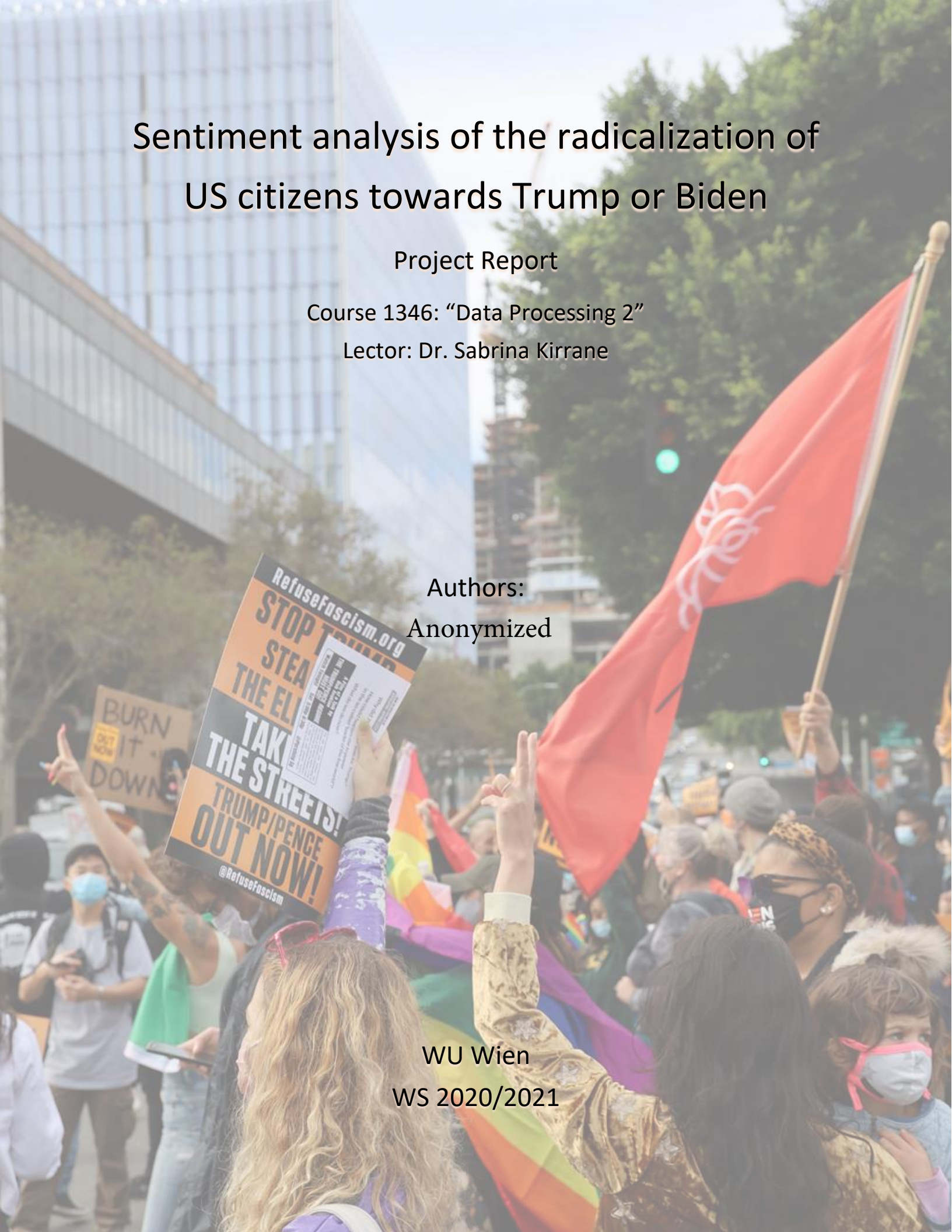
Sentiment analysis of the radicalization of US citizens towards Trump or Biden

Project Report

Course 1346: “Data Processing 2”

Lector: Dr. Sabrina Kirrane



Authors:

Anonymized

WU Wien

WS 2020/2021

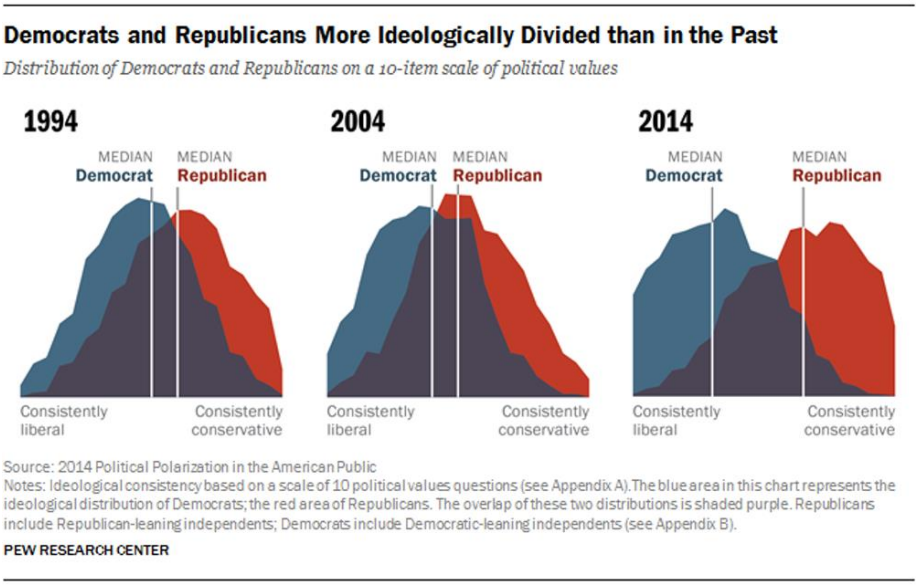
Table of Contents

Table of Figures

1. Project Overview

1.1. High level description of the big data project

Our project is connected to the current extensive political polarization of US citizens with its large, separate clusters of the population, that endorse ideologically consistent stances across all issues, and love their own party while loathing the other. People do not just fight over their own political view and disagree with each other, but they also refuse to live in the same neighborhood, to send their kids to the same school and to work in the same office as people who have different political beliefs. This antagonism is the reason for higher violence rate, tension between different groups, conformism, and fading fate in the government. (ZAID JILANI, 2019) Only in the past two decades, the percentage of Americans who consistently hold liberal or conservative beliefs, rather than a mix of the two, which is the case for most people, doubled from 10 percent to over 20. (LEE DE-WIT, 2019)



Many studies show that social media is to a big extent responsible for this worrying trend. People fall into filtered bubbles and only see news and posts that confirm their own opinion, that convince them further in the adequateness of their distorted reality. (Mims, 2020) Another problem is that politicians understand that the emotional more appealing messages work best and wake the interest of people, so they use this power and write radicalizing, more influential posts that further expand the gap between the two parties. (Paulus F.M., 2019)

Since this situation has a huge influence on one of the most developed countries in the world that has enormous impact on international affairs of all kinds, it is only natural for us to be interested in the topic. With this project we analyzed the level of polarization using the online support the faces of the two major political parties get, namely tweets and Reddit posts regarding Joe Biden and Donald Trump. We observed the level of radicalisation towards the two politicians and if some changes in the beliefs of Trump supporters occurred after Biden's inauguration and the storming of the Capitol.

<u>We wanted to get an answer to the following questions:</u>
"What amount of the tweets/posts are highly polarized?"
"Which politician is mentioned more on social media?"
"Are the two politicians mentioned more in a negative or in a positive context?"
"Did the recent political events in the USA have some influence on the sentiment of US citizens?"
"Are the sentiments towards Biden and Trump different on Twitter compared to Reddit?"

“Which are the words mostly mentioned in posts or tweets about Trump and Biden?”

