plays a major role in deciding sentiment as well as candidate of tweets, so this was explored in details as well.

*1) **Visualizing Most frequent Word used in tweets**:* The words used in tweets are also analyzed in detail. Fig 21 shows word-cloud for tweets. Showing the most common word is Trump followed by Joe Biden which are two candiate of elections. Surprisingly Dice is among the top words.

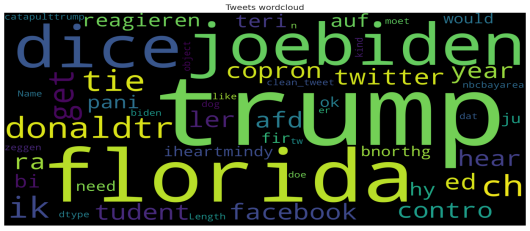

Fig. 21. Word cloud for all tweets

Further the word cloud of each candidates were also plotted as shown in Fig 22 and Fig 23. The Word-cloud of Trump shows

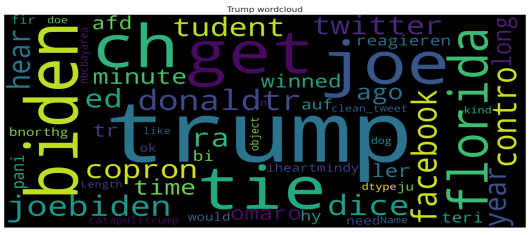

Fig. 22. Word cloud for all tweets

that besides the name of both candidates, the name of states and time related words are most common. The Word-cloud of

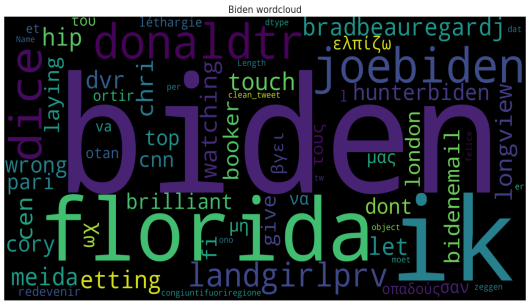

Fig. 23. Word cloud for all tweets

Biden does not show Trump as the top word instead donaldtr is common word which shows the popularity of Trump among people who tweets for Biden.

*2) **Visualizing top Hashtags used in tweets**:* The data was explored in detail to see the top hashtags used in tweet. The Most common hashtags used are plotted in figure 24. The plots shows that most common hashtags are name of candidates. Further, candidate wise analysis of hash tags is also visualized

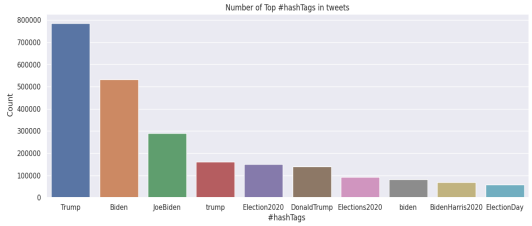

Fig. 24. Top Hash Tags

as shown in Fig-25. The plot shows that hashtags are almost similar with a difference in frequency of hash tags.

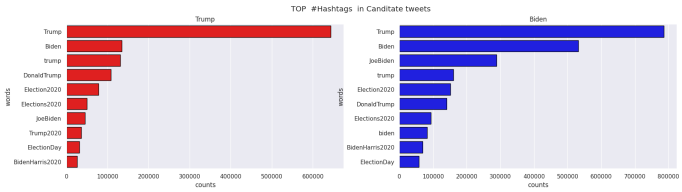

Fig. 25. Candidate wise analysis of hash tags

G. *Using Maps to Visualize Tweet Counts and Likes*

To visually represent the count of tweets in the dataset on the basis of the candidate column, a geo-plot was constructed. Each tweet was first separated into categories of candidates - Trump and Biden which was then plotted on the geo-map using the longitude and latitude columns of the dataset. The longitude and latitude corresponds to the location from where the tweet was posted. The figure shows that the number of tweets made for Trump and Biden were almost same.

To understand the distribution of likes on the basis of the

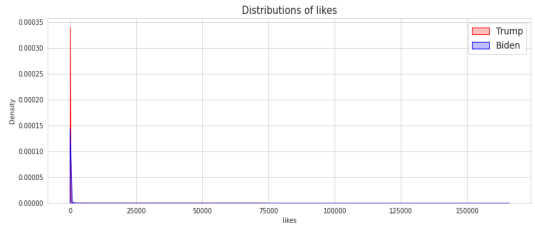

Fig. 26. Visualization of likes

candidate column, the data was again separated into two separate datasets - one for Trump and the other for Biden. The 'likes' column out of these two datasets was plotted on a spectral map to analyse the popularity of the candidates based on the number of likes. Fig 18 shows that the number of likes for Trump were significantly more than the likes for Biden.

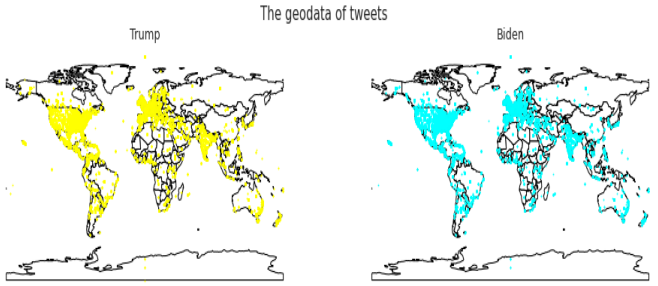

Fig. 27. Visualization of tweet count

H. User Based Analysis of Tweet for based on Number of tweets, Like, followers count and Retweets

For this section we start by looking at the statatics of like, retweets In this section for each Candidate we have made analysis based on number of tweets, likes and retweets. First we visualize top user by number of tweets.

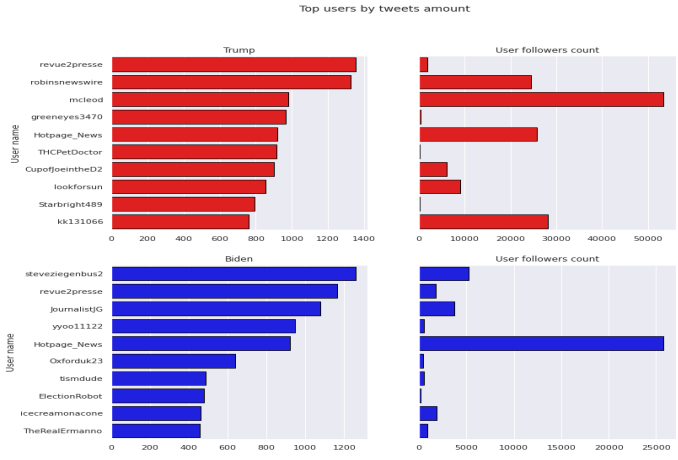

Fig. 28. Visualization of followers of top user on tweet count

This shows that the user with most tweets does not have the highest number of followers. The user who tweets about Trump have higher followers count.
Secondly, we visualize top user by retweets of tweets.

