USA Elections

Filip Strzałka (242556),Gianmarco Pepi(531425), Monia Bennici(534792)

January 14, 2021

1 Introduction

We are going to analyse data about the elections in Usa in 2020, to understand the general sentiment of people around the world through tweets. The original dataset is composed by 776k tweets related to Joe Biden and 970k to Donald Trump. Each tweet is described by its text and 19 additional attributes: 4 tweet attributes (timestamp, likes, retweets, source: webapp/mobile),8 user attributes (nickname, screen name, join date),7 location attributes (country, city, state, latitude, longitude). To train a model we also have a support dataset, that contains sentences and the sentiment of that sentence labeled as positive, neutral or negative. The main goal of this work is to create a timeline visualisation of general twitter community attitude to candidates.

2 Stage 1: Data Cleaning

We opened the csv files as dataframes through SparkSession. Both Trump file and Biden file contains missing values in the columns source, user_name, user_description, user_location, lat, long, city, country, continent, state and state_code. Since we do not need all the 20 attributes for the analysis, we drop the columns we considered useless. For the purpose of the project we just used created_at and tweet that do not contain missing values, but we also left likes, lat, long and city for future aims maintaining the missing values. Also, we detected 1251 duplicates rows in Trump file and 1851 in Biden one. So we deleted them because they are not useful. The last step consists of removing special chars (as hashtags, url's etc) and we did it with the function clean_text, obtaining rows without text when the tweet is just composed of these chars. This problem will be solved in the next steps. Finally, we saved the clean dataframes as csv files in HDFS. In figure 1 and 2 are shown the final dataframes.

	created_at	tweet	likes	lat	long	city
0	2020-10-15 00:12:49	Any of you EVER tell the truth tested positive...	0.0	-31.952712100000003	115.86047959999999	Perth
1	2020-10-15 00:16:13	Under Section Twitter and Facebook limit Rudy ...	0.0	39.7837304	-100.4458825	None
2	2020-10-15 00:16:35	Its never gonna be the end for us is it Crowd ...	2.0	45.5202471	-122.6741949	Portland
3	2020-10-15 00:22:57	tteribul MelliBitch dumob jtlittle Tazdad fuff...	4.0	None	None	None
4	2020-10-15 00:25:09	andymstone Yeah and why do you let disinformat...	0.0	39.7837304	-100.4458825	None

Figure 1: Trump dataframe

	created_at	tweet	likes	lat	long	city
0	2020-10-15 00:03:17	Hunter introduced his father then Vice Preside...	1.0	40.7127281	-74.0060152	New York
1	2020-10-15 00:16:13	Under Section Twitter and Facebook limit Rudy ...	0.0	39.7837304	-100.4458825	None
2	2020-10-15 00:22:02	No Today were going to make the evil nutter a ...	1.0	None	None	None
3	2020-10-15 00:22:57	tteribul MelliBitch dumob jtlittle Tazdad fuff...	4.0	None	None	None
4	2020-10-15 00:33:06		0.0	38.475840600000005	-80.84084150000001	None

Figure 2: Biden dataframe

Also the support dataset had to be clean. First of all, we had two csv files (train.csv and test.csv) containing the same information (text, sentiment) plus the column "selected_text in train.csv, so

we decided to merge them in one, dropping from train the columns "selected test". The dataset contained some missing values, we deleted them and there was not any duplicate row. Also in this case, we deleted special char with clean_text and we saved the obtained dataframe in HDFS. It is shown in figure 3.

	text	sentiment
0	i lost all my friends im alone and sleepy i wa...	negative
1	yeah I was thinking about that ahaha	positive
2	The birds are out oh man Thats NOT cool I didn...	negative
3	Im missing crab legs and attending my going aw...	negative
4	there were attempts to somehow extend inner c...	neutral