1.2. Data Sources

We used two main data sources – the Twitter API and the Reddit APIs - Praw and Pushshift, to be able to make a comparison between both and to arrive at a more insightful result with stronger evidence. We also tried to use the NewsAPI from https://newsapi.org/ for an additional perspective, but the API we found didn’t return the whole content – only the first 200 characters, so we sticked to the other two that gave us more than enough data for further analysis.

1.2.1. Twitter API

The foundation of our project is the Twitter API. The public Twitter API has an API for streaming that provides access to a high volume of tweets from all around the world with low latency. Twitter is a great source of data for social research both current and historical because it boasts 316 million monthly active users with 500 million tweets per day and makes the data easily available (Sayce, 2020). One downside were the many specific legal, ethical, and privacy issues that can arise when conducting research using Twitter data, but we explain their overcoming more detailed later in the documentation.

By the importing of the tweets we categorized the information per tweet in the columns: tweet_id, tweet_created_at, tweet_text, user_name, user_created_at, profile_bio, followers, user_location. Out of all our columns the tweet_text is the most important one because to find out more about the general sentiment of the twitter user we can use Natural Language Processing on the specific tweet texts. The tweet_created_at column helps us to categorize the tweets into their time of creation to do further comparisons of the sentiments on specific days or before and after specific events. Using the user_location we could categorize and compare the sentiments of specific regions for example the states in the US. We also only scraped tweets in English because the used steps of natural language processing are best applicable in English. We were able to extract around 163.000 tweets in total, around 12.000 from the day of Biden’s inauguration.

In order to compare different dates from the last couple of weeks we needed to access tweets which are older than 7 days and therefore could not be streamed. We decided to use tweepy to access the historical data. Unfortunately, only one of our developer accounts was able to access historical data and could only get around 3.100 tweets for the date 06/01/2021, because the other ones couldn’t create a dev environment for full archive sandbox.

1.2.2. Reddit API – PRAW and Pushshift

In general, we wanted one more data source to get the bigger picture about how the people feel towards Trump and Biden and to see, if different social networks appeal to have different sentiments towards topics. We chose reddit because their data is easily accessible, all free and open to the public, we find it to be a decent source for news and a great source to learn more about specific topics, because it is built for discussions and used by millions of people. It is also one of the top 10 websites in the world. (Weinstein, 2019). We could extract around 84.000, around 36.000 from the capitol storming day, around 10.000 from the day of the inauguration.

We extracted the data in the columns id, author, comment_txt, score, pinned and created_at. Similar to the twitter data we are mostly interested in the comment_txt as it contains the text the user wrote. We can apply the exact same Natural Language Processing algorithms on the reddit posts to get a better understanding of the users' sentiment towards Trump or Biden.

1.3. Qualification as a Big Data Project

In general, a project is classified as a Big Data Project if it matches a specific number of criteria called the V´s. Different research enables a different amount of V´s but usually we can classify the minimum of 4 V´s and the maximum of 7 V´s. They 7 V´s are Volume, Variety, Veracity, Velocity, Visualisation, Variability and Value. When at least 4 out of these 7 factors are covered in a project, the project can be classified as a Big Data project.

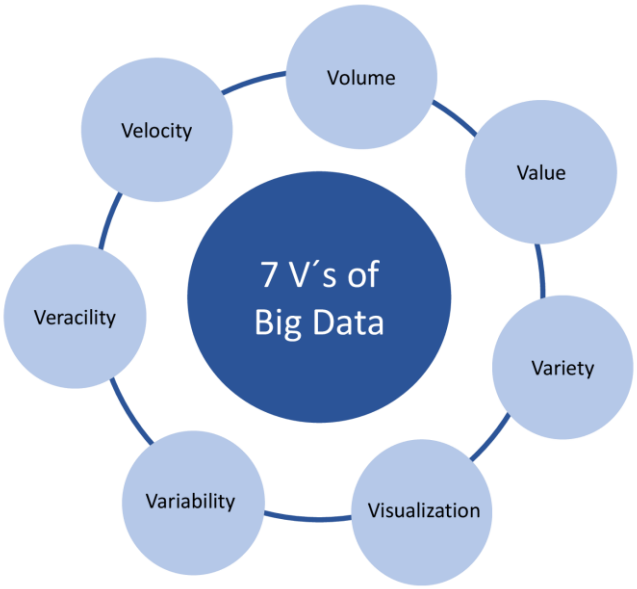


Figure 1 - 7 Vs of Big Data

Our project classifies as a Big Data project because it matches all 7 of the V´s. The first V is the value. Our project is giving a value to any person who is interested in the current sentiment around Trump and Biden but especially to people who work in politics and might be curious about the actual situation or people who work towards reducing the anger between political groups. The second V is the Volume. To get a realistic and representative understanding we must use huge quantities of tweets and posts, therefore we match the criteria of having a big volume of data. Additionally, to be able to gather the information of tweets and posts we need to stream them with API´s. Therefore, we match the criteria of velocity. Because of our fast changing and developing environment the tweets and therefore the sentiment around the tweets is always in a consistent change and movement which matches the criteria of having variability in the data.

To not follow only one direction in our project we compared the sentiment of twitter users with the sentiment of reddit users. Therefore, we match the criteria of having variety. Additionally, we tried to use two different libraries for the sentiment analysis TextBlob and Vader and therefore our project includes a different variety of resources. To show and share our results we use different visualisation techniques and approaches. We include word clouds and bar charts as well as a scattertext to present our results in a clear and understandable way. Therefore, we match the criteria of visualisation.

1.4. Proposed Solution

Our project is conducted in Python, using mostly Apache Spark, Apache Arrow and Pandas. We prototyped a lot of the things with pandas and then converted them to spark, since we all worked with it for a first time. In the following we explain its architecture, libraries and algorithms used in more detail.

1.4.1. Architecture

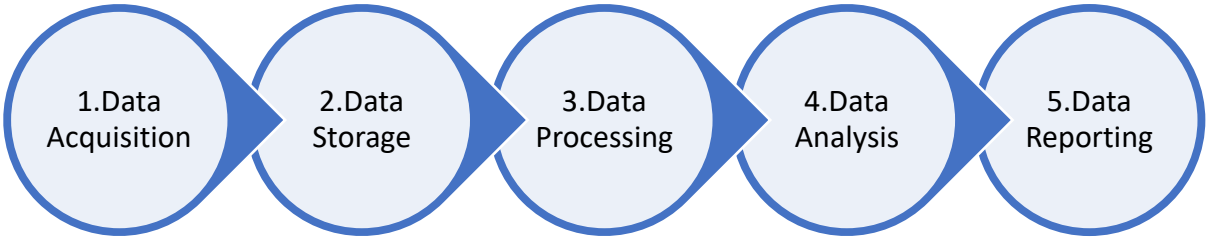


Figure 2 Architecture of the Big Data project

For our architecture we started with categorizing our project in its main parts. We divided it into data acquisition, data storage, data processing, data analysis and data reporting. During our first meeting we discussed the structure and all our next steps, so we can split the work appropriately. The first step was the data extraction from Twitter, Reddit, and the unsuccessful trial to get data also from the NewsAPI. We stored the data for continuous use using the sqlite3 library. Later, after the scraping, we exported two separate csv files – one for the Twitter and one for the Reddit data. Afterwards we did some research regarding other useful for our case libraries and how could we clean and anonymize the data, so we don't overstep some legal and ethical boundaries. Then the sentiment analysis of the four types of data followed, using TextBlob and Vader Sentiment. In the next steps we were able to make a lot of comparisons – between Reddit and Twitter data, present and historical data, data about Trump versus data about Biden etc. Some nice visualizations followed using four different libraries, namely matplotlib, seaborn, wordcloud and scattertext. After this part was done, we wanted to apply some machine learning methods for classification. Therefore, we tried to create a Machine Learning algorithm which can classify a tweet being a twitter tweet or a reddit post. The data processing and analysis were mainly carried out using Apache Spark. Lastly, we applied some more interesting visualization techniques and used them for the reporting of the most important insights from the analysis.