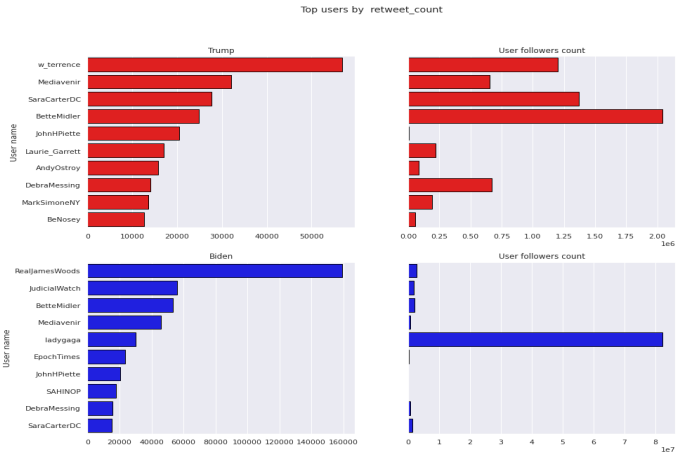

Fig. 29. Visualization of of followers of top user on likes

The figure shows positive correlation between retweet count and user followers for trump. The same trend cannot be seen for Biden. Thirdly, we visualize top user by likes of tweets. The user with most likes on tweets does not have highest number of followers. The plots of Trump shows more positive correlation then Biden.The plots of Biden are similar in all the cases.

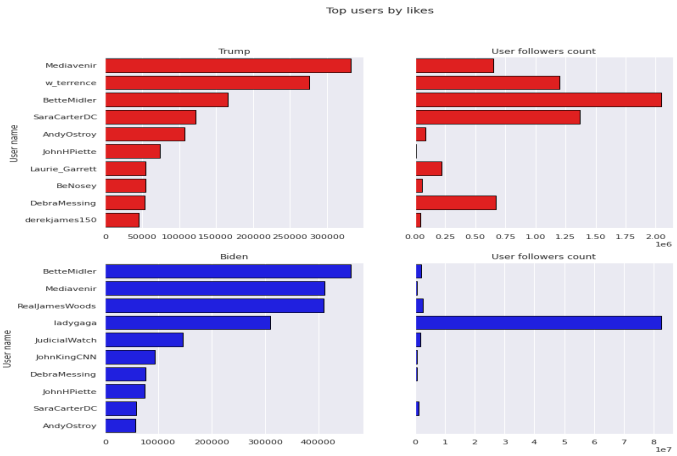

Fig. 30. Visualization of tweet count

VI. Cluster Analysis

The process of clustering was performed on the US Election Tweets 2020 dataset with the help of two different clustering algorithms. These two algorithms were chosen to balance out the limitations of one algorithm with the use of the other algorithm.

A. Algorithm 1: K-Means Clustering

K-means clustering is a method of unsupervised cluster analysis where the dataset is divided into a pre-defined number

of clusters which each object belonging to the cluster with the closest mean. We are using K-means clustering because given the large dataset that we're given, it is computationally efficient to use K-means clustering since it can handle large datasets, with many variables and many clusters.

Procedure

1. We first select the number of clusters - k.
2. We choose k random data points from the dataset as the initial centroids for each cluster
3. Assign the data points to clusters. Each data point is assigned to cluster with the closest centroid in distance.
4. Recalculate the centroids from each cluster by calculating the mean value.
5. Repeat until the algorithm converges and the clusters no longer change significantly.

1) ***K-Means Clustering on Tweets****:* We applied k-means clustering on initial clusters ranging from k = 1 to k = 25 and used the columns 'Tweets'. The quality of each cluster was assessed using WCSS (Within-Cluster Sum of Squares), which measures the compactness of each cluster. A lower value of WCSS indicates that points within a cluster are tightly packed around the centroid, suggesting a more meaningful and well-defined cluster. We used the Elbow Method to determine the optimal number of clusters using the WCSS value. With the column 'tweets', this optimal value was chosen to be k = 11. We then used the WordCloud library to visualise the clusters

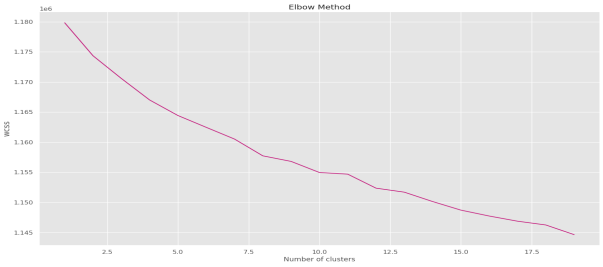

Fig. 31. Result of Elbow Method on Tweets

using the optimal value of k.

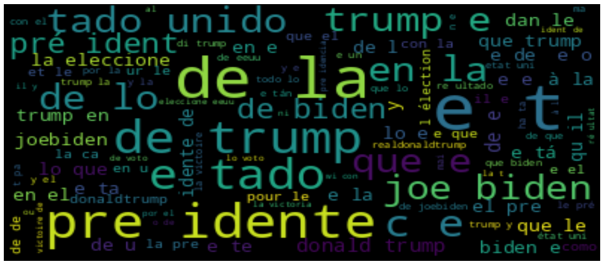

Fig. 32. An example cluster from running K-Means visualised using WordCloud

2) ***K-Means Clustering on State and Likes****:* We applied k-means clustering on initial clusters ranging from k = 1 to k = 11 and used the columns 'State' and 'Likes'. The quality of each cluster was assessed using WCSS (Within-Cluster Sum of Squares). We used the Elbow Method to determine the optimal number of clusters using the WCSS value. With the columns chosen, this optimal value was chosen to be k=4.

The clusters formed using the optimal value of k were

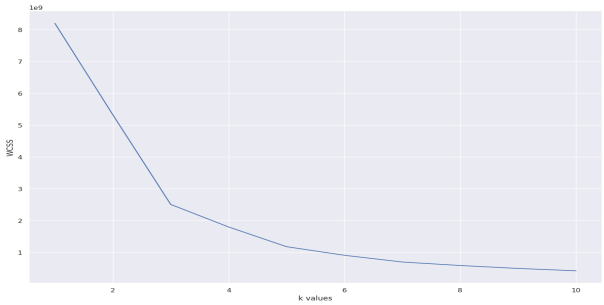

Fig. 33. Result of Elbow Method on State and Likes

visualised using a plot, with the centroid of each cluster also displayed.

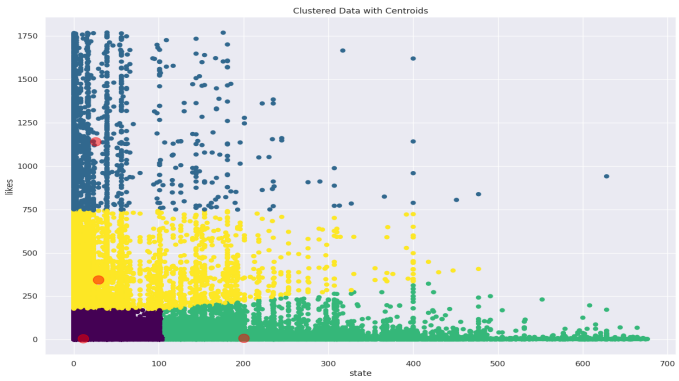

Fig. 34. Clustered Data with State and Likes

3) ***K-Means Clustering on City and Source****:* We applied k-means clustering on initial clusters ranging from k = 1 to k = 11 and used the columns 'City' and 'Source'. The quality of each cluster was assessed using WCSS (Within-Cluster Sum of Squares). We used the Elbow Method to determine the optimal number of clusters using the WCSS value. With the columns chosen, this optimal value was chosen to be k = 3.

