Term Project:
Programming and Data Processing
Group 8

Title: Influence of Tweets on Canadian Federal Elections in 2019

(*The Python code used for our analysis is attached a supplement to this document*)

Abstract

The purpose of this research is to analyze the impact of tweets & various actors who tweeted on the migration during the Canadian Federal elections 2019. In this research, we performed sentiment analysis and found out that the tweets mostly included negative sentiments. We also implemented time-series analysis to observe the variations of sentiments with time and figured out that the trend of tweets was positive at the beginning of the period but eventually converted to a negative trend as the elections approached near. Furthermore, this research also incorporates an ML model to predict the sentiment of the tweets.

Keywords: Immigration, Immigrant, Elections, Refugee, Trudeau, Canadian Politics.

Introduction

Purpose

This research was performed to analyze the role of Twitter in Canadian Federal elections in 2019 shaping the online discussions related to migration.The purpose & objective was-

- To perform sentiment analysis on the tweets to categorize the tweets on the basis of their negative or positive influence.
- To analyze the influence of the tweets & actors tweetings on migration & election
- Time series analysis to observe variation of sentiments throughout the duration of elections.
- Implement Machine Learning model to predict sentiment of tweets.

Scope

The scope of this project was limited to the analysis of the 4,966 Election Tweets in JSON format which was shared with us by the Professor. The dataset is usually obtained via twitter APIS and contains important columns like status updates, usernames of people who wrote the tweets, their locations, hashtags etc.

Literature Review

- **Understanding the dataset.**

The dataset we used as our source are tweets encoded in JSON format.
*"All Twitter APIs that return Tweets provide that data encoded using **JavaScript Object Notation (JSON)**. JSON is based on key-value pairs, with named attributes and associated values. These attributes, and their state are used to describe objects." (Introduction, n.d., para1)*

With each tweet various objects are used to store the related data-

- ➢ **Tweet object**- contains fundamental attributes like
    1. id- Unique identifier of the tweet.
    2. Created_at- The datetime when the tweet was created.
    3. Text- The actual UTF-8 text of the status update. (*Tweet Object*, n.d., *Tweet Data Dictionary)*
- ➢ **User object-** User objects can be retrieved from id and screen_name.
    1. id- Unique identifier of the user.
    2. name- Name of the user.
    3. Screen_name- alias, handle that the user identifies as on the screen.(*User Object*, n.d., *User Data Dictionary)*
- ➢ **Entities object-** contains objects like hashtags, urls, user mentions.
- ➢ **Extended entities object-** For tweets that have attached photos, animated GIF or videos.(*Extended Entities Object*, n.d., *para2)*
- ➢ **Geo objects-** When users assign locations to their tweets.

The research for analyzing the impact of tweets on the Canadian Federal election involved numerous tasks and in order to support this analysis we utilized varios python libraries available that helped us to achieve desired results.

- **Cleaning the data and removing noise using Natural Language Toolkit library and storing in pandas dataframe.**

"NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries."(NLTK :: Natural Language Toolkit, n.d., para1)

We utilized this library to analyze tweets and removed noise of unnecessary data like urls, numbers, repetitions, punctuations etc. to narrow down to only useful words in the dataset using packages like string, PortStemmer, Wordnet Lemmatizer, regex. All the cleaned data was then stored in pandas dataframe to prepare it for analysis.

- **Sentiment Analysis using TextBlob library.**

*"TextBlob returns polarity and subjectivity of a sentence. **Polarity** lies between [-1,1], -1 defines a negative sentiment and 1 defines a positive sentiment. TextBlob has semantic labels that help with fine-grained analysis. Subjectivity lies between [0,1]. **Subjectivity** quantifies the amount of personal opinion and factual information contained in the text. The higher subjectivity means that the text contains personal opinion rather than factual information." (Shah, 2021, para 5)*

- **Sklearn for Machine Learning.**

Scikit learn library of Python provides various algorithms for building machine learning models. We utilized this library to build a **logistic regression model** because the dependent variable 'y' was a **categorical variable** with two possible classes- **'Negative' and 'Positive'.** We trained the model by dividing data into 80%-20% ratio to train and test the data.

Data Description

The dataset used for the analysis is a collection of about 4,966 tweets during the elections in 2019 in **json** format. The most important elements in the dataset that could be helpful to analyze the influence of tweets on the elections are -

- **Full Text**- consists of the actual tweets.
- **Screen_name**- Usernames of the actors.
- **Name**- Actual name behind those screen names.

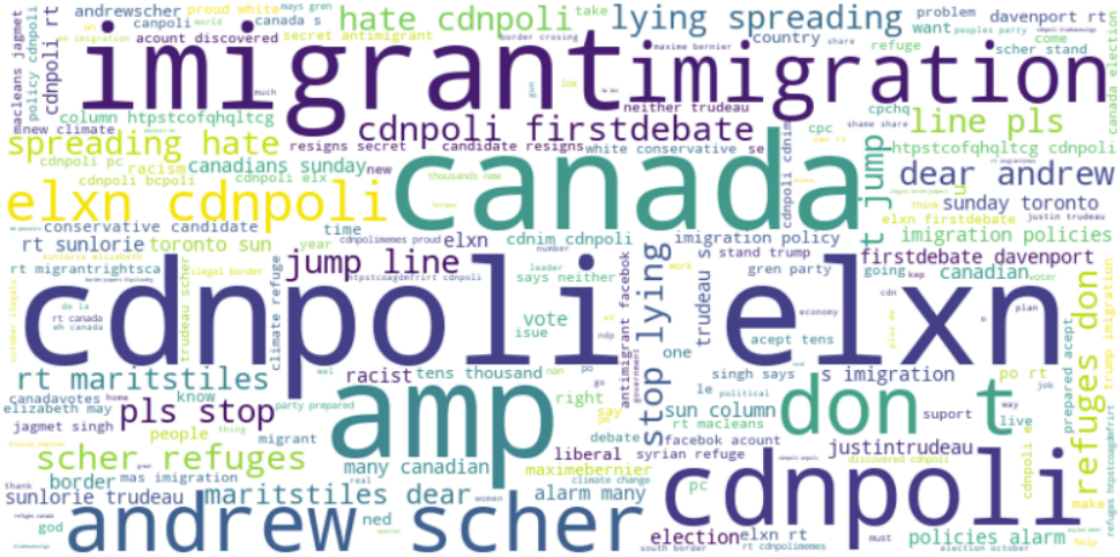
A. Methodology

1. The original dataset was in a json file, so it was converted into a usable format to prepare it for performing the analysis. The most useful columns such as **full_text, screen_names, names** were extracted and cleaned for the research and appended to the original dataset.