1.4.2. Libraries and Packages

We used different kinds of libraries to extract, process, analyze and visualize our data. During the class we got to know some libraries and additionally we researched on the internet trying to find some more fitting our needs for the project. All libraries and their usage for the project are listed below.

Figure 3 Libraries

1.4.3. Algorithms

We started with the notebook called data-extration.ipynb that has four different parts for each kind of data we were interested in – Twitter/Reddit Stream Data and Twitter/Reddit historical data, needed for the comparison of the sentiment after the storming of the Capitol and after Biden's inauguration day.

1.4.3.1. Data Extraction

Twitter Present Data

For the twitter streaming we used tweepy to access the Twitter API and its stream listener, which is essential for the real time streaming, and saved the tweets directly into a json file in our SQLite database. After we scraped the data, we exported a csv file so the data could be later retrieved as a spark RDD or a spark data frame. To get the best results and no duplicates we only used tweets and no retweets and only from profiles located in the United States. To fetch the data fitting all of those mentioned needs we wrote the following class method (only a part of it visible):

```
class OurStreamListener(tweepy.StreamListener):


    def on_status(self, status):
        api_response = status._json
        # we take only tweets that are not retweets or quote tweets and come from USA -> ideally we remove tweets that are replies so we get 'pure opinion' as first tweet
        if status.user.location:
            if 'USA' in status.user.location and not hasattr(status, 'retweeted_status') and status.is_quote_status == False:
```

Furthermore, we only used tweets mentioning "trump" or "biden":

```
def main():
    auth = tweepy.OAuthHandler(tw_client_key, tw_client_secret)
    auth.set_access_token(tw_access_key, tw_access_secret)

    stream_listener = OurStreamListener()
    OurStream = tweepy.Stream(auth=auth, listener=stream_listener, tweet_mode='extended')
    OurStream.filter(track=['trump', 'biden'])
```

Twitter Historical Data

Since only one member of our group who had his own developer account was able to make an environment for full archive sandbox, we had limited access to the maximum of 3.100 historical twitter tweets. To access the historical tweets, we used tweepy again and saved the tweets in our SQLite database in the same file as the real time tweets. We chose the date 06.01.2021 as it was the date of the Capitol storming and used it to later on split the data into historic and present one (taking only rows with this exact data). We only extracted tweets in English and either with the word "trump" or the word "biden". Here is a screenshot of the query of the historical tweets:

```
history = tweepy.Cursor(api.search_full_archive,
                        environment_name='dev',
                        query='trump OR biden lang:en', # -is:retweet cant exclude retweets, premium feature
                        fromDate='202101061800',
                        toDate='202101061900').items(100) # 5000 max out
```

Reddit Present & Historical Data

Our second data source is Reddit. To get the present data we used PRAW, acronym for "Python Reddit API Wrapper", because it is easy to use and again gives us the posts in a json format. We saved those to our SQLite database and exported a csv file for the Reddit data as well. We only used comments in the subreddit "politics":

```
def main():
    for comment in reddit.subreddit('politics').stream.comments():
```

Furthermore, we also wanted to obtain historical data from Reddit as well and we used Pushshift for that, because of the limitations of PRAW when it comes to extracting submissions between specific dates. Pushshift is a big-data storage space created by Jason Baumgartner. Whenever a reddit post is posted it is directly forwarded and stored in Pushshift (https://www.reddit.com/r/pushshift/, n.d.). Therefore, we can access the Pushshift database to collect all posts from the subsection "politics" on the date 06/01/2021:

```
subreddit = 'politics'
size = 100
capitol_day = datetime.datetime.strptime('06/01/2021',"%d/%m/%Y")
delta_limit_up = 90
delta_limit_low = 89
```

1.4.3.2. Data Pre-Processing

In the next step we clean our data and anonymize it in order to minimize the potential harm, because the data we use is sensitive, since it's connected to the political beliefs of the users. To do that we firstly remove certain stop words we do not need for the analysis and afterwards we write some functions that get rid of @mentions, #hashtags, and links in tweet text and extract them into separate columns and anonymize users. We saved usernames so we can try and analyze if there are some propaganda bots later. Same username will always have same hash. Afterwards we turn those functions into UDFs, so they work on spark data frames. Here is an overlook over a table with already anonymized users:

tweet_id	tweet_created_at	tweet_text	user_name	user_created_at	profile_bio	followers	user_location	links	tags
1350566551343288323	2021-01-16 22:12:12	your Tweets are g...	587e4499baa818c84...	2019-07-13 15:26:20	BS Computer Engin...	30	Florida, USA	[]	[evidence, proof,...]
1350566551942881280	2021-01-16 22:12:12	Their no match fo...	1349a52b6d6f27427...	2019-02-24 21:59:48	Patriot	30	California, USA	[]	[]

1.4.3.3. Sentiment analysis

Before we start with our in-depth analysis, we had to import the SparkSession module and connect our database to Spark to start the SparkSession. Additionally, we read all of our data into the SparkSessison object.

[79]:
```
# Build the SparkSession
spark = SparkSession.builder \
    .master("local") \
    .appName("final_project") \
    .getOrCreate()
```

To be able to explore the level of polarization of US citizens, we needed to conduct a sentiment analysis. We wanted to use two pre-built sentiment classifiers (TextBlob and Vader) to make a comparison and get

a feeling if the results from both are realistic and give us a strong enough evidence. Both of those classifiers are built on top of the Natural Language Processing library for Python. TextBlob is believed to be more useful, when it comes to analysis of texts in formal language, while Vader is better with reading emotions from emojis, slang etc. (White, 2020), which probably makes it the right choice for analysis when it comes to social media posts, but we still wanted to try both ourselves, compare the results and manually check how adequate they both classify the tweets and the Reddit posts.

TextBlob is a library for processing textual data and is built on NLTK and pattern. It has a variety of different features for example part-of-speech tagging and word & phrase frequencies, but we will mainly use it for sentiment analysis. It provides us with the polarity and the subjectivity of the text.

```
df_blob = df6.withColumn(
    'blob_rated', # chained when
    F.when((F.col("text_blob") < 0.0), 'Negative')\
    .when((F.col("text_blob") > 0.0) , 'Positive')\
    .otherwise('Neutral')
)
```

To present our results in a logic and easily understandable way we created different charts with the help of the library seaborn. In the following charts the color green means the sentiment positive, the color blue the sentiment neutral and the color red the sentiment negative.

```
def plot_difference(title, df1, df2, column, file_to_save, title_one, title_two):
    order = ['Positive', 'Neutral', 'Negative']
    palette = ['#00ff00', '#0000ff','#ff0000'] # green-positive, blue-neutral, red-negative

    fig, (ax1, ax2) = plt.subplots(1, 2, sharey=True, figsize=(15,5))
    fig.suptitle(title, fontsize=16)
    sns.set_style("dark")
    sns.despine(left=True, bottom=True)
    sns.countplot(ax=ax1,
                  x=column,
                  data=df1,
                  palette=palette,
                  order=order)

    sns.countplot(ax=ax2,
                  x=column,
                  data=df2,
                  palette=palette,
                  order=order)

    ax1.set_title(title_one)
    ax1.set_ylabel('')
    ax1.set_xlabel('')

    ax2.set_title(title_two)
    ax2.set_ylabel('')
    ax2.set_xlabel('')


    fig.show()
    plt.savefig('./data/graphs/' + file_to_save)
```

Bar Charts – TextBlob

The next bar charts represent a comparison between present data and data from Biden's inauguration day gathered with the help of the Twitter API on the exact date. The second one is with adjusted parameters, because those within range -0.2 and +0.2 are counted like neutral (not only the ones that equal zero). We adjusted it this way to be able to inspect more polarized opinions.