The clusters formed using the optimal value of k were

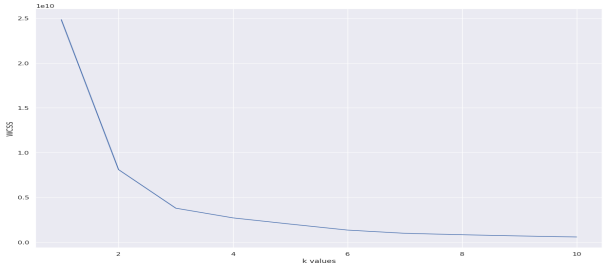

Fig. 35. Result of Elbow Method on City and Source

visualised using a plot, with the centroid of each cluster also displayed.

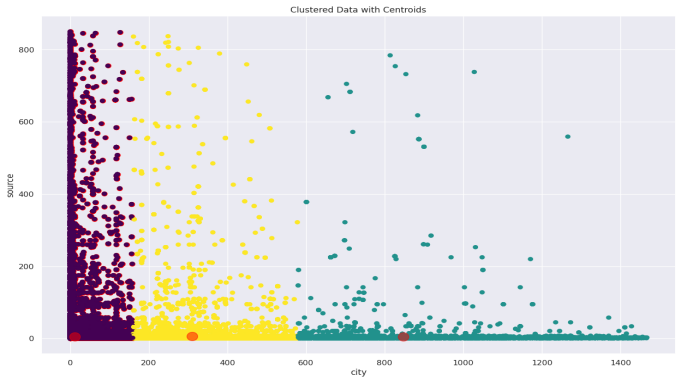

Fig. 36. Clustered Data with City and Source

*4) **K-Means Clustering on Country and Source**:* We applied k-means clustering on initial clusters ranging from k = 1 to k = 11 and used the columns 'Country' and 'Source'. The quality of each cluster was assessed using WCSS (Within-Cluster Sum of Squares). We used the Elbow Method to determine the optimal number of clusters using the WCSS value. With the columns chosen, this optimal value was chosen to be k = 4.

The clusters formed using the optimal value of k were

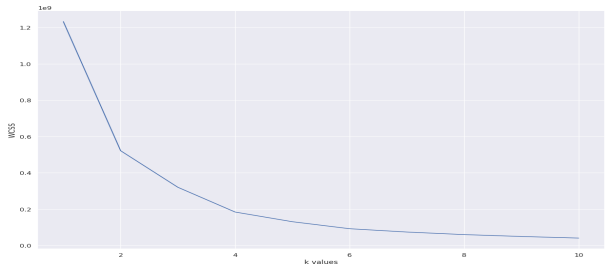

Fig. 37. Result of Elbow Method on Country and Source

visualised using a plot, with the centroid of each cluster also displayed.

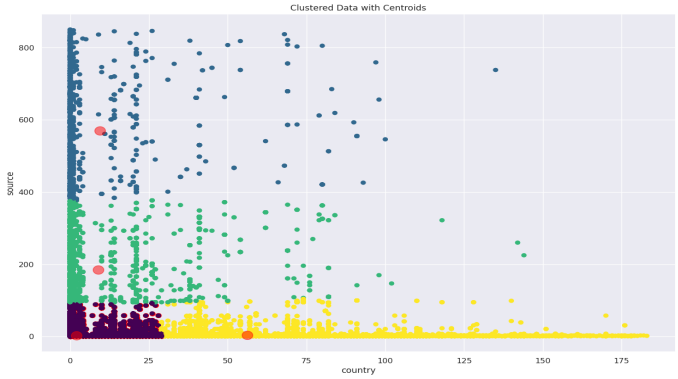

Fig. 38. Clustered Data with Country and Source

*5) **K-Means Clustering on Longitude and Latitude**:* We applied k-means clustering on initial clusters ranging from k = 1 to k = 15 and used the columns 'Long' and 'Latitude'. As there were many nulls in these columns so they were removed and only 54% of the data remained. The longitude latitude plot is shown in the fig below which was considered for clustering. The quality of each cluster was assessed using

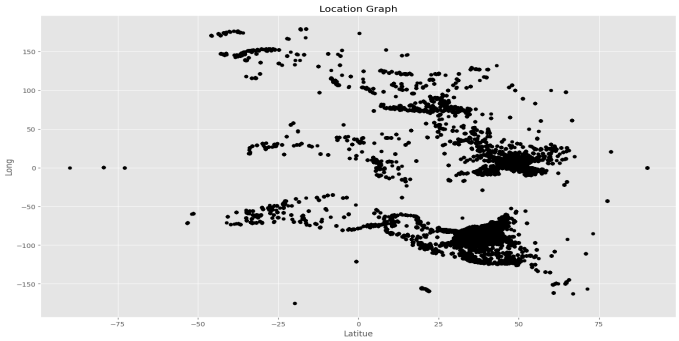

Fig. 39. Location Wise plot of Data

WCSS (Within-Cluster Sum of Squares). We used the Elbow Method to determine the optimal number of clusters using the WCSS value. With the columns chosen, this optimal value was chosen to be k = 5.

The cluster formed by k-mean algorithm is shown in the fig

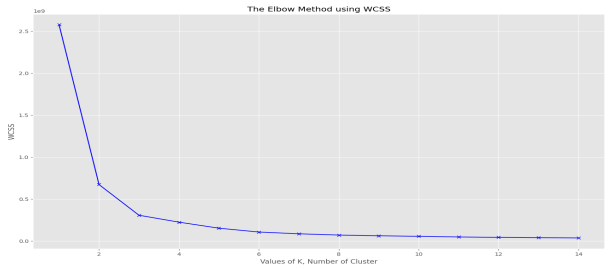

Fig. 40. Result of Elbow Method on Longitude and Latitude

41. The cluster seems to be formed really well.

These were visualized with continent data and similar cluster

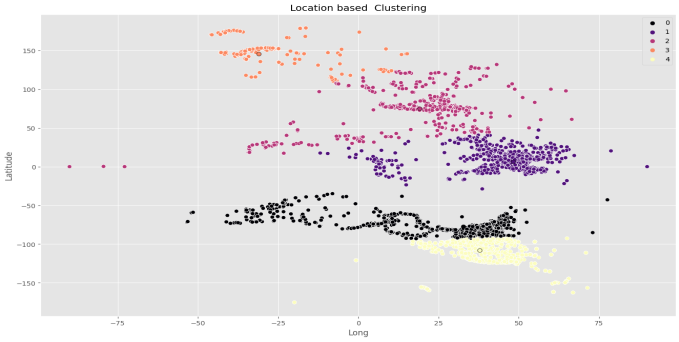

Fig. 41. Clustered Data with longitude and latitude

were scene there. The algorithm has formed the cluster with Continent quite well.