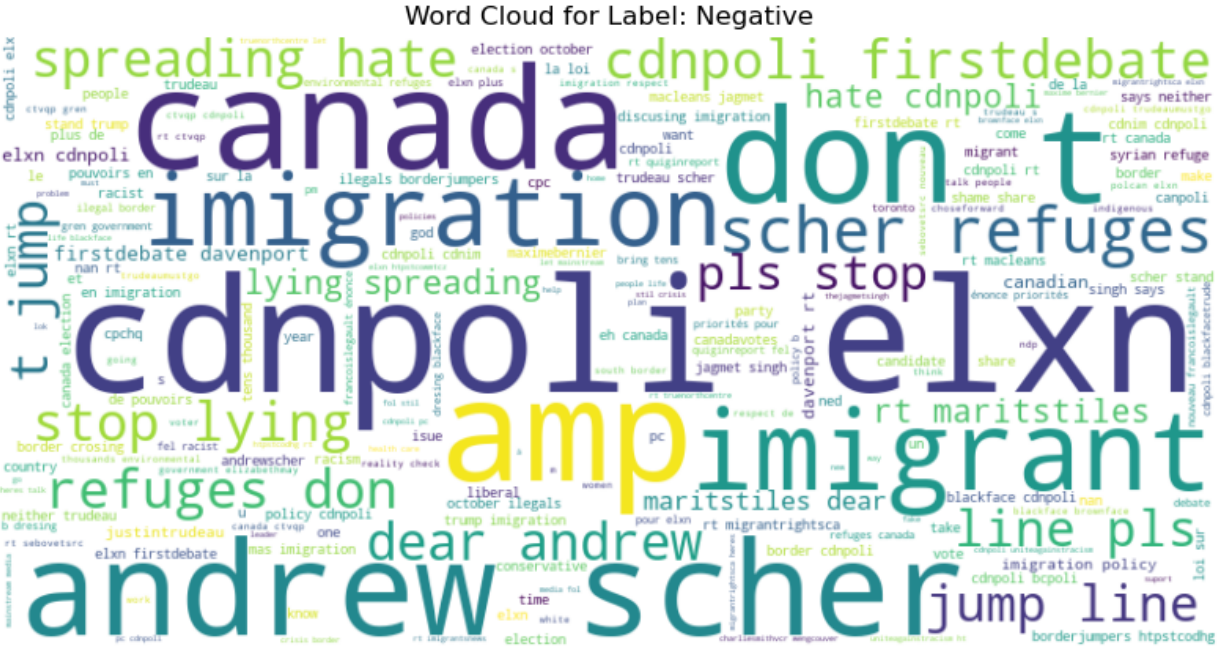
2. Using the **NLTK** library, any stopwords, punctuations, repetitive words, urls, numbers were removed from the tweet texts in order to remove noise from the data. Further, **stemming** was performed to remove any prefix or suffix from the words and also, **lemmatization** was performed to find the original dictionary word on the basis of the context of a tweet.
3. For the screen_name column, the most frequently involved usernames were extracted and plotted on a bar graph.
4. After cleaning the full_text column, each tweet was tokenized and **sentiment analysis** was performed on it using the **'TextBlob'** library and tweets were categorized on the basis of **positive and negative** sentiments. This information was stored in columns and appended to the original dataset.
    - '**Polarity**' - A float value between -1 and 1, negative value represents negative sentiment and positive value represents positive sentiment of the tweet.
    - '**Labels**'- Whether a tweet had positive or negative sentiments.
5. The ratio of positive vs negative tweets was calculated to analyze the overall influence of the tweets on the elections.
6. The frequency distribution of positive and negative words was also calculated to identify the words that were highly used for each sentiment.
7. The tweets were also analyzed based on their source(screen_name) to find top screen_names which had positive or negative influence.
8. A machine learning model was implemented using the **'sklearn'** library to predict sentiment of a tweet using logistic regression.
9. Using the **'networkx'** library, networks were created on the basis of co-occurrence of positive and negative words in the tweets.
10. **Time - Series analysis** of variation in average polarity of sentiments of tweets and maximum and minimum value of polarity by each week as well as each day.
11. Then, by observing the resultant visualizations, the overall impact of these tweets was concluded.
12. Python visualization libraries like **matplotlib, seaborn** and **plotly express** were used to create insightful graphs.

B. Exploratory Data Description

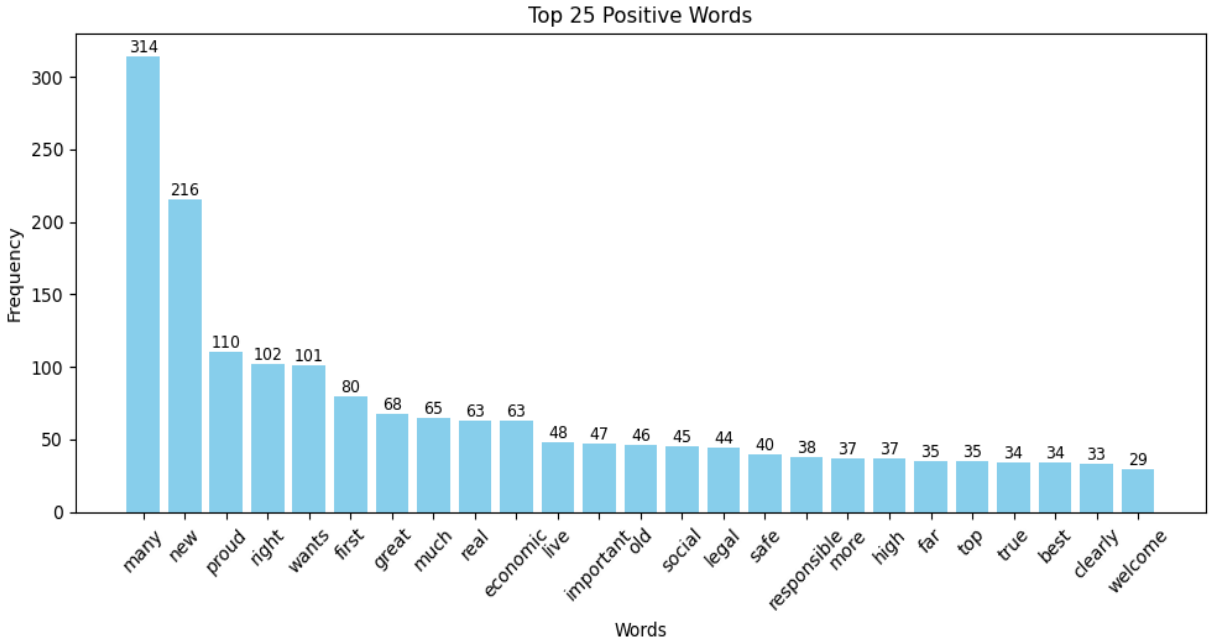
- **Word cloud for most used words in the tweets.**