Figure 3: Support Dataset

3 Stage 2: Text Analysis

For the text analysis it was necessary to split the text into words and delete rows with the blank text created in the previous step and we did it with the function split_text. We first analyse the support dataset. The analysis is based on the frequency of a word to be in a class of the support dataset with respect to the frequency to be in the support dataset in general. The result are shown clearer in figure 4. In this way, we consider the most probable words in every class, but discarting the words that appear very often in every sentence (like some stopword). Using the same approach, we computed the probability for a word to be in a tweet of Trump dataframe, with respect to the probability to be in the other dataframes, as shown in figure 5. The same was carried using Biden dataframe. An alternative approach is a simple word count, that just returns the most frequent words. To avoid having the stopwords as most frequent, we eliminated them. It can be seen, howev that the most used words are not really meaningful, since they are the name of the candidates and some verb (vote or going). We show the results in figure 6 and, despite they can be good, we decided not to delete the stopwords, because, as shown in the previous image, they can be useful in order to classify a tweet as positive, negative or neutral (es: the word "or" appears between neutral words). Nevertheless, this trivial method shows an interesting result, eg. that the argument "amp" (american priority) was a debated argument in twitter community. In general we can notice how the tweets align with current events ("unscietific", "social security", "ghislaine", "climate") and how twitter community pushes people for vote.

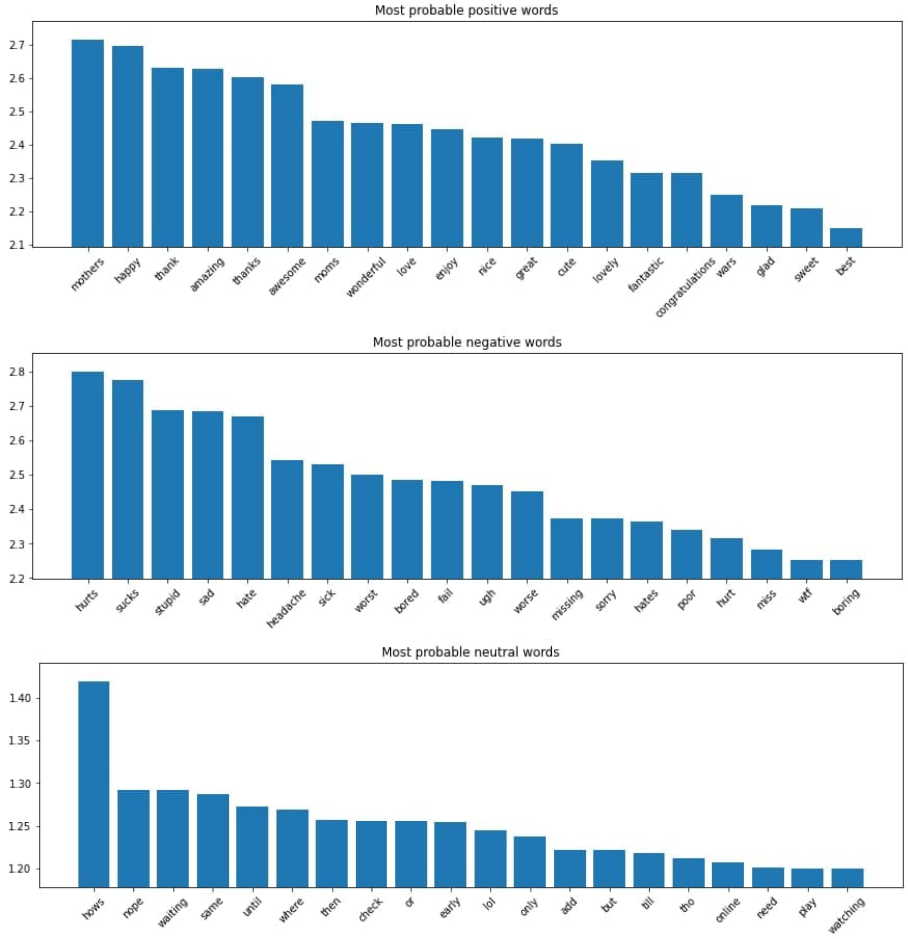
Most probable positive words
2.7
2.6
2.5
2.4
2.3
2.2
2.1
mothers
happy
thank
amazing
thanks
awesome
moms
wonderful
love
enjoy
nice
great
cute
lovely
fantastic
congratulations
wars
glad
sweet
best
Most probable negative words
2.8
2.7
2.6
2.5
2.4
2.3
2.2
hurts
sucks
stupid
sad
hate
headache
sick
worst
bored
fail
ugh
worse
missing
sorry
hates
poor
hurt
miss
wtf
boring
Most probable neutral words
1.40
1.35
1.30
1.25
1.20
hows
nope
waiting
same
until
where
then
check
or
early
lol
only
add
but
till
tho
online
need
play
watching