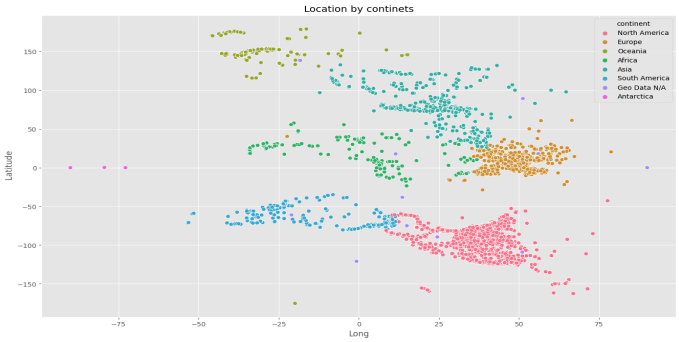

Fig. 42. Visualizing longitude and latitude by continent

*6) **K-Means Clustering on Longitude ,Latitude and most Popular hashtags**:* We further analyzed the location by the most popular hastags. The grahp is shown fig 43. The most popular hastags are represented by frequency. The quality of

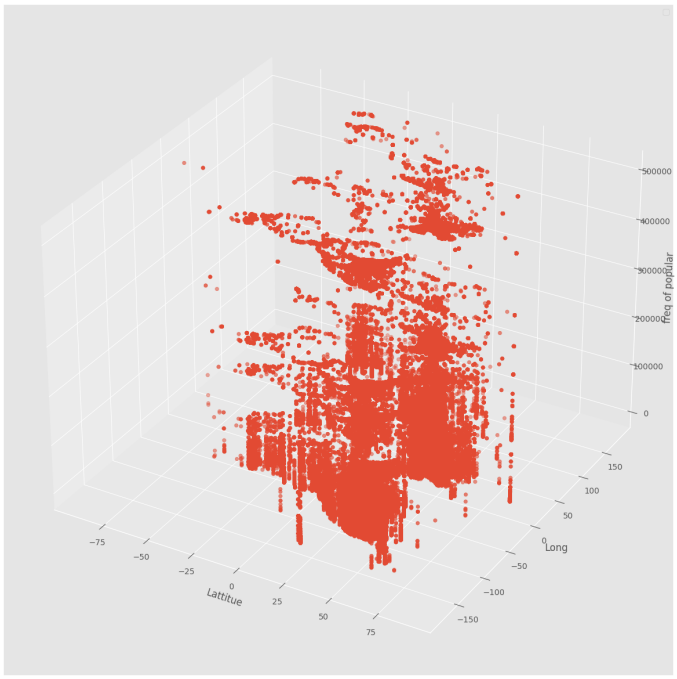

Fig. 43. Location Wise plot of Data by popular tweets

each cluster was assessed using WCSS (Within-Cluster Sum of Squares). We used the Elbow Method to determine the optimal number of clusters using the WCSS value. With the columns chosen, this optimal value was chosen to be k = 3.

The cluster formed by k-mean algorithm is shown in the fig 45. The cluster seems to be formed really well.

B. Algorithm 2: DB-Scan Clustering

In the K-means algorithm, the clustering was done based on brute force (trying out multiple number of clusters until one which gave an elbow point was discovered). This introduces uncertainty in the actual number of clusters that a

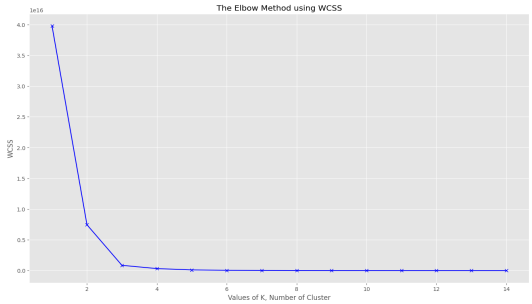

Fig. 44. Result of Elbow Method on Longitude, Latitude and popular hastags

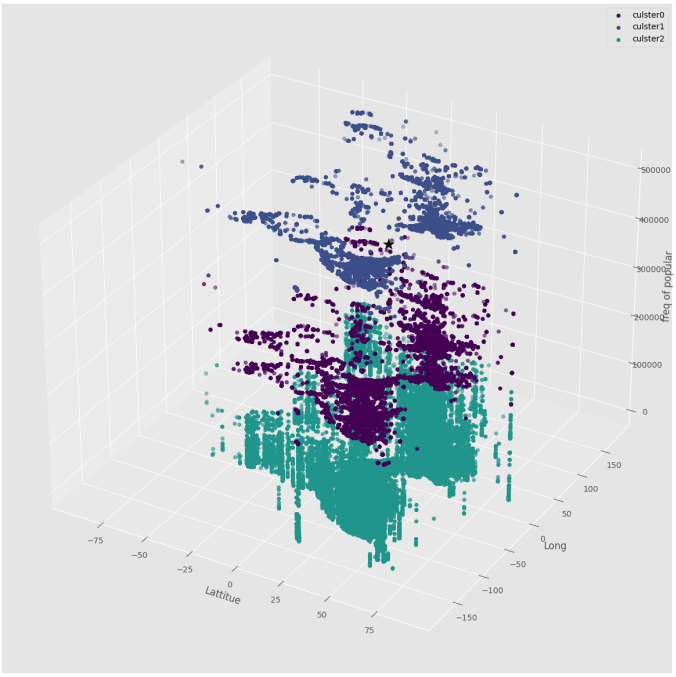

Fig. 45. Clustered Data with longitude and latitude

particular data can be divided into since the clusters depend on the elbow point. However, in DB-scan, there is no need to specify the number of clusters in the beginning and the data is clustered based on the actual number of clustered that exist in the data. Furthermore, k-means is susceptible to erroneous results due to it being sensitive to outliers, db-scan, on the other hand, is not affected by outliers and can easily ignore them from amongst the data being considered.

Procedure

1. This algorithm requires specifying the input parameters eps and minsamples. These basically define the region around a point that would be considered a cluster.
2. Select a random point from the dataset.
3. If the randomly chosen point has at least minsample neighboring points within distance eps, then create a new cluster and add the point to it. All the neighboring points within distance eps of the point are added to the cluster as

well.
4. Check if other neighboring points of the point in consideration have atleast minsample points within distance eps, then add them to the cluster. Repeat this step until no more points can be added to the current cluster.
5. If the point cannot be added to the cluster, mark it as an outlier.
6. Repeat the above steps until all the dataset has been traversed.
7. As a result, the clusters that form are the ones that are connected to each other or have a close association amongst each other.

1) ***DB-Scan clustering of country and likes****:* We applied DB-Scan clustering on the country and likes columns of the dataset to find out the occurrences where they exist together. The algorithm was performed with a min sample value of 10, eps as 30 and distance metric as euclidean. The algorithm clustered the data into a few differently coloured segments based on the provided variables. Total 6 clusters are formed as plotted in the figure below. The value of eps and min sample was chosen after visualizing clusters as well as the number of outliers for different values and best fitted values are use for final clustering.

Furthermore, a scatter plot was potted of the clustered data

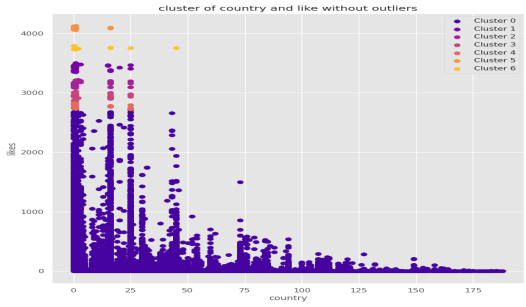

Fig. 46. DB-Scan clustering of country and likes

which showed that there were around 364 outliers in the data. The cluster are quite spread out thereby signifying that the country of origin and number of likes of a tweet are spread out with respect to each other.