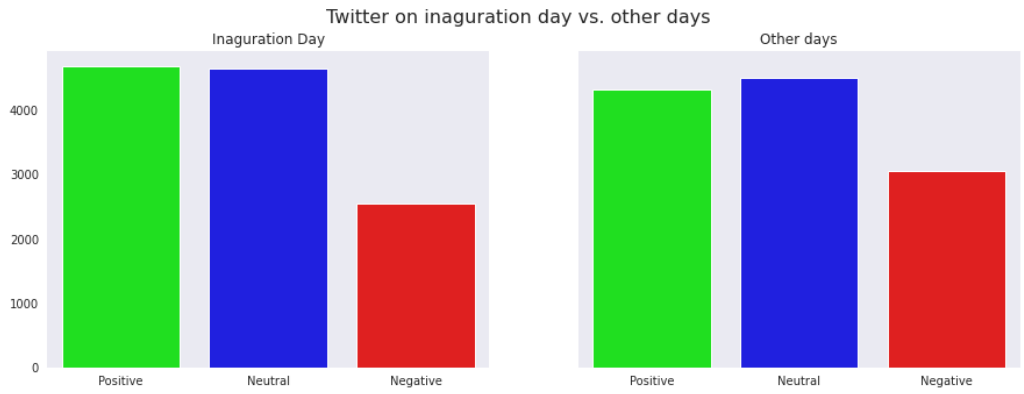


Figure 4 TextBlob Analysis Twitter, Neutral = 0.0

The first bar chart shows that on the Inauguration day the negative sentiment towards the candidates is lower and the positive higher. Not all of the negative ones turned to be positive ones though, because the number of neutrals has also increased.

From the second one it is easy to see that the majority of the positive and the negative ones were not extremely polarized, because after the parameter's adjustment the number of neutral tweets has increased dramatically. The difference between the sentiment of the US citizens towards the two party leaders in the two different time frames is the same as in the first bar chart.

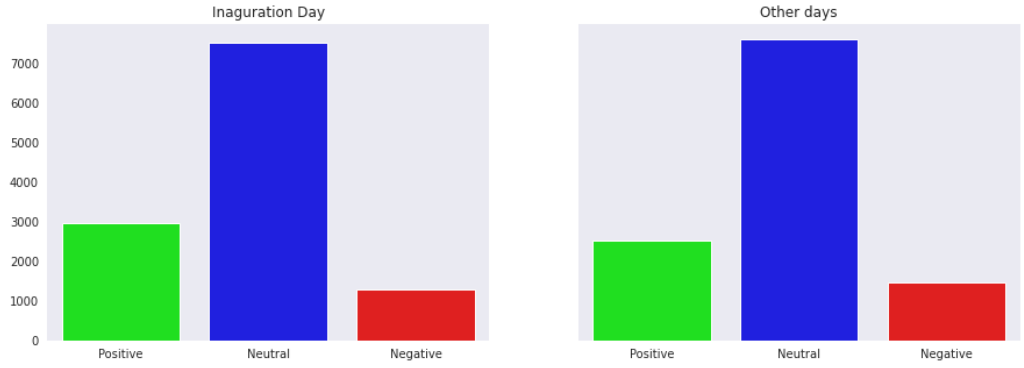


Figure 5 TextBlob Analysis Twitter, Neutral -0.2 to +0.2

The following bar chart shows a visualization of reddit posts on the 06.01.2021 and the days afterwards. We used the TextBlob library to categorize the posts in positive, neutral, and negative. The plots show that during normal days the sentiment of tweets from the subsection politics are mostly positive, followed by neutral. In comparison, the tweets on the day of the Capitol Storming were a lot less positive. Most of them were neutral and a little more positive than negative. One interesting insight is that people did not get more negative after the storming of the Capitol, they just got more neutral.

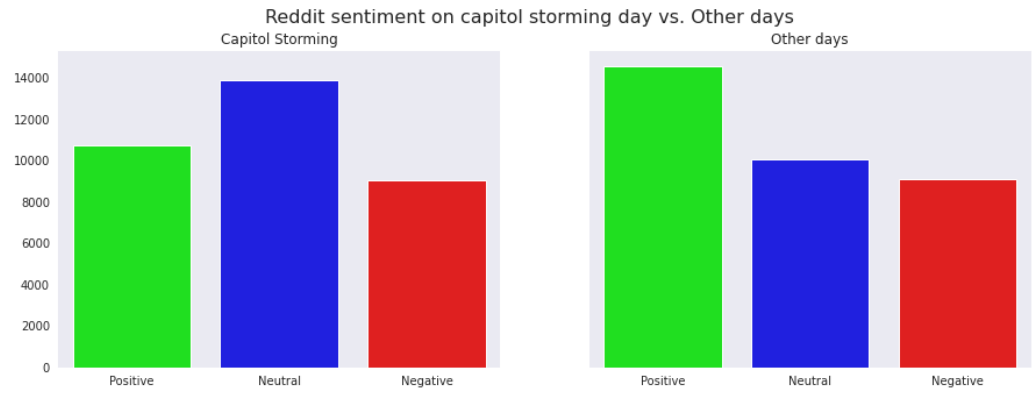


Figure 6 TextBlob Analysis Reddit

Comparison of the sentiments towards the president and the former president

Furthermore, we wanted to see if people have different sentiments about Biden and Trump. We separated the streamed tweets in tweets containing the word “trump” and tweets containing the word “biden”. The visualization shows that in general both receive more positive sentiments than neutral or negative. But there is a higher number of tweets with a negative sentiment in tweets about Trump than in tweets about Biden.

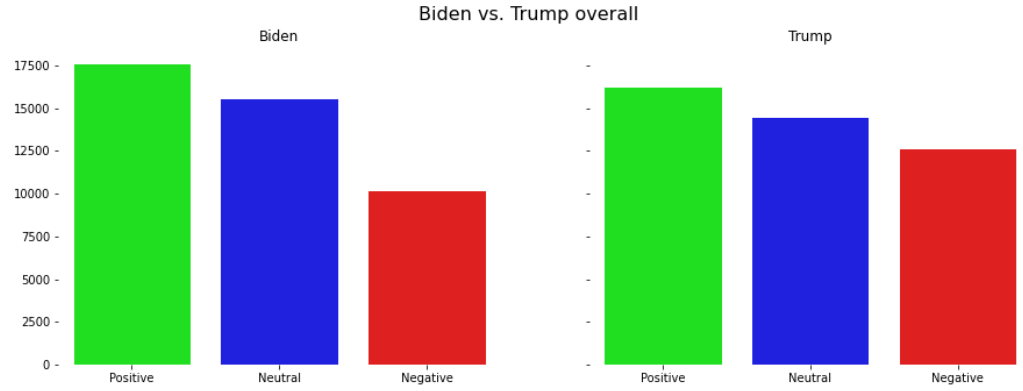


Figure 7 Biden vs. Trump Overall

We also made a comparison between the sentiment towards the two party leaders on certain eventful dates. The first one is the storming of the Capitol on 06.01.2021, a more Trump connected event. Since there is a limit on the historical data access, this comparison is based on only 3.100 tweets.

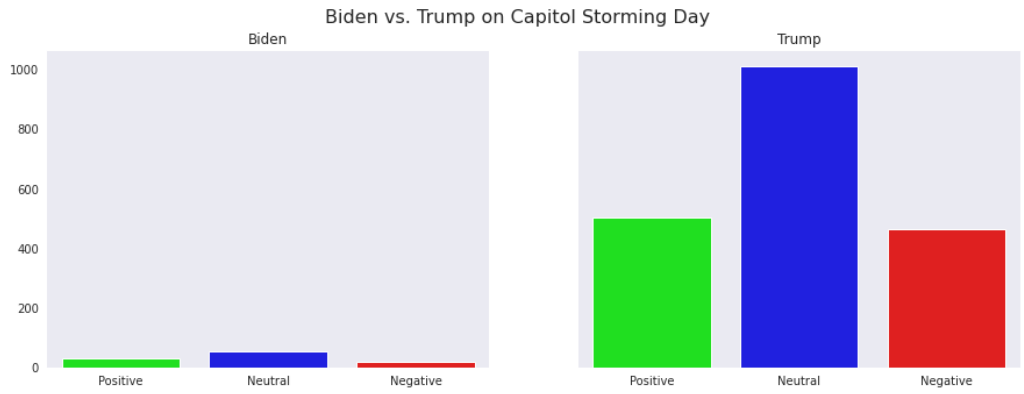


Figure 8 Biden vs. Trump - Capitol storming

It is obvious that the majority of people tweeted about Trump, but it is also interesting to notice that the sentiment towards Trump was mostly neutral and more positive than negative.