Figure 4

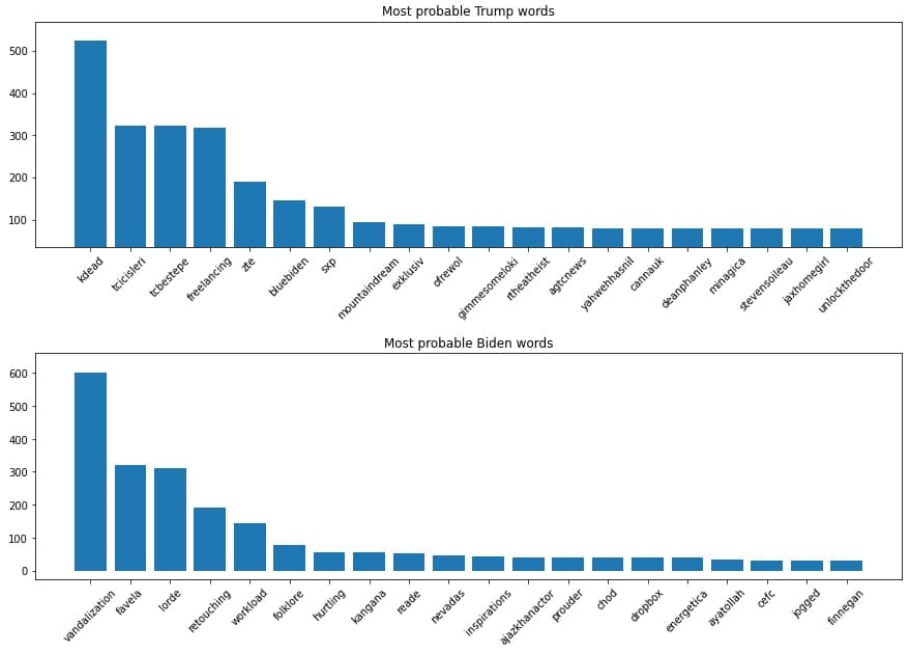

Figure 5

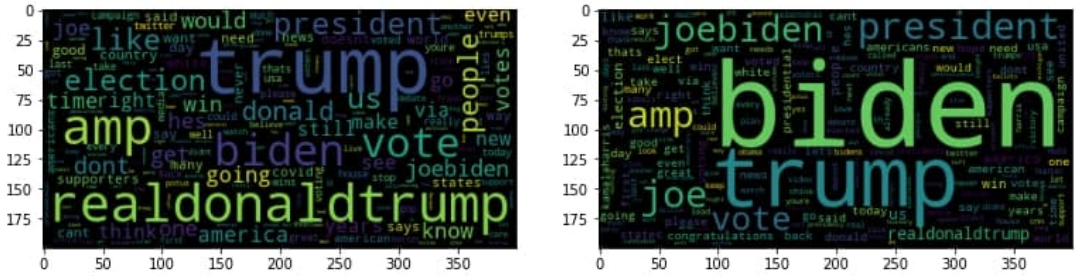

Figure 6

4 Stage 3: Classification

In order to classify our dataset, we featured the text data. In particular, we used three vectorization methods, that are word2vec, bag of words and TF-IDF. Reminding that every line of the dataset is split into words, the bag of words build a dictionary from the set of these words, returning vectors. Also word2vec uses the single words and it place every word in a point in the space, where the more two words are near, the more are similar. Finally, the TF-IDF compute the probability to find a word in the document, giving more importance to rare words (so the words that appear less in lines). Using these three method, we trained two classifiers on the support dataset: Naive-Bayes (results shown in table 1) and Multilayer Perceptron (results shown in table 2).

Vetorization method	Accuracy	F1-score	W-Precision	W-Recall
Word2vec	0.47	0.40	0.57	0.47
BOW	0.62	0.62	0.66	0.62
TF-IDF	0.53	0.52	0.57	0.53

Table 1: Naive Bayes Classifier

Vetorization method	Accuracy	F1-score	W-Precision	W-Recall
Word2vec	0.56	0.55	0.56	0.56
BOW	0.64	0.64	0.66	0.64
TF-IDF	0.55	0.55	0.55	0.55

Table 2: Multilayer Perceptron Classifier

Despite bag of words does not take semantic meaning in consideration, it brings the best results with both classifiers. So we choose the Multilayer perceptron as classification method.