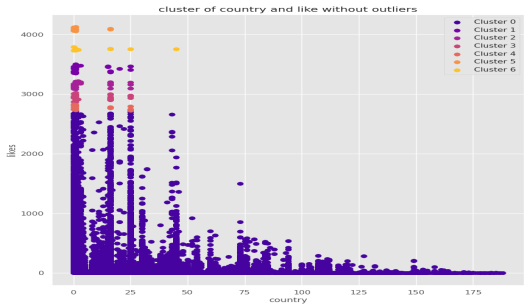

Fig. 47. DB-Scan clustering of country and likes without outliers

2) ***DB-Scan clustering of likes and user followers count****:* We applied DB-Scan clustering on the likes and user followers count columns of the dataset to find out the occurrences where they exist together. The algorithm was performed with a min sample value of 500, eps as 700 and distance metric as euclidean. The large values were used after visualizing the columns under consideration for different values of parameter. Small values were forming large number of cluster and point which could be in same clusters were put in different clusters. The algorithm clustered the data into a few coloured segments based on the provided variables. The number of clusters formed are 6.

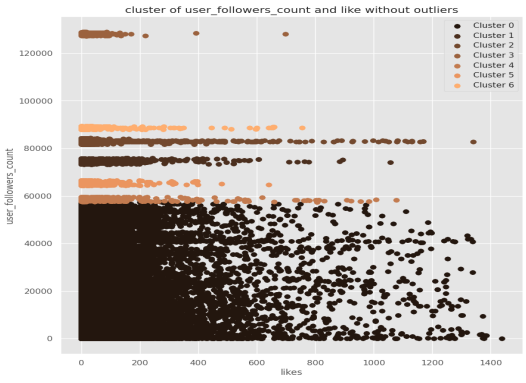

Fig. 48. DB-Scan clustering of likes and user followers count

Furthermore, a scatter plot was potted of the clustered data which showed that there were not many clusters in the data, thereby signifying that the likes on a tweet and number of number of followers of the publishing account usually occur together and are correlated. There were around 39030 outliers which are plotted with clusters in fig 49.

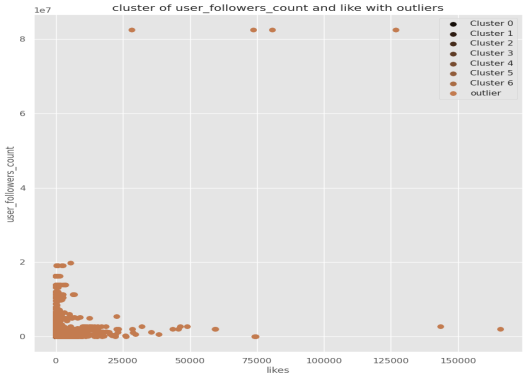

Fig. 49. DB-Scan clustering of likes and user followers count with outliers

3) ***DB-Scan clustering of likes and retweet count****:* We applied DB-Scan clustering on the likes and user followers count columns of the dataset to find out the occurrences where they exist together. The algorithm was performed with a min sample value of 10, eps as 20 and distance metric as euclidean. This was chosen after visualizing cluster for different values of parameter. The chosen parameters seem to perform best. It

formed around 13 clusters. The algorithm clustered the data into a 13 coloured segments based on the provided variables. Furthermore, a scatter plot was potted of the clustered data

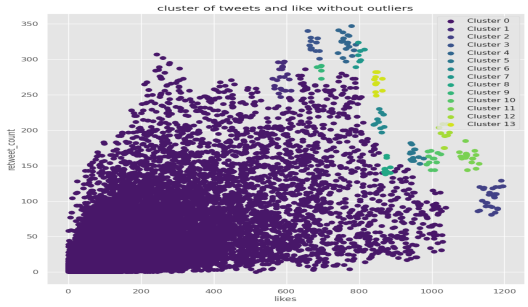

Fig. 50. DB-Scan clustering of likes and retweet count

which showed that there were not many clusters in the data, thereby signifying that the likes on a tweet and number of number of followers of the publishing account usually occur together. There are around 2009 outliers, which are plotted in fig 51.

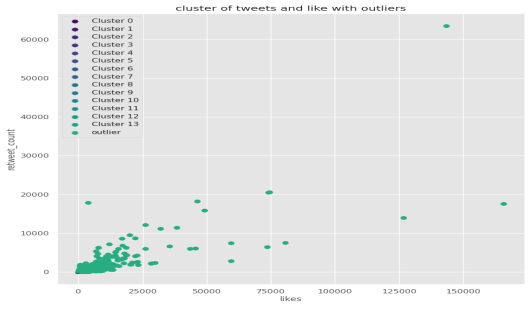

Fig. 51. DB-Scan clustering of likes and retweet count

4) ***DB-Scan clustering of retweet and user followers count****:* We applied DB-Scan clustering on the number of retweets and user followers count columns of the dataset to find out the occurrences where they exist together. The algorithm was performed with a min sample value of 570, eps as 240 and distance metric as euclidean. These values were selected by visualizing the clusters for different values of min-samples and eps. Large values are kept as most of the

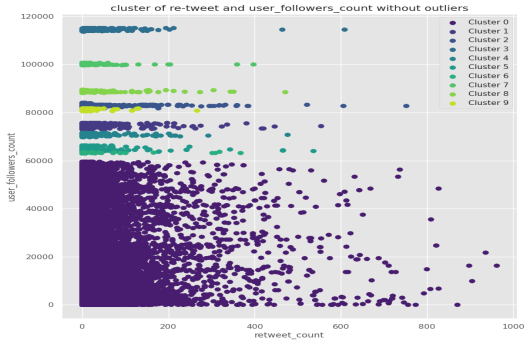

Fig. 52. DB-Scan clustering of retweet and user followers count

same points were located with a small distance and have same trend.The algorithm clustered the data into a few coloured segments based on the provided variables. For the selected parameters 10 cluster were formed as shown in the plot in fig 52.

Furthermore, a scatter plot was potted of the clustered data which showed that there were not many clusters in the data, thereby signifying that the number of retweets on a tweet and number of number of followers of the publishing account usually occur together. However, since there are alot of outliers, around 31435, this was not an accurate depiction of the relation. The plot with outlier is also plotted for visualization.

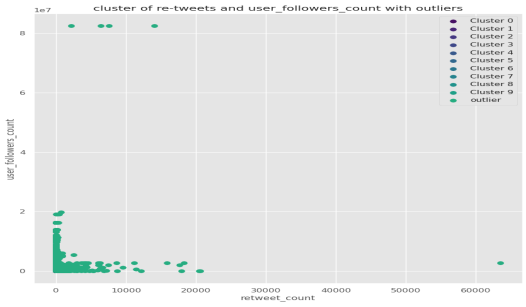

Fig. 53. DB-Scan clustering of retweet and user followers count

5) ***DB-Scan clustering of country and city****:* We applied DB-Scan clustering on the country and city columns of the dataset to find out the occurrences where they exist together. The algorithm was performed with a min sample value of 7, eps as 15 and distance metric as euclidean.These values were selected by visualizing the clusters for different values of min-samples and eps. For larger values of eps and well as min-samples, the number of clusters formed are very few and does not have natural tendency to form cluster. For smaller values a lot of cluster were being formed. For the selected parameters 10 cluster were formed as shown in the fig 54.