We wanted to make the same comparison for a Biden connected event, namely the Inauguration day. It is easy to notice that Biden has received way lower negative comments than Trump, but people who did not say something negative about Biden rather said something neutral about him, not positive, since the positive rates of both politicians are almost equal.

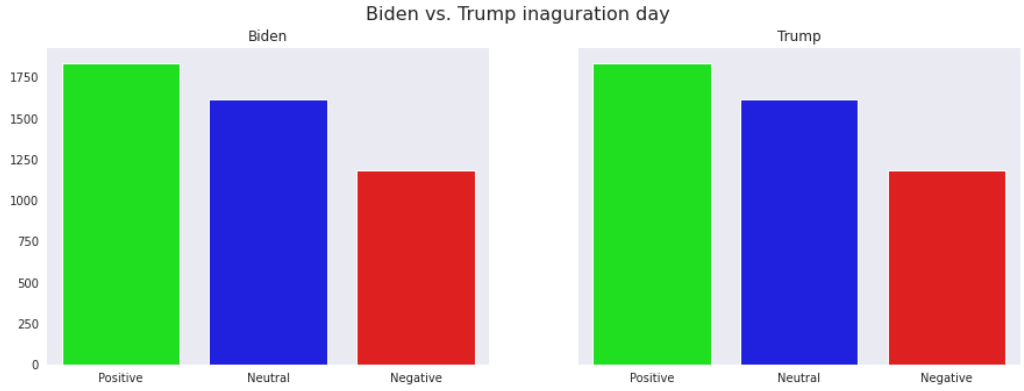


Figure 9 Biden vs. Trump - Inauguration day

Bar charts – Vader

VADER, an abbreviation for Valence Aware Dictionary and Sentiment Reasoner, is a lexicon and rule-based sentiment analysis tool. It is used to quantify how much of positive or negative emotion the text has and the intensity of emotion. It has the following categories: positive, negative, neutral, and compound, compound being the sum of the first three scores, normalized between -1 (extreme negative) and +1 (extreme positive).

The following bar chart shows the change of sentiment on the eventful 6th January this year.

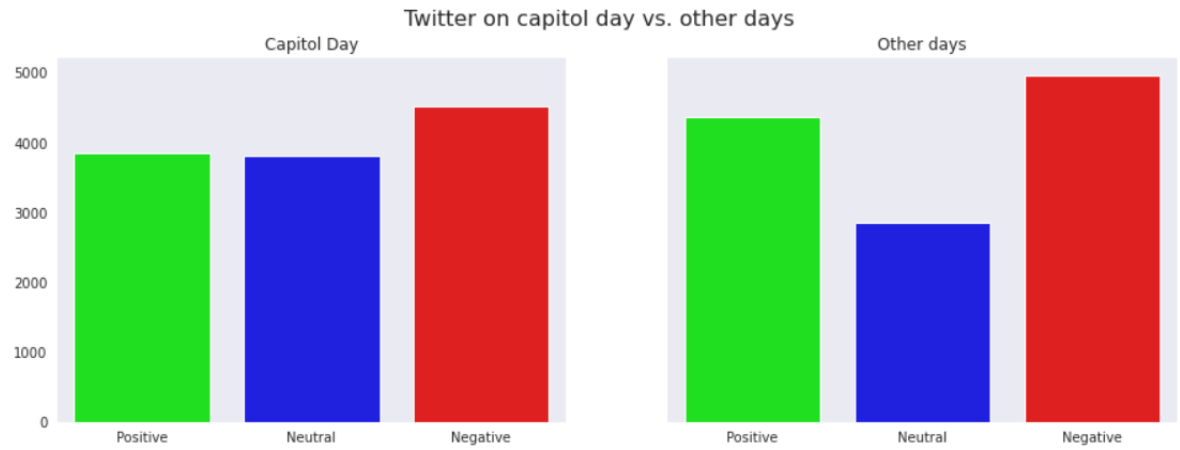


Figure 10 Twitter on Capitol day vs. other days - VADER

According to the plot people posted more negative political tweets on other days than on the day of the storming of the Capitol, which was pretty surprising for us. They rather got more neutral than negative on the eventful day.

The next diagram analyzes the second political event – Biden's Inauguration day.

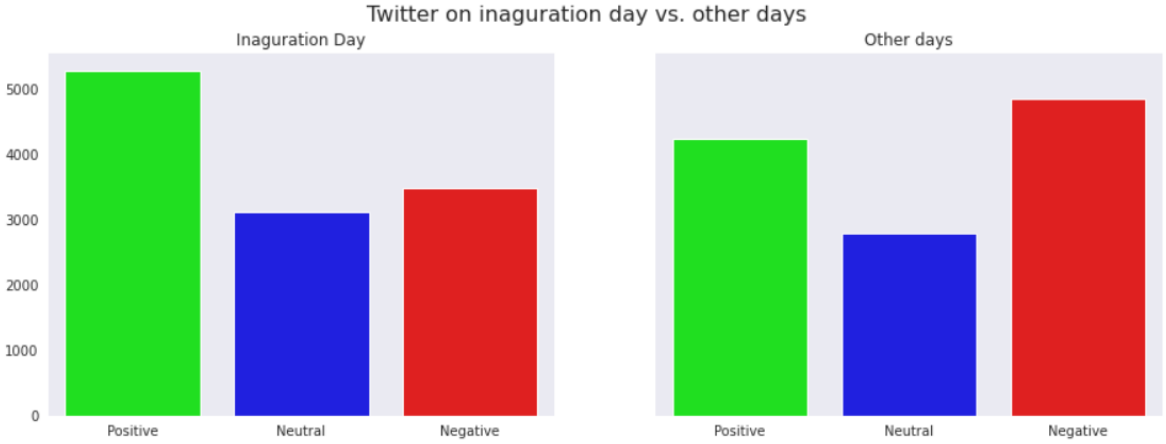


Figure 11 Twitter on Inauguration day vs. other days - VADER

One interesting insight here is the obvious reduction of negative posts and the increase of the positive ones, which could mean acceptance of the new president. This was also the expected result.

Comparison between TextBlob and VaderSentiment

The obvious difference between the two tools is in the number of neutral and negative tweets, since Vader categorizes a bigger percentage of the tweets as negative. The trends are overall the same – more positivity during the Inauguration day, less positivity during the Capitol storming, but they are with a different positive/neutral/negative ratio. After some manual checks, we noticed that the reason behind that is the ability of Vader to read the sentiment of emojis and slangs, while Textblob considered them all to be neutral.

Word Clouds

After analyzing the different sentiments from our different data sources as well as the different sentiments from different dates, we wanted to dig deeper and see what exactly people wrote. Therefore, we decided to look at the frequency of the words in the specific tweets and posts.

We used word clouds for a visual representation of the most mentioned words, since in our opinion this is a great way to get insights on trends and patterns from a huge dataset.

The following word cloud shows that in Reddit, according to the data we personally retrieved, Trump and his party are mentioned way more often than Biden and the democrats, since a couple of the words with the biggest font size are “Trump” and “Republican”.