- **Frequently Tweeted Negative Words**

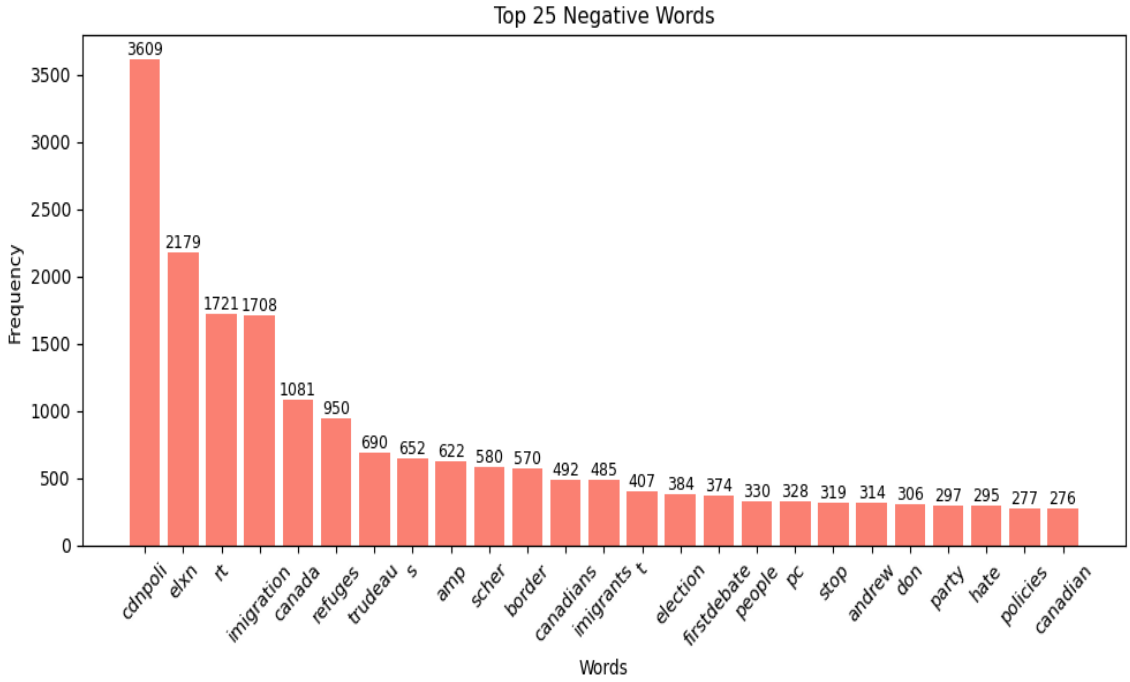


- **Top 25 most frequently used words carrying positive sentiments.**



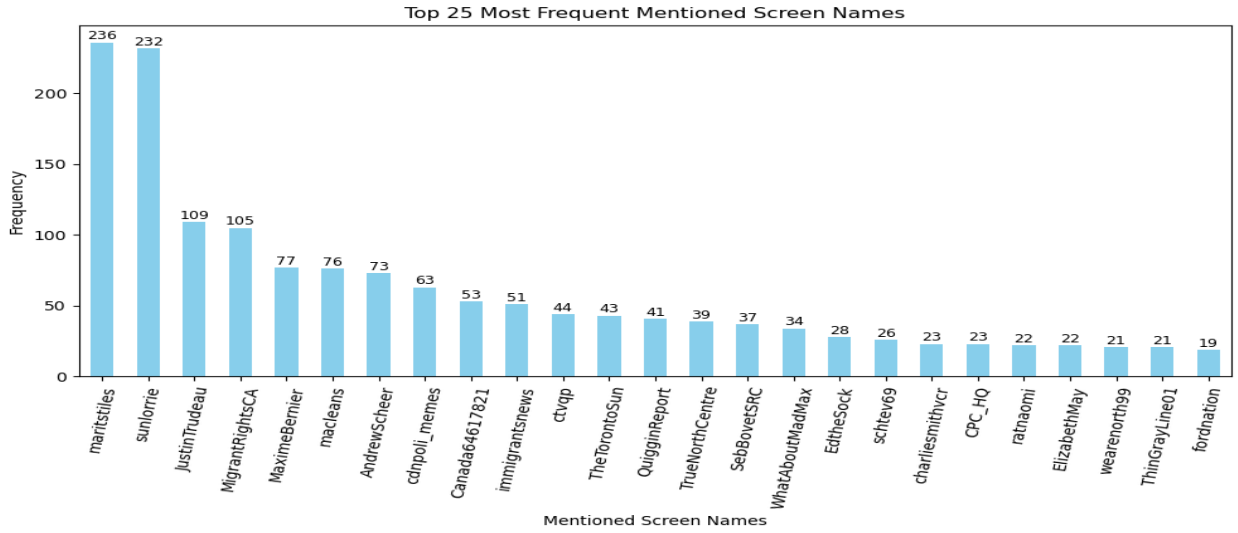
- **Top 25 most frequently used words carrying negative sentiments.**

Words like **'immigration', 'immigrant', 'cdnpoli'** were most frequently mentioned in the Tweets. (as per the WordCloud & TextBlob Analysis), & they had a negative connotation.



- **Screen Names driving the climate.**

Words like **'justinTrudeau', 'MigrationRightsCA', 'immigrantnews' & 'sunlorie'** among the top users who tweeted, & overall top drivers had more negative tweets vs positive tweets.



C. Limitations

Data Transformation

- Flatten out JSON dataset in a cleaned & structured format for analysis.
- Understanding the data schema & how records were related to each other.
- Memory crashes in data processing.
- Selection of best Libraries & Methods to use for the research.
- To choose whether to perform analysis by Tweets or by Words?