5 Stage 4: Visualization

Finally, applying the classifier to our dataset, we obatined the predicted sentiment on tweets. Starting from this, we computed a dayly mean of the sentiment, considering positive= 1, neutral= 0 and negative= -1. In this way, we can see the trend of the sentiment about the candidates, shown in figure 7. The graphs show a similar sentiment trend for both candidates, until the beginning of November, where there is an increasing of positive attitude towards Biden, infact the top mean score for Biden is 0.06, meanwhile for Trump stops at -0.02. Looking for news in newspapers, we can see what happened the day before the peaks occur, noticing that the 27th of October, the day near the negative peak for Trump, Twitter censured Trump's twitter post,meanwhile at the beginning of November both had an increasing in positive sentiment and it can be due to the vote count in Georgia.

Thanks to geo-spatial attributes, and using a visualization software, we could also visualize the placement of the tweets in different days. Since we can not report the interactive image, we show a capture of it in the days where the positive peak (for Biden) and negative peak (for Trump) occur. The size of the balls determine the number of tweets of the selected sentiment, so negative for Trump and positive for Biden.

We also tried to clusterize the data with respect to likes, but the algorithm divides tweets per continents, so it just keep into consideration the geo-spatial data.

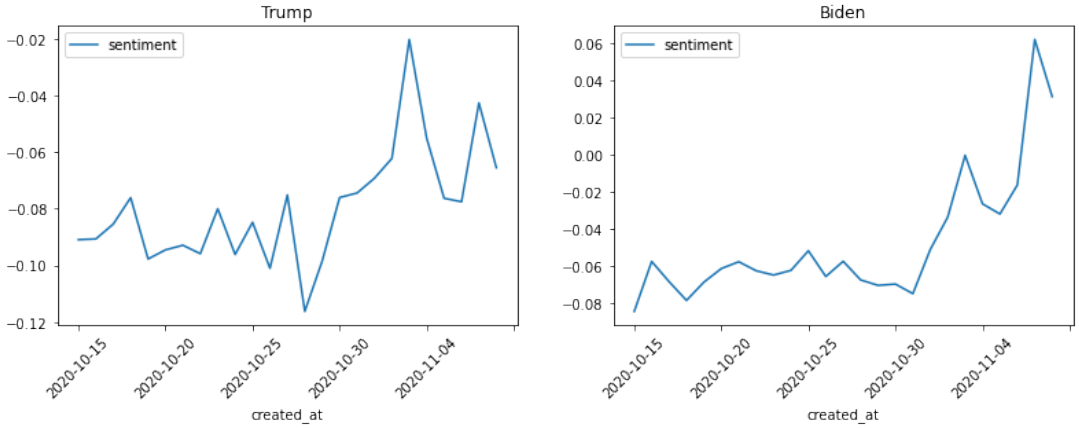

Figure 7: Sentiment trend

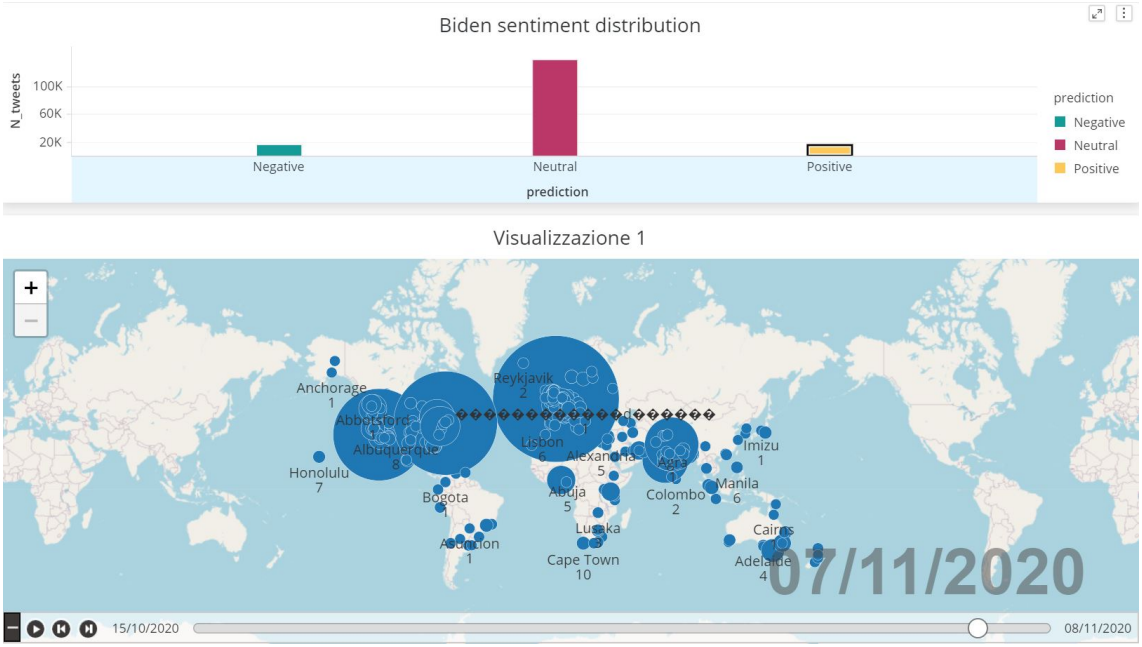

Figure 8

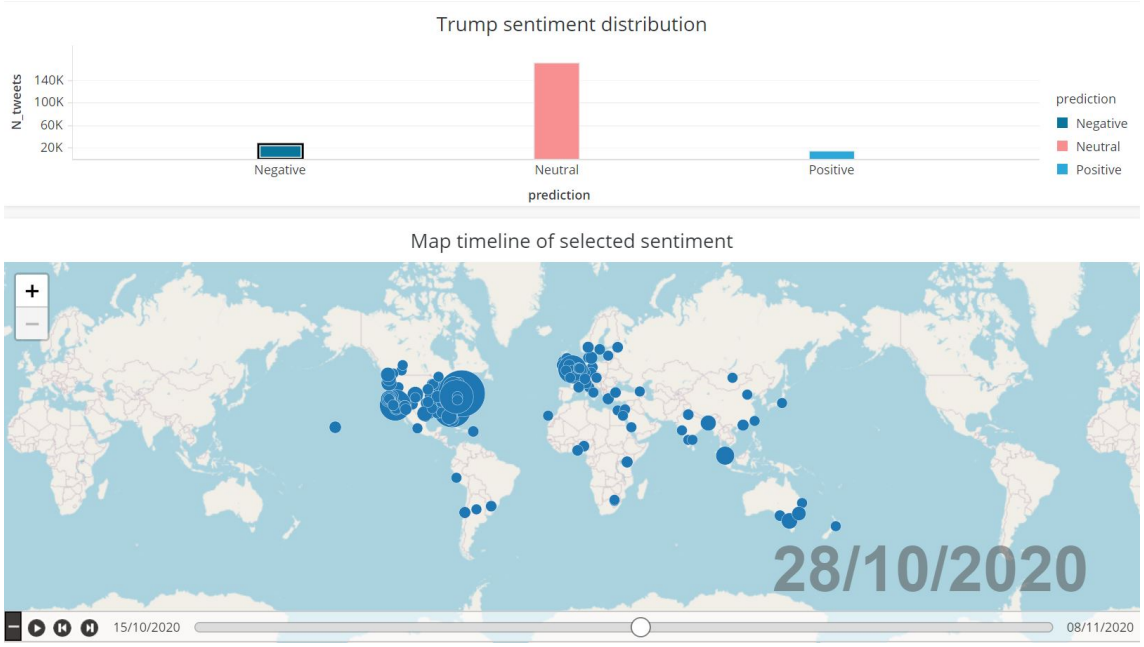
Trump sentiment distribution
N_tweets
140K
100K
60K
20K
Negative
Neutral
Positive
prediction
prediction
Negative
Neutral
Positive
Map timeline of selected sentiment
28/10/2020
15/10/2020
08/11/2020

Figure 9