In the 80k of tweets we analyzed, the word Trump appeared 9.123 times ant the word Biden appeared only 2.986 times.

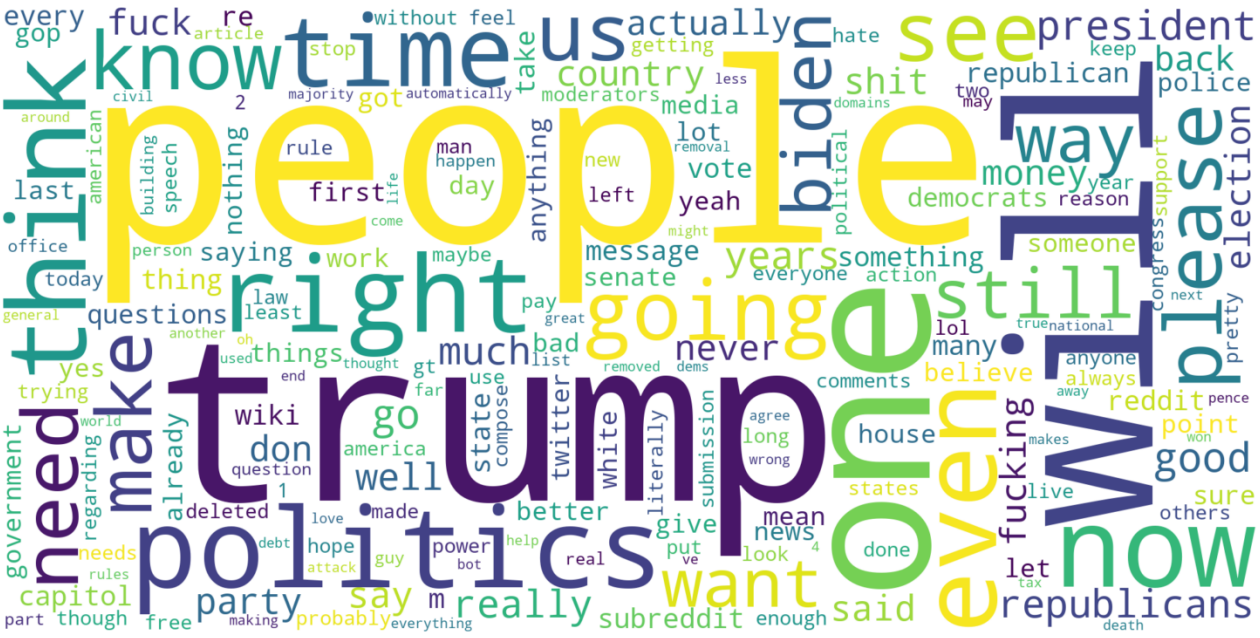


Figure 12, Word cloud - Reddit

The next word cloud shows the frequency of the words used in the tweets. We can see that the word Biden and Joe Biden appears a lot more than it did in the reddit word cloud. Nevertheless, the word Trump is still a bit bigger. The word people which is the biggest word in the reddit word cloud is a lot smaller in here.

Generally, it makes sense that the words Trump and Biden appear to have the highest frequency because the twitter tweets were filtered with the both words Trump and Biden.

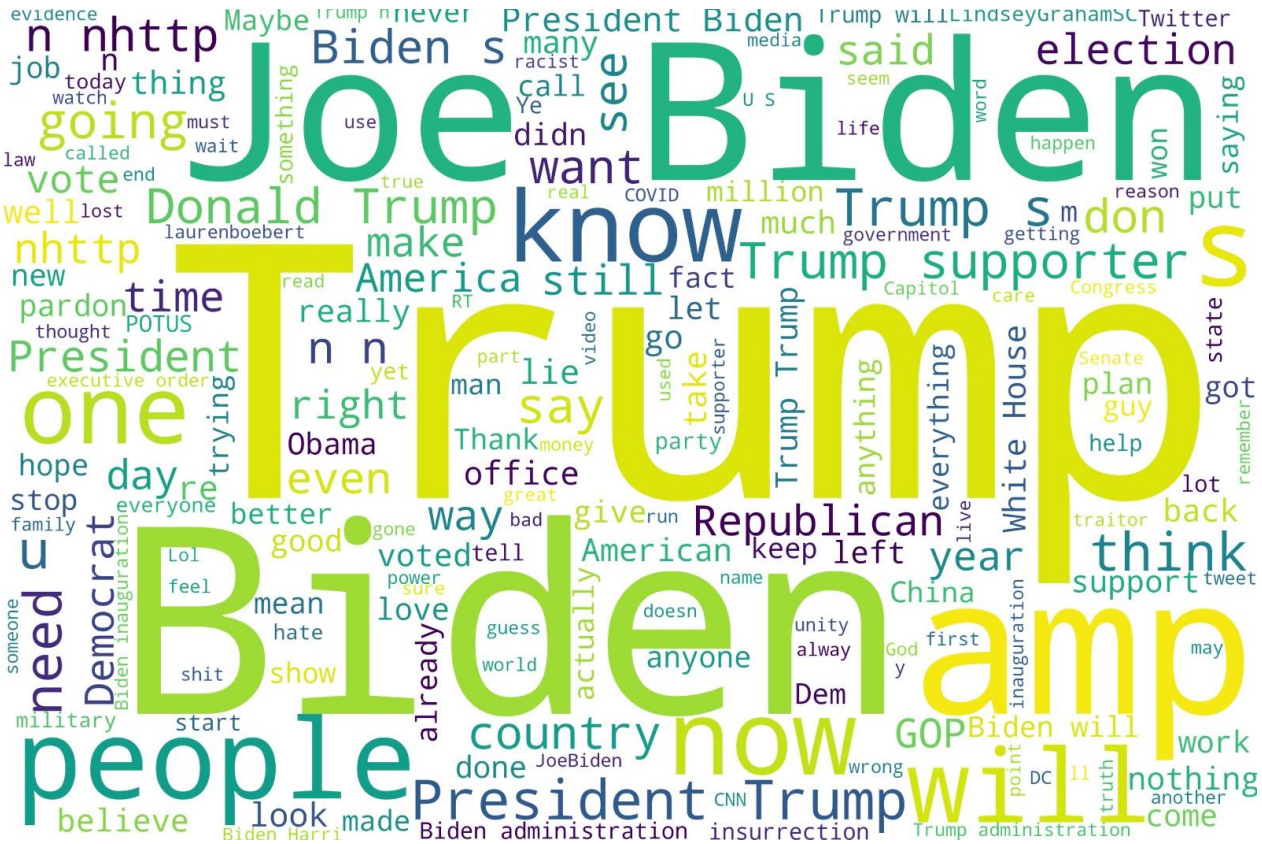


Figure 13 Word cloud - Twitter

Scattertext

Furthermore, we used the scattertext library to visualize not only the most frequently used words but also to see their categorization, since scattertext considers the category-specific precision and the term frequency. While a term may appear frequently in both categories (Top Biden/Trump mentioned), the f-score determines whether the term is more characteristic of a category than others.

For the following figure we used only 1.000 tweets from people with more than 10.000 followers who are considered to be influencers. One of the key insights is the fact that Trump is mentioned almost 1.5 times more than Biden. It is also important to mention that words that were frequently mentioned in the Trump connected tweets have a more negative sentiment, for example “impeachment”, “trial”, “arrested”, “insurrection”, while the tweets discussing Biden mostly contain words with a neutral sentiment, e.g. “ executive”, “inauguration”, “elect”, “day”, which is easy to explain having in mind his recent Inauguration.