Visualization

- Inferring meaningful insights from the data & deriving on the overall influence of Tweets on Election Behavior

Machine Learning

- Choose best ML Library to use for Predictive Modeling

Prediction Models

Steps followed to design the model:

Step 1: Used Textblob using the sentiment.polarity function to convert tweets to label them as positive or negative

In [190]:
```
from textblob import TextBlob
```

In [191]:
```
def getPolarity(text):
        return TextBlob(text).sentiment.polarity
```

In [192]:
```
def sentiment_analysis(data):

    # Iterate over each row in the DataFrame
    for index, row in data.iterrows():
        text = row['full_text']
        text = ' '.join(text)
        blob = TextBlob(text)
        polarity = blob.sentiment.polarity
        polarity = getPolarity(text)

        # Assign values to new columns for each row

        data.at[index, 'Polarity'] = polarity

        # Assign sentiment analysis based on polarity
        data.at[index, 'Labels'] = 'Negative' if polarity <= 0  else 'Positive'

    return data

# Call the sentiment analysis function with your data DataFrame
data = sentiment_analysis(data)
print(data)
```

Step 2: We used TfidfVectorizer to convert text to feature vectors (0 or 1) for designing our model

Step 3: We created a split of 80% training & 20% for test & designed Logistics Regression model (1000 max iterations) to get the best parameters

Step 4:

i) We received an accuracy & precision score of ~85% from the model - reflected the models was really good in predicting the tweets labels for us

ii) Analyzing the confusion matrix on the test data, we get 81% True Negative & 93% True Positive, so it reflects the model was very accurate in predicting the test data values as well

ML MODLE LOGISSTIC REGRESSION

In [225]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.preprocessing import LabelEncoder
```

In [218]:
```
# Extracting features (full_text) and labels from the DataFrame
data['texts'] = data['full_text'].apply(lambda x: ' '.join(x))  # Joining the list of words into a single string per row

# Label encoding for categorical labels
label_encoder = LabelEncoder()
encoded_labels = label_encoder.fit_transform(data['Labels'])
```

In [222]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data['texts'], encoded_labels, test_size=0.2, random_state=42)
```

In [223]:
```
# Initialize the TF-IDF vectorizer
tfidf_vectorizer = TfidfVectorizer()

# Fit and transform the training data into TF-IDF matrix
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)

# Transform the test data into TF-IDF matrix using the fitted vectorizer
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Initialize and train the Logistic Regression model
logistic_regression = LogisticRegression(max_iter=1000)
logistic_regression.fit(X_train_tfidf, y_train)
```

Out[223]:
```
LogisticRegression(max_iter=1000)
```

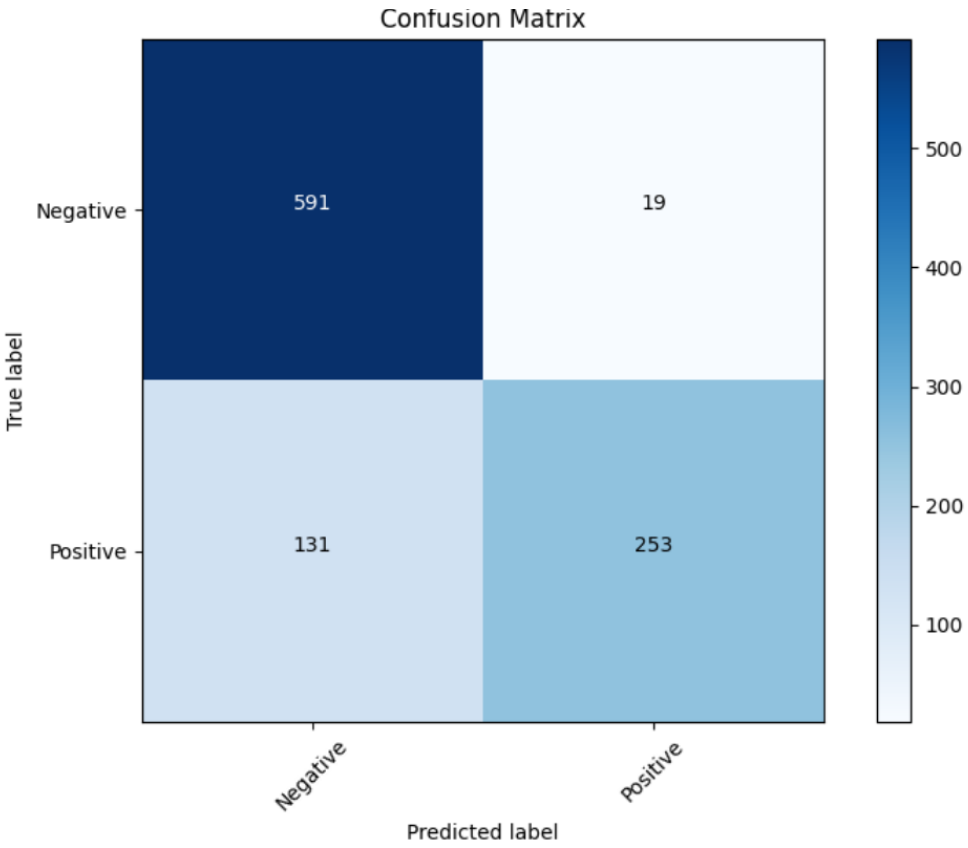
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [224]:
```
# Make predictions on the test set
predictions = logistic_regression.predict(X_test_tfidf)
# Inverse transform the encoded predictions to original labels
decoded_predictions = label_encoder.inverse_transform(predictions)
```

In [226]:
```
# Calculate metrics
accuracy = accuracy_score(y_test, predictions)
precision = precision_score(y_test, predictions, average='weighted')
recall = recall_score(y_test, predictions, average='weighted')
f1 = f1_score(y_test, predictions, average='weighted')

# Print the metrics
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1}")
```

```
Accuracy: 0.8490945674044266
Precision: 0.8616678070705477
Recall: 0.8490945674044266
F1 Score: 0.8425567688707503
```