The algorithm was also able to detect outliers which were

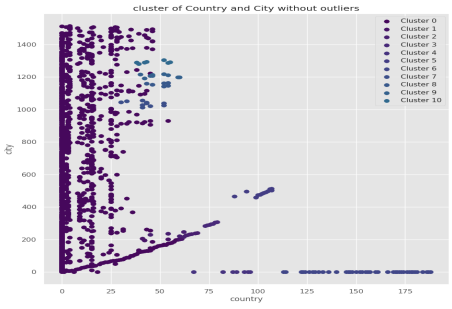

Fig. 54. DB-Scan clustering of country and city

around 271. The scatter plot with outliers is also plotted in the fig below. The plots signifies that the country and city of the publishing account are very variable with respect to each other. Since there are less outliers, this was an accurate depiction of the relation.

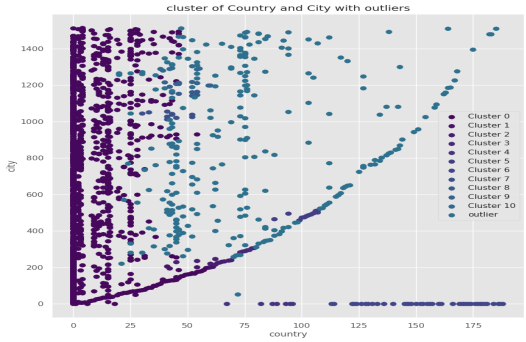

Fig. 55. DB-Scan clustering of country and city with outliers

6) ***DB-Scan clustering of city and state****:* We applied DB-Scan clustering on the city and state columns of the dataset to find out the occurrences where they exist together. The algorithm was performed with a min sample value of 17, eps as 30 after visualizing data and clusters for different value of parameters and distance metric as euclidean .The algorithm clustered the data into 7 coloured segments based on the

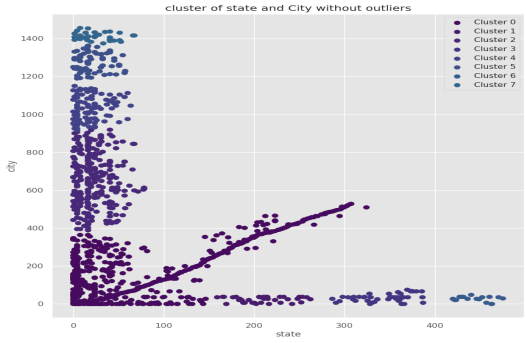

Fig. 56. DB-Scan clustering of city and state

provided variables.
Furthermore, a scatter plot was potted of the clustered data which showed that there were 7 clusters in the data, thereby signifying that the city and country of the publishing account are very variable with respect to each other. Since there are less outliers, around 943, this was an accurate depiction of the relation. The scatter plot is shown in fig 57.

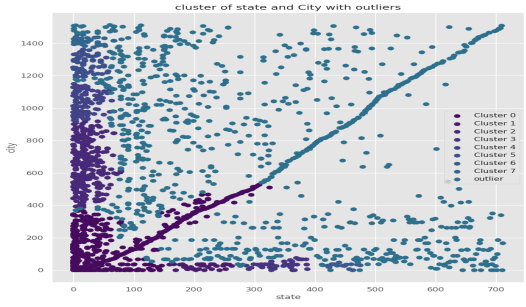

Fig. 57. DB-Scan clustering of city and state with outliers

7) ***DB-Scan clustering of likes and Candidate****:* We applied DB-Scan clustering on the likes and Candidate columns of the dataset to find out the occurrences where they exist together. The algorithm was performed with a min sample value of 14, eps as 25 and distance metric as euclidean, selected after careful analysis. The algorithm clustered the data into 6 coloured segments based on the provided variables. The plots of clusters is shown in fig 58.

Furthermore, a scatter plot was potted of the clustered data

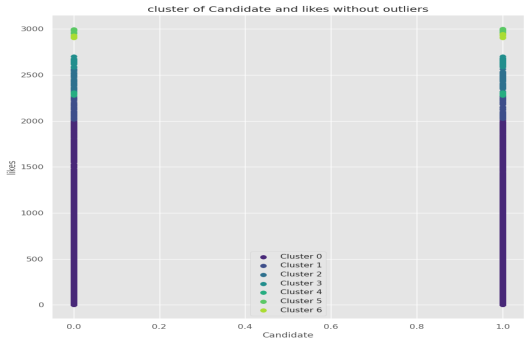

Fig. 58. DB-Scan clustering of likes and Candidate

which showed that there were 6 of clusters in the data, thereby signifying that the number of likes and Candidate are dependent on each other. Since there are 474 outliers, this was an accurate depiction of the relation.

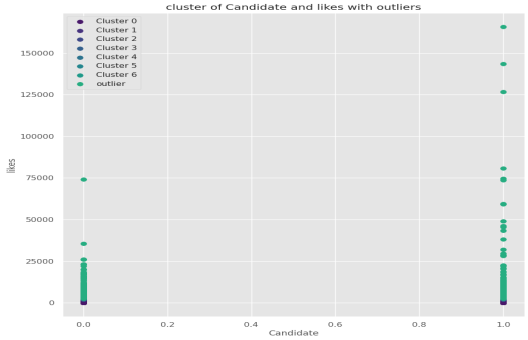

Fig. 59. DB-Scan clustering of likes and Candidate with outliers

VII. Frequent Pattern Mining

For this project as the data was text data, we tried to extract useful and frequent patterns on tweets and hashtags on the basis of various attributes of the data. We also deeply analysed usage of sources of tweets with respect to location (country) and days on which tweets were posted.
Before finding frequent Patterns, we filtered out in-frequent values as they won't form any frequent pattern. The criteria of selecting frequent values in all the attributes is as follows:

1) Country: There are 184 unique countries out of which we only use 8 countries i.e 'Geo Data N/A', 'United States of America', 'United Kingdom','Germany', 'Canada', 'France', 'India' and 'Italy' as rest have less then 0.01 percent of tweets.

2) Source: There are total 851 unique sources out of which only top 20 were selected for analysis. These sources have at least 500 tweets.

3) Hashtags: There are total 249365 unique hashtags out of top 200 were selected for analysis which occurred around 1800 times in all the data set provided of 1,747,805 rows.

A. Algorithm selection

For some attributes, we found frequent patterns by both FP Growth and Apriori Algorithms. As FP Growth seemed to perform faster, so for the rest of the attributes and analysis, FP-growth is used.