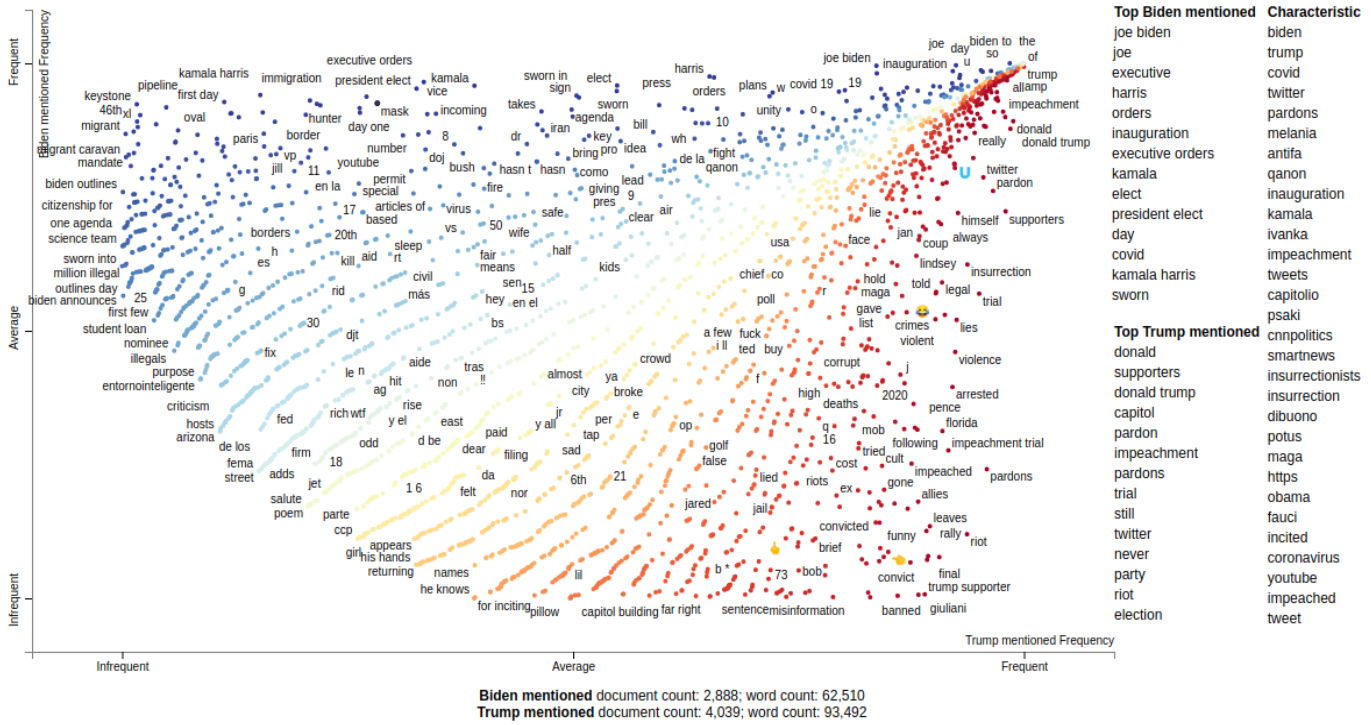


Figure 14 Scattertext

1.4.3.4. Conclusion

After analyzing two different data sources, different days in January and February and even attempting to compare different libraries we would like to make a little conclusion.

The most obvious fact is that the word Trump appeared a lot more in general and especially on the day of the capitol storming. That didn´t really surprise us because we already experienced to have seen a lot more newspaper article about Trump than any other president. The huge quantities on content could either mean Trump has a lot followers liking what he does, or it could mean he has a lot of followers who criticize him.

Our sentiment analyses showed that there are for sure both, people who write positive about him and people how write negative about him. We could see that in our streamed twitter tweets more people wrote positive and neutral about Trump than negative. Even though we thought we heard a lot of criticism about Trump in Europe on the day of the capitol storming the sentiment towards Trump in the US was more positive towards him than negative.

After widening the neutral spectrum to see how many people have extremely strong opinions, we realized most of the tweets were not really polarized because the amount of neutral tweets increased immensely. We expected to have more stronger positive and stronger negative ones, but we still have to keep in mind that the libraries are not perfect and for example sarcasm is not detected all the time.

Comparing the twitter and reddit results we can see a bit of a difference in the general sentiments, but it is not a significant big difference. There are a little more positive tweets on Twitter containing the word

Trump or Biden than the posts in the subreddit politics. Therefore, we could conclude that people on reddit are either more critical or more negative about the both. But generally, it is not possible to see a huge difference and we would have to look at bigger amounts of data and filter for the exact same words to make more detailed comparisons.

We wanted to compare the results of the two libraries TextBlob and Vader because as already shown their results differ. The Vader library marks less tweets as neutral and more as negative compared to the TextBlob library. According to our experience Vader was in fact better for analysis of social media text, but also way slower to execute, especially in Jupyter. Therefore, most of our insights are based on the TextBlob analysis. It is important to mention that both tools have difficulties with reading the sentiment of for example sarcastic texts and rhetorical use of language, but combined they are more than enough to gather some interesting conclusions.

1.4.4. Classification of Reddit vs Twitter posts

Our goal was to give our algorithm a text and classify whether that text came from Twitter or from Reddit. This is not a 100% useful application, but it gives us a great chance to combine our data from Twitter and Reddit and understand how NLP classification can work. Our data is in so far labeled, as we know which data came from which source.

We started with labeling all twitter posts with 1.0 and all reddit posts with 0.0 and combine all of them in one data frame. After applying a cleaning function which removes links, hashtags and mentioning on our data we started to split our data. We split our data in a 60-40 split, with 60% for the training set and 40% for testing.

Naïve Bayes

As next step, we started with the machine learning and defined all the necessary components for Naïve Bayes: HashingTF, IDF, Tokenizer, Pipeline, Naïve Bayes, MulticlassClassificationEvaluator, ParamGridBuilder and CrossValidator Modules.

```
# Building our pipeline
tokenizer = Tokenizer(inputCol="text", outputCol="words")
hashing_tf = HashingTF(inputCol=tokenizer.getOutputCol(), outputCol="features")
idf = IDF(minDocFreq=3, inputCol="features", outputCol="idf")
nb = NaiveBayes()

pipeline = Pipeline(stages=[tokenizer,
                            hashing_tf,
                            idf,
                            nb])


paramGrid = ParamGridBuilder().addGrid(nb.smoothing, [0.0, 1.0]).build()

# Crossvalidation -> if we understood it right, with this approach we will train our model
cv = CrossValidator(estimator = pipeline,
                    estimatorParamMaps = paramGrid,
                    evaluator = MulticlassClassificationEvaluator(),
                    numFolds = 2
                   )
```

Additionally, we defined the pipeline that runs through the sequence of stages and build a parameter grid for the k-fold cross validation and performed the crossvalidation. Then we ran the model and observed the predictions.

```
# Running model
print('Model started')
cvModel = cv.fit(training_df)
print('Model has finished')
```

```
Model started
Model has finished
```

```
# Running model on test data
result = cvModel.transform(test_df)
result.select("text", "label", "prediction").show(10)
```

```
+--------------------+-----+----------+
|                text|label|prediction|
+--------------------+-----+----------+
|i may lose follow...|  1.0|       1.0|
|What a fucking sh...|  0.0|       1.0|
|Disagrees with th...|  1.0|       1.0|
|He is far right. ...|  0.0|       0.0|
|The end of Donald...|  1.0|       1.0|
|I'm frustrated an...|  1.0|       1.0|
|Yep, every time a...|  0.0|       1.0|
|Because we live i...|  0.0|       0.0|
|Why are these rat...|  0.0|       0.0|
|Just got back fro...|  0.0|       0.0|
+--------------------+-----+----------+
only showing top 10 rows
```

After running our model, the best model was selected, and the predictions were manually observed. We can already observe that it is not always correct but to understand the quality of our model we evaluate the performance.