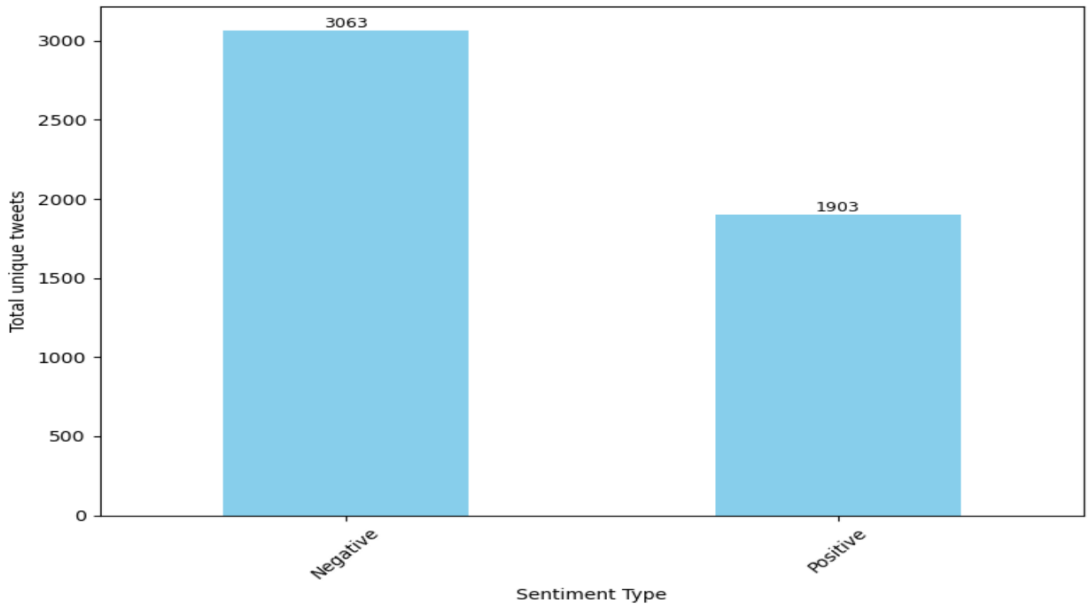

Confusion Matrix
Negative
Positive
True label
591
19
131
253
Negative
Positive
Predicted label
500
400
300
200
100

Experimental Results & Analysis

Sentiment Analysis

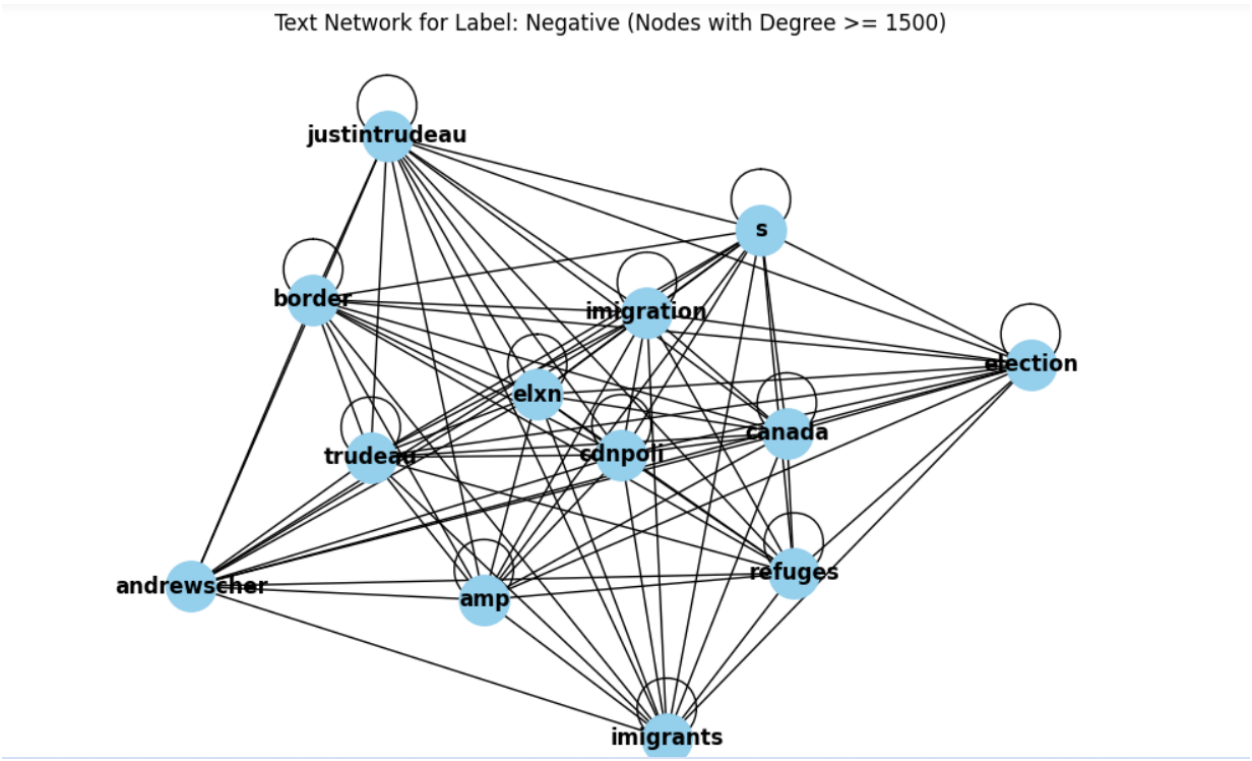
- **Overall Tweets Thermometer**

60%+ Tweets had negative sentiments & were overall classified as negative (per Textblob polarity analysis) - this implies that overall the tweets were trying to create a negative influence on modifying the election behavior