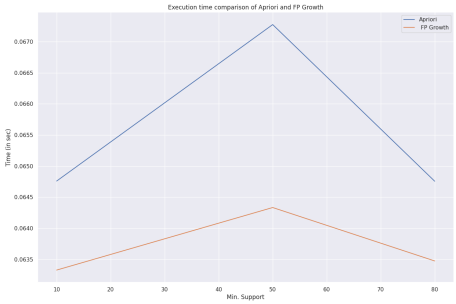

Fig. 60. Run-time of FP-growth and Approri

B. Frequent Pattern On Country and Source

To find frequent patterns of country and source in the data, for each source an array of country is created where the source has been used at lest 500 times to post tweets. This resulted in a 2d array with 20 rows and 8 columns.Each row represents a source and each column represents a country. Transaction encoding, is used to transform data in a form useable by the algorithm. Each cell was then marked as true or false depending on if the country had posted the tweet or not. As we have already considered frequent source and countries, so min-support was kept high (on 80% and 85%). The length was not specified to find all frequent patterns. For 80% as min-support we found 63 frequent patterns which was reduce to 15 when 85% was used as min-support.

The frequent pattern shows that even among top 8 countries only 4 countries have tweets on top 20 sources with support above 85% . These is also shown in fig 50. The similar pattren was seen with minimum support of 80 with some additional countries.

1) Association rules: There were 50 rules form and all rules have above 90% confidence. This shows that if a source is use in one of the 4 top countries it is most likely to be used in the other countries. The rules with 100% confidence are as shown in fig 51.

C. Frequent Pattern On Infrequent Countries and Days

While analysing days with countries, we found that almost all of the countries have posted tweets everyday expect few in-frequent countries. In this part we have analyzed those countries by days. For this part we have considered countries

	index	support	itemsets	len
0	0	0.95	(United States of America)	1
1	1	0.90	(Geo Data N/A)	1
2	2	0.85	(United Kingdom)	1
3	3	0.85	(India)	1
4	8	0.85	(United States of America, Geo Data N/A)	2
5	9	0.85	(United Kingdom, Geo Data N/A)	2
6	10	0.85	(United States of America, United Kingdom)	2
7	11	0.85	(United States of America, United Kingdom, Geo...	3
8	12	0.85	(United Kingdom, India)	2
9	13	0.85	(India, Geo Data N/A)	2
10	14	0.85	(United States of America, India)	2
11	15	0.85	(United Kingdom, India, Geo Data N/A)	3
12	16	0.85	(United States of America, United Kingdom, India)	3
13	17	0.85	(United States of America, India, Geo Data N/A)	3
14	18	0.85	(United States of America, United Kingdom, Ind...	4

Fig. 61. Frequent Posting countries by top sources with minimum support of 85%

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
2	(United Kingdom)	(Geo Data N/A)	0.85	0.90	0.85	1.0	1.111111
5	(United Kingdom)	(United States of America)	0.85	0.95	0.85	1.0	1.052632
6	(United States of America, United Kingdom)	(Geo Data N/A)	0.85	0.90	0.85	1.0	1.111111
7	(United States of America, Geo Data N/A)	(United Kingdom)	0.85	0.85	0.85	1.0	1.176471
8	(United Kingdom, Geo Data N/A)	(United States of America)	0.85	0.95	0.85	1.0	1.052632
10	(United Kingdom)	(United States of America, Geo Data N/A)	0.85	0.85	0.85	1.0	1.176471
12	(United Kingdom)	(India)	0.85	0.85	0.85	1.0	1.176471
13	(India)	(United Kingdom)	0.85	0.85	0.85	1.0	1.176471
14	(India)	(Geo Data N/A)	0.85	0.90	0.85	1.0	1.111111
17	(India)	(United States of America)	0.85	0.95	0.85	1.0	1.052632
18	(United Kingdom, India)	(Geo Data N/A)	0.85	0.90	0.85	1.0	1.111111
19	(United Kingdom, Geo Data N/A)	(India)	0.85	0.85	0.85	1.0	1.176471
20	(India, Geo Data N/A)	(United Kingdom)	0.85	0.85	0.85	1.0	1.176471
21	(United Kingdom)	(India, Geo Data N/A)	0.85	0.85	0.85	1.0	1.176471
22	(India)	(United Kingdom, Geo Data N/A)	0.85	0.85	0.85	1.0	1.176471
24	(United States of America, United Kingdom)	(India)	0.85	0.85	0.85	1.0	1.176471

Fig. 62. Association rules with 100% confidence

which have tweets less then 0.001% tweets in the data set. There were 39 such countries.To find frequent patterns of country and source in the data,for each source an array of country was used. This resulted in a 2d array with 25 rows and 39 columns. Each row represents a post day and each column represents a country. Transaction encoding, is used to transform data in the form use able by algorithm. Each cell was then marked as true or false depending on if the country had posted the tweet or not.

As we have considered frequent source and in infrequent countries, so min-support was kept low (on 10%). The length was not specified to find all frequent patterns. For 10% as min-support we found 231 frequent patterns. With out considering the length of item set most frequent set contain only British Virgin Islands with 36%. For further analysis we specified length of the item set to be greater then 2. The resulting frequent pattern is shown in the fig below.

As it can be seen all have support equals to 12 %.The longest sequence have length 7.

1) Association rules: There were 2701 rules form and all rules around have 12% confidence.there were 4 rules with

	level_0	index	support	itemsets	len
29	29	29	0.12	(North Macedonia, Mauritius, Saint Kitts and N...	3
30	30	30	0.12	(Mauritius, Saint Kitts and Nevis, Belize)	3
31	31	31	0.12	(North Macedonia, Mauritius, Belize)	3
32	32	32	0.12	(North Macedonia, Mauritius, Saint Kitts and N...	4
43	43	43	0.12	(Belize, North Macedonia, Democratic Republic ...	3
...	...	...	...	...	...
226	226	226	0.12	(Belarus, Montenegro, Guinea, Puerto Rico, Nor...	6
227	227	227	0.12	(Belarus, Democratic Republic of the Congo, Gu...	6
228	228	228	0.12	(Belarus, Democratic Republic of the Congo, No...	6
229	229	229	0.12	(Belarus, Democratic Republic of the Congo, Mo...	6
230	230	230	0.12	(Belarus, Democratic Republic of the Congo, No...	7

154 rows × 5 columns

Fig. 63. Frequent pattern mining by Post on day by less frequent countries

support of 16%. This shows that the post being made by these countries is around 12%. The rules shows that if the country from the post from the anstecents post there are only 12% of chance to see a post from consequent country.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	(British Virgin Islands)	(Liechtenstein)	0.36	0.16	0.16	0.444444	2.777778
1	(Liechtenstein)	(British Virgin Islands)	0.16	0.36	0.16	1.000000	2.777778
2	(North Macedonia)	(Mauritius)	0.28	0.24	0.16	0.571429	2.380952
3	(Mauritius)	(North Macedonia)	0.24	0.28	0.16	0.666667	2.380952
4	(British Virgin Islands)	(Mauritius)	0.36	0.24	0.12	0.333333	1.388889
...	...	...	...	...	...	...	...
2697	(North Macedonia)	(Belarus, Democratic Republic of the Congo, Gu...	0.28	0.12	0.12	0.428571	3.571429
2698	(Guinea)	(Belarus, Democratic Republic of the Congo, No...	0.24	0.12	0.12	0.500000	4.166667
2699	(Puerto Rico)	(Belarus, Democratic Republic of the Congo, No...	0.20	0.12	0.12	0.600000	5.000000
2700	(Montenegro)	(Belarus, Democratic Republic of the Congo, Gu...	0.16	0.12	0.12	0.750000	6.250000
2701	(Mauritius)	(Belarus, Democratic Republic of the Congo, Mo...	0.24	0.12	0.12	0.500000	4.166667

2702 rows × 7 columns

Fig. 64. Association rules for less frequent countries confidence

D. Frequent Pattern Analysis on Hashtag

The are around 249365 unique token out of which 200 top token were used to find frequent pattern in tweets. These hashtags occurs at least 1800 time in the whole data. The rest of the hashtags were considered infrequent. We start by finding frequent pattern of hashtags by tweets. On average 2 of the frequent hashtags occurs in tweets and the max number of hashtags of a tweet is 40. The data is transformed to a 2d array with 1187093 rows × 200 columns.Each row represents a tweet and each column represents a hashtag. Transaction encoding, is used to transform data in the form use able by algorithm. Each cell was then marked as true or false depending on if the hashtag occur in the tweet or not.