To evaluate the performance of our model we checked the accuracy of our prediction with the below evaluators – Accuracy and F1.

```
# Evaluating our results
evaluator = MulticlassClassificationEvaluator(predictionCol="prediction")
print('Accuracy:', evaluator.evaluate(result, {evaluator.metricName: "accuracy"}))
```

```
Accuracy: 0.8235901171614156
```

```
print('f1: ', evaluator.evaluate(result, {evaluator.metricName: "f1"}))
```

```
f1:  0.8235615204227635
```

That means our algorithm has an accuracy of around 82.4% and an F1 score of 82.4%. The F1 score is the harmonic mean between precision and recall and the higher it is to 100% the more accurate is our model. This accuracy shows the models ability to detect whether a text is coming from twitter or reddit. This not perfect but shows that given our data, we can build a decent performing model. It is very important to state that this is not a super useful application and it is filled with errors and biases. Nevertheless, we could build a model that detect whether a text is coming from Twitter or Reddit, given our input data.

There are several caveats that need to be addressed. First, our data source for reddit was limited to a specific subreddit called r/politics which mainly list news about US politics rather than personal opinions.

On the contrary, Twitter posts mostly represent an opinion towards a certain political party. Therefore, one cannot believe that every post on Reddit or Twitter can now be classified with 85% accuracy – it hugely depends on the subreddit of the reddit data. Nevertheless r/politics is the most popular subreddit for US politics and therefore serves our cause for this project. Second, we briefly need to mention biases here as well. While our model just classifies between Twitter and Reddit posts and e.g. discrimination of people is not an issue here; our classification is biased towards voicing an opinion vs. stating facts

Logistic Regression

Additionally, we performed logistic regression to distinguish between reddit and twitter posts to observe the performance and validity of a second method. Again, we built our pipeline and built a training and test set.

```
# Building our pipeline
regexTokenizer = RegexTokenizer(inputCol="text", outputCol="words", pattern="\\W")
add_stopwords = ["http","https","amp","rt","t","c","the", "s", '&gt']
stopwordsRemover = StopWordsRemover(inputCol="words", outputCol="filtered").setStopWords(add_stopwords)
countVectors = CountVectorizer(inputCol="filtered", outputCol="features", vocabSize=10000, minDF=5)

pipeline = Pipeline(stages=[regexTokenizer,
                            stopwordsRemover,
                            countVectors])


# pushing dataframe through pipeline
pipelineFit = pipeline.fit(shuffle)
dataset = pipelineFit.transform(shuffle)

training_df = dataset.limit(130000)
test_df = dataset.subtract(training_df)


print('Training Dataset len:', training_df.count()) # we will use same data we already splitted
print('Test Dataset len:', test_df.count())
```

```
Training Dataset len: 130000
Test Dataset len: 30909
```

As next step the logistic regression model is build and trained without cross validation. Finally again, the accuracy of the model is observed.

```
# Building and training our model without cross validation
lr = LogisticRegression(maxIter=20, regParam=0.3, elasticNetParam=0)
lrModel = lr.fit(training_df)

# Running model on test df
predictions_df = lrModel.transform(test_df)
```

```
predictions_df.select('text', 'label','prediction').show(10)
```

```
+--------------------+-----+----------+
|                text|label|prediction|
+--------------------+-----+----------+
|So Is Trump Payin...|  1.0|       1.0|
|          their site|  0.0|       0.0|
|There’s a fuckin ...|  0.0|       0.0|
|Hahaha. I just mi...|  0.0|       0.0|
|He is the best. T...|  1.0|       1.0|
|Here’s to hoping ...|  0.0|       1.0|
|    So it is a coup?|  0.0|       0.0|
|Stop the steal gu...|  0.0|       0.0|
|Maybe they discus...|  1.0|       1.0|
|Noting a lack of ...|  1.0|       1.0|
+--------------------+-----+----------+
only showing top 10 rows
```

```
MulticlassClassificationEvaluator(predictionCol="prediction")
evaluator.evaluate(predictions_df)
```

```
0.8949339061729664
```

The accuracy now achieves an even higher score at around 89.5%. Again, we can already observe in our example predictions that the model is not always perfect. Normally, logistic regression is used when the output is Boolean as in our case. Same as in the Naïve Bayes case, our model is definitely biased given our data source r/politics. The possible solutions are discussed in the Unfair discrimination and biases section.

The general difference between these two methods is that logistic Regression makes a prediction for the probability using a direct functional form, whereas Naïve Bayes figures out how data was generated given the results. Overall, we see a better performing model at the logistic regression which might be attributed to the relatively big sample size.

2. Legal and Ethical Issues

2.1. Legal Guidelines

2.1.1. Licenses

- **Twitter API Developer Agreement**
  https://developer.twitter.com/en/developer-terms/agreement-and-policy
  In this Agreement there are also the following incorporated developer terms, that we had to agree to, namely the Developer Policy, API Restricted Use Rules, Twitter Rules, Display Requirements, Brand Resources, Automation Rules, Periscope Community Guidelines, and Periscope Trademark Guidelines. If there is a conflict between those and this Agreement, the rules of this Agreement are to follow. There are also some restrictions we had to consider that are connected to reverse

engineering, forbidden use of twitter marks, storing of location data and exceeding of the rate limits, no monitoring and measuring and attention to the security rules.

Regarding Twitter Content with personal data, we should keep it confidential and secure from unauthorized access by using safeguards. In the User Protection section, it is mentioned that political affiliation and beliefs are considered sensitive data and it is not allowed to display and distribute such data without Twitter's permission in writing, so we will not publish our finished project anywhere without it.

- **News API Terms of Service**
  https://newsapi.org/terms
  The API gives the users extensive rights and is free of charge for our project since it is non-commercial. There are two main licenses as part of those terms, on which we payed close attention to – the Use and the Data license. It was easy to follow the restrictions of the first one, since we didn't want to try anything violating any laws, we didn't have the incentive of reverse engineering of the software and to register more than one API key at one time. However, regarding the second one, we had to be careful with the handling of the personal data. The API does not contain any personal identifiable data of the end users, but it could contain unlicensed intellectual property data and we had to in every case keep the copyright, the ownership and other labels of the original sources as they are. According to the second one we are also not allowed to publicly release or disclose any data or usage statistics or other information regarding the data obtained. It is also worth mentioning that News API has an attribution clause in its terms which means that the user must indicate in the work that their technology is used.

- **Reddit's APIs Terms of Use (one of them being Pushshift API)**
  https://www.reddit.com/wiki/api-terms
  Under those terms falls not only the worldwide, non-exclusive, non-transferable, non-sublicensable, and revocable license Reddit grants as with to use its APIs, but also the Reddit User Agreement, Reddit API Rules and Privacy Policy, which we had to comply to. We had to be careful since those are being changed from time to time. The license grants us with the right to copy and display user's content, but we were not allowed to modify the User Content except to format it and needed to comply with any requirements or restrictions by the content's owners, which may include "all rights reserved" notices, Creative Commons licenses or other terms and conditions. According to the Privacy Policy we had to disclose how we collect, use, store, and disclose data collected from visitors. Furthermore, we were only permitted to access Reddit APIs using OAuth 2 and had the same restrictions as by the other two APIs mentioned above – no reverse engineering, illegal use, exceeding the limits provided, introduction of worms and viruses etc.

2.1.2. General Data Protection Regulation (GDPR)

As Legal Guidelines we followed not only the licenses of the APIs used, but also the General Data Protection Regulation. The most applicable rules for us were the principals of processing personal data, namely lawfulness, transparency and fairness, storage limitation, integrity, and confidentiality and also the processing of special categories of personal data section, where the political beliefs, which we deal with are to be found. Gladly, since tweets and Reddit posts are data made public by the data subject, it is not

prohibited of processing. It is also important to mention that according to the GDPR, data should not be used for purposes other than its original one without the consent of the content creator. (Union, 2018)

2.2. Ethical Guidelines

Code of Conduct for Professional Data Scientists

http://www.code-of-ethics.org/

Since our project is directly connected to sensitive personal data such