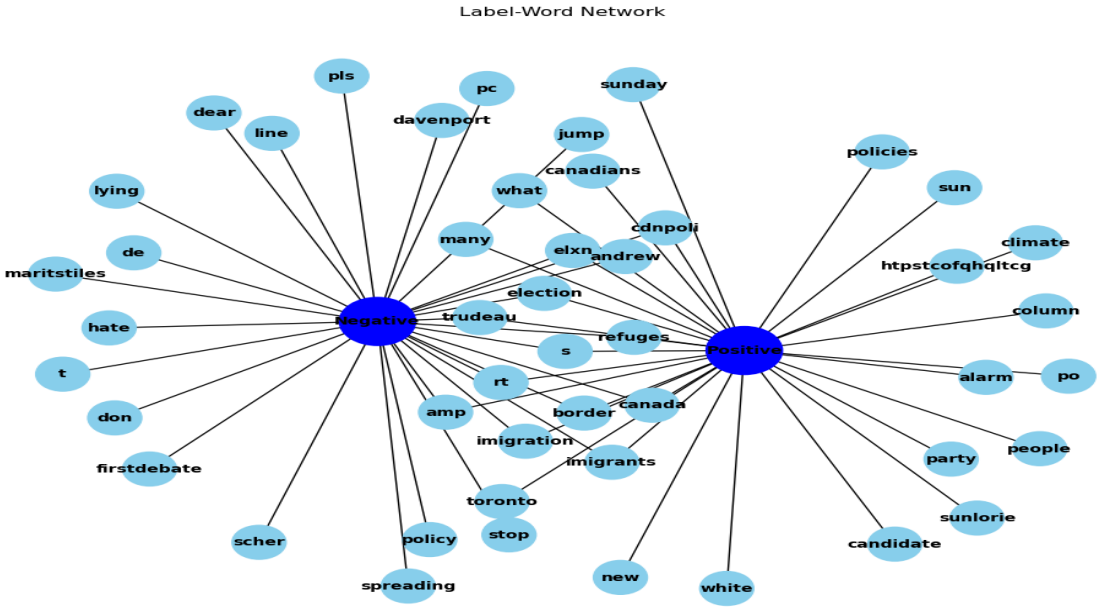


Network of negative vs positive tweets

- As per the top 25 frequently used negative words (per Textblob analysis), most of the negative words were directly targeted on immigration & migration; eg **'immigration', 'canada', 'refugees', 'threadu', 'immigrants', 'elections'**.
- As per the Network Diagram Analysis, words like **'trudeau', 'refugees', 'immigrants' & 'canada'** were tightly connected in the network, along with the negative words
- **Most used negative words that frequently appeared together:**



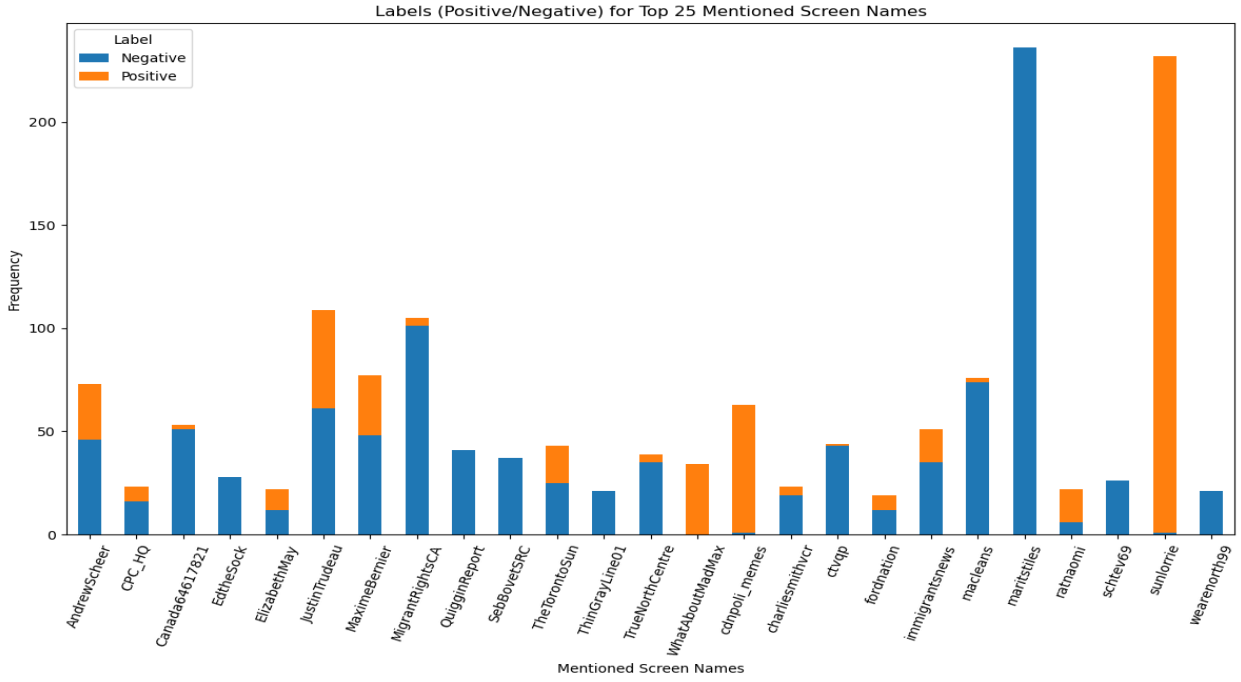
- **Network diagram for positive and negative words.**



- **Screen Names Driving the Climate & what they were saying**

  - Most of the users had higher frequency of words in negative connotations vs the positive ones.

  -Most political entities were actively engaged with the tweets ('AndrewScheer', 'ElizabethMay', 'JustinTrudeau', 'MaximeBernier', 'Macleans'), and most of the actors were more negative compared to the positive ones.



- Analysis of top 10 keywords related to migration & top 10 actors actively tweeting. Most of the keywords related to migration were frequently mentioned in a -ve tone.

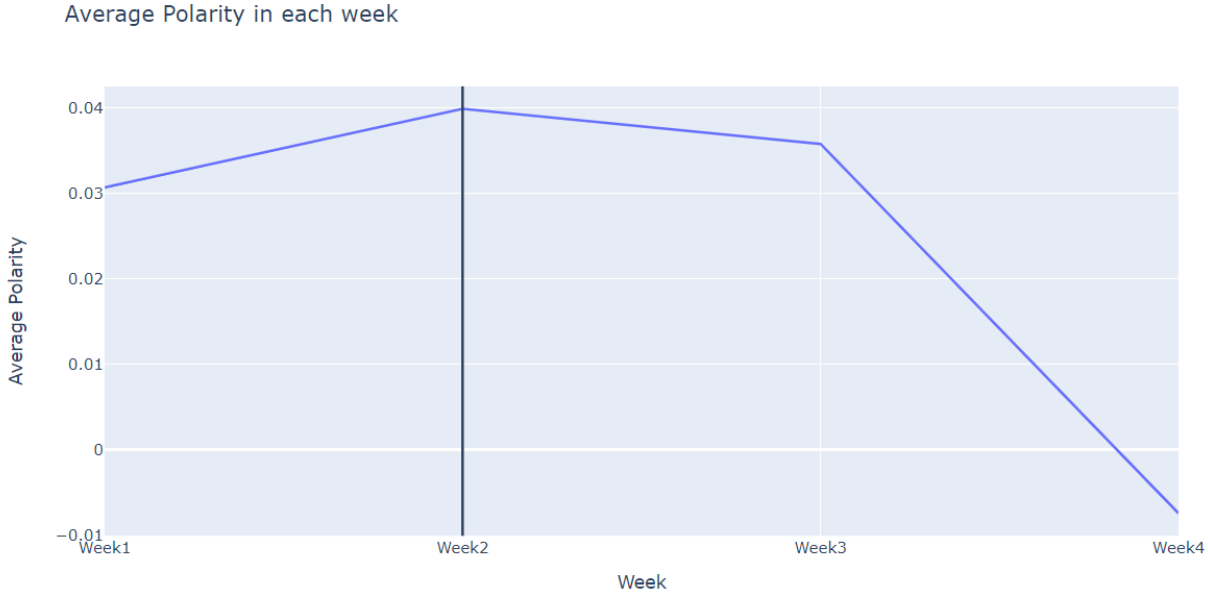
  1. **Immigration:** 40%+ negatively mentioned
  2. **Canada:** 79% negatively mentioned
  3. **Refugees:** 78% negatively mentioned