As the data set is quite large and the average number of hashtags occurring together is very low, so the min support consider is 1%. There were 712 frequent pattern with out consider the length in consideration. The maximum length in frequent pattern is 4. The frequent pattern by considered at minimum length of itemset of 3, we got 106 frequent patterns. is shown in the fig below.

	support	itemsets	len
217	0.001120	(Trump, BidenHarris2020, Trump2020)	3
218	0.003452	(Election2020, Trump, Trump2020)	3
219	0.001293	(Biden, Trump2020, Election2020)	3
220	0.001136	(Biden, Trump2020, Elections2020)	3
221	0.001788	(Trump, Trump2020, Elections2020)	3
...	...	...	...
641	0.001114	(Election2020, ElectionDay, Trump2020, Electio...	4
690	0.001559	(Election2020, Trump, ElectionResults2020)	3
691	0.001130	(Biden, ElectionResults2020, Election2020)	3
706	0.001001	(Biden, Election2020results, Election2020)	3
707	0.001010	(Election2020, Election2020results, JoeBiden)	3

106 rows × 3 columns

Fig. 65. Frequent pattern mining by Hastags

1) Association rules: There were 637 rules form. We consider the rules with at least 40%confidence. There were 342 rules. This shows that the post being made by these hastages there is 40% chance the consequent has-tag will occurs with it.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
1	(WhiteHouse)	(Trump)	0.003896	0.413920	0.001993	0.511568	1.235911
5	(HunterBiden)	(JoeBiden)	0.008708	0.166679	0.005257	0.603657	3.621681
6	(Trump2020)	(Trump)	0.034868	0.413920	0.019645	0.563394	1.361119
7	(BidenHarris2020, Trump2020)	(Trump)	0.002488	0.413920	0.001120	0.450389	1.088109
8	(Election2020, Trump2020)	(Trump)	0.005810	0.413920	0.003452	0.594171	1.435476
...	...	...	...	...	...	...	...
612	(USWahlen2020)	(Trump)	0.001703	0.413920	0.001055	0.619189	1.495916
613	(USAElections2020)	(Trump)	0.009425	0.413920	0.003776	0.400608	0.967840
615	(ElectionResults2020)	(Trump)	0.012960	0.413920	0.005364	0.413910	0.999976
626	(USElectionResults2020)	(JoeBiden)	0.005831	0.166679	0.002889	0.495522	2.972916
636	(PresidentElectJoe)	(JoeBiden)	0.005400	0.166679	0.002809	0.520281	3.121461

342 rows × 7 columns

Fig. 66. Association rules for Hashtags

VIII. Findings

1) Preprocessing and Exploratory Data Analysis (EDA): For this project, we used the US Tweet Elections Dataset from 2020 and performed various preprocessing steps on them. The most important column was the tweets column, which was preprocessed by removing symbols/emojis from the tweets and removing duplicate entries from them as well. Many of the other columns were preprocessed by removing null entries and duplicates. Columns with a large number of null values were dropped from the dataset. Types of attributes were changed according to the actual property of the column (eg, created_at column was changed to date-time type). In exploratory data analysis, we plotted graphs (scatter plots, bar charts, maps, pie charts, line graphs, heat maps) for various columns to show their relation and the extent of different types of entries present within the columns. EDA on the

tweets dataset showed that Trump had an overall greater popularity than Biden, as can be seen by the total number of tweets and the highest number of hashtags, but sentiment analysis if the tweets shows that most of the negative tweets were associated with Trump as well. Further evidence of Trump's popularity is that when the states and cities were analysed individually, Trump still had a higher number of tweets in all the major areas like New York or California. When the most frequently used words in the tweets were analysed, it was observed that for Trump's tweets, 'joebiden' was a very frequently used word. On the other hand, for Biden's tweets, 'trump' or 'donaldtr' was used significantly less.

2) Cluster Analysis: For this, we used k-means and DBscan clustering to cluster the dataset into sections. Since k-means required us to specify the number of clusters ourselves, thereby introducing chances of errors in the clusters, we used DBscan to automatically select the number of clusters and plot them accordingly. Different combinations of related columns were used for both of these processes. Attributes with variable entries resulted in large number of clusters while those with less unique values resulted in fewer clusters.

It was found through the scatter plots of the clustering algorithms that the popularity of a tweet was dependent on the number of followers of the account that posted it and the retweets and likes of such a tweet are also dependent on the popularity of the user account. Furthermore, analysing the country, city and states of origins of tweets showed that the tweets originated from many different areas (different states, cities and countries) which signifies the worldwide interest in US politics. The popularity of a tweet was also dependent on the candidate that it was being posted in favor of. Clustering carried out on the content of the tweets showed the most frequent words used and the relative intensity of each. (eg, since this was the US elections tweet dataset, two of the most frequent words were the candidates who were running for office). Clustering the number of likes based on the candidates showed that most of the posts that were in favor of Trump garnered a larger popularity than the ones about Biden.

3) Frequent Pattern Mining: For this, we used FP-growth and apriori to find frequent patterns. Every combination of columns was first analysed to divide them into unqiue entries and find their corresponding quantities. The algorithms were in turn applied on them to create tables of the number of frequent patterns. Some of these were plotted on graphs to visualize the results. Since FP-growth is faster than Apriori, majority of the patterns were produced with FP-growth.

These patterns showed that the top country that posted the most amount of tweets was the US. Additionally, British Virgin Island was the countries which posted the least about the elections. The most popular hashtags for the tweets were "Trump", "BidenHarris2020" and "Trump2020". While analysing hashtags, it was noticed that most of the hashtags were in support of Trump.

IX. Conclusion

In short, this project used exploratory data analysis, frequent pattern mining, and cluster analysis techniques on the US tweet election dataset 2020. Through exploratory data analysis, we gained insights into the dataset's distribution, correlations, and outliers. Frequent pattern mining allowed us to identify frequent itemsets and rules in the tweet data related to the 2020 US election. Cluster analysis revealed distinct clusters of tweets based on their content and sentiments. From these, we were able to gain a comprehensive understanding of the tweet data and provide insights into the 2020 US election. However, there are some limitations to these processes, such as the biases in the Twitter data. Future analysis can focus on using other data sources and more advanced processing and clustering techniques to improve the accuracy and reliability. Overall, this project highlights the importance of data mining techniques in analyzing large-scale social media data to gain insights into real-world events.