	cdnpoli		elxn		rt		imigration		canada		refuges		trudeau		s		amp		scher		Total_user_mentions
	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	
maritstiles	0.0	236.0	0.0	0.0	0.0	472.0	0.0	0.0	0.0	0.0	0.0	236.0	0.0	0.0	0.0	1888.0	0.0	0.0	0.0	236.0	3068.0
sunlorrie	231.0	1.0	1.0	0.0	348.0	0.0	161.0	0.0	0.0	0.0	59.0	0.0	161.0	0.0	1585.0	7.0	0.0	0.0	0.0	0.0	2554.0
JustinTrudeau	37.0	49.0	27.0	33.0	23.0	19.0	8.0	15.0	17.0	27.0	10.0	10.0	66.0	94.0	515.0	577.0	14.0	21.0	5.0	8.0	1575.0
MigrantRightsCA	2.0	39.0	4.0	100.0	2.0	117.0	0.0	0.0	0.0	1.0	0.0	0.0	2.0	1.0	36.0	511.0	1.0	7.0	0.0	5.0	828.0
MaximeBernier	26.0	34.0	12.0	27.0	10.0	25.0	20.0	40.0	11.0	20.0	1.0	1.0	11.0	10.0	240.0	426.0	6.0	9.0	1.0	5.0	935.0
macleans	1.0	74.0	2.0	72.0	1.0	70.0	2.0	71.0	0.0	1.0	0.0	4.0	1.0	71.0	24.0	532.0	3.0	1.0	1.0	73.0	1004.0
AndrewScheer	21.0	34.0	8.0	32.0	13.0	15.0	8.0	8.0	11.0	10.0	5.0	12.0	12.0	7.0	290.0	514.0	5.0	20.0	37.0	73.0	1135.0
cdnpoli_memes	124.0	1.0	0.0	1.0	62.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	372.0	17.0	0.0	0.0	0.0	2.0	580.0
Canada64617821	2.0	0.0	0.0	0.0	2.0	51.0	0.0	0.0	6.0	102.0	0.0	0.0	0.0	0.0	6.0	306.0	0.0	0.0	0.0	0.0	475.0
immigrantsnews	6.0	16.0	12.0	26.0	22.0	48.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	6.0	87.0	213.0	0.0	6.0	2.0	8.0	455.0
Total	450.0	484.0	66.0	291.0	483.0	817.0	199.0	134.0	45.0	165.0	75.0	263.0	253.0	189.0	3155.0	4991.0	29.0	64.0	46.0	410.0	NaN

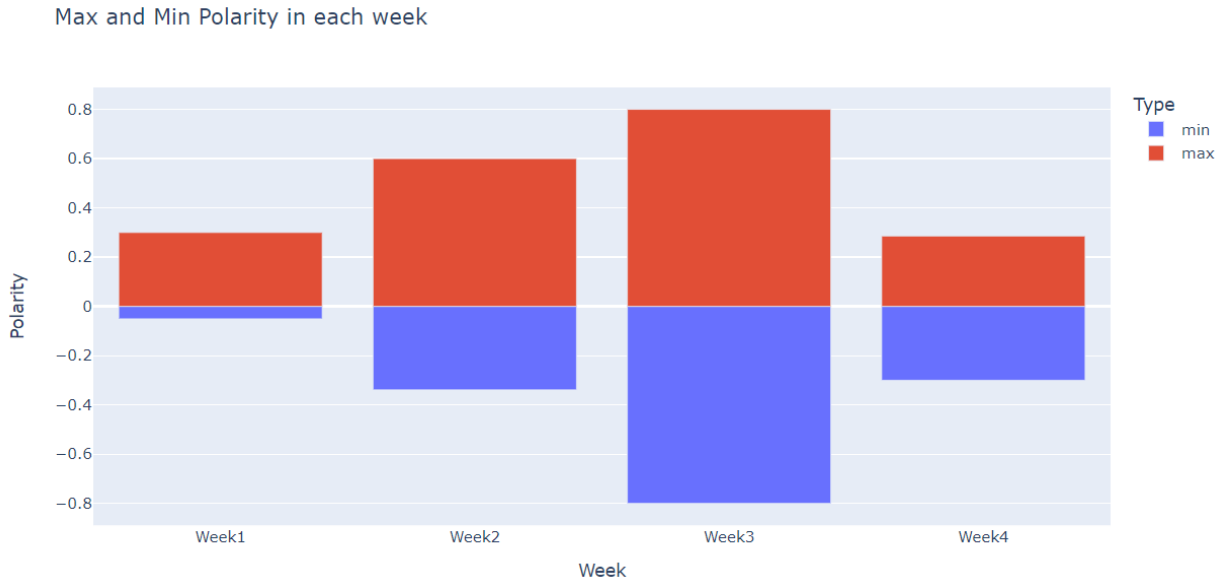
Time Series Analysis of Tweets related to Migration

BY EACH WEEK

- **Change in Average polarity of sentiment of tweets.**
  In the first week, the average polarity of tweets was positive and reached the peak of 0.04 in the second week, but started declining after that and eventually dropped to the lowest value of -0.01 in the last week.



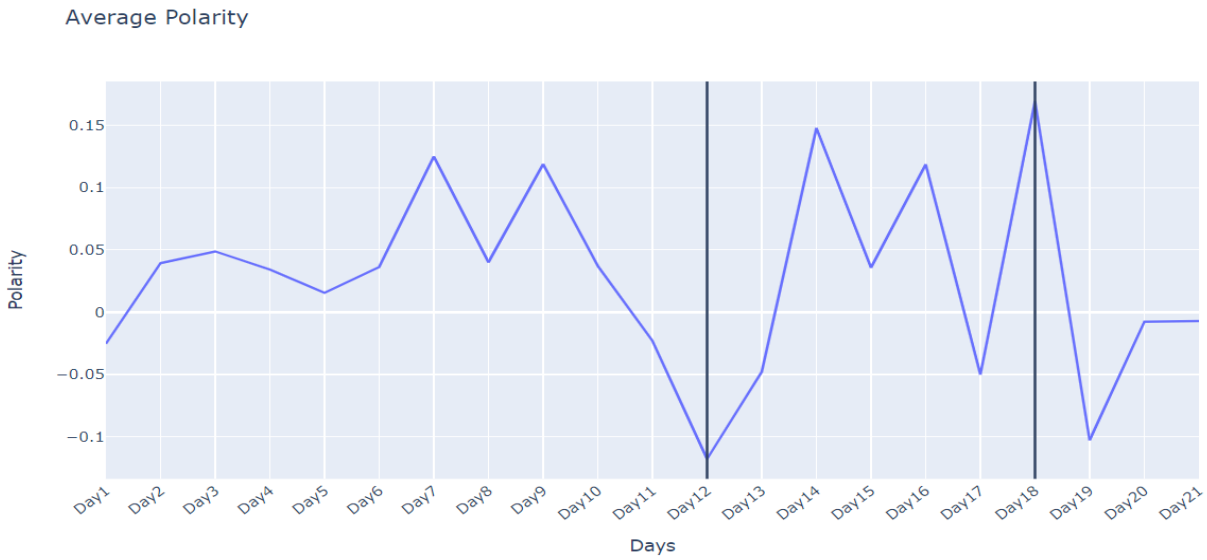
- **Variation in maximum and minimum value of polarity of tweet sentiments by each week.**
  In the first week, the influence of negative sentiments was very low as compared to positive tweets but as we approached the third week the maximum and minimum polarity of sentiments was same and finally in the last week negative sentiments dominated.



BY EACH DAY

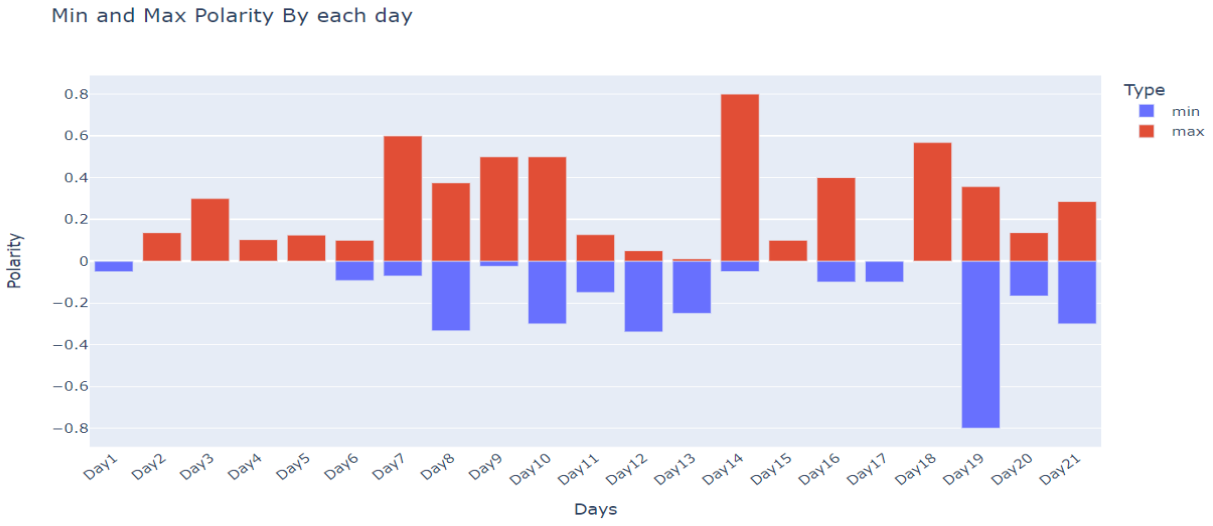
- **Change in Average polarity of sentiment of tweets**

  The average polarity was positive in the initial days but hit the lowest on Sep 22 and recovered back to positive in the following days, but ended up on the negative side on Oct 11 indicating that average sentiments were negative on the last day.



- **Variation in maximum and minimum value of polarity of tweet sentiments by each day.**

  The tweets had more negative influence on the first day but the maximum value of polarity was on the positive side for the next few days. On Sep 25, positive polarity dominated with a value of 0.8 but on Sep 28, negative sentiments had taken the charge and dominated till the last day.



Conclusion

Overall, the Tweets during the Canadian Federal Election 2019 regime negatively influenced the election campaign:

- **Sentiment Analysis:**

Most of the tweets were negative & most frequently tweeted words related to immigration & migration were also the most frequently tweeted negative words & also the top negative words were also focused on migration & similar concepts (eg immigration/immigrant/refugees) . So, overall the tweets were anti-migration & directly opposing any sort of migration or any similar concept & any policies related to that.

Most of the active users tweeting were political entities ((AndrewScheer', 'ElizabethMay', 'JustinTrudeau', 'MaximeBernier', 'Macleans'), and most of their tweets were more negative than positive - this implied that political entities were trying to create a negative environment on immigration/migration as well & were trying to influence the election behavior accordingly

- **Network of tweets:**

Most of the migration related concepts (immigration/immigrant/refugees) were tightly connected in the network diagram & were frequently mentioned together in the tweets (had >=1500 degree), & also formed the negative words spectrum of -ve words network as well. So, this implied that most of the migration related concepts were frequently mentioned together, & frequently mentioned in a negative connotation/context.

Also, in the network analysis of negative words, most of the political entities (trudeau, andrew scheer) were tightly connected with immigration/migration related words that were portrayed in a negative context, this implied, that most of the tweets opposed migration & all political actors who supported a similar concept as well & in a strong way as well (tightly connected negative word network!)

- **Time Series Analysis:**

According to our analysis, the tweets related to elections started from a slightly negative trend but then transformed into positive for some days but as Oct 1, 2019 approached near, most of the tweets with negative sentiments dominated. This implied that near the start of the election time , the overall temperature/sentiment on social media was negative & it negatively influenced the election regime

References

[1] NLTK :: Natural Language Toolkit. (n.d.). https://www.nltk.org/

[2] Introduction. (n.d.). Docs | Twitter Developer Platform.

https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview

[3] Tweet object. (n.d.). Docs | Twitter Developer Platform.

https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet.

[4] User object. (n.d.). Docs | Twitter Developer Platform.

https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/user

[5] Extended entities object. (n.d.). Docs | Twitter Developer Platform.

https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/extended-entities

[6] Shah, P. (2021, December 15). Sentiment Analysis using TextBlob - Towards Data Science. Medium.

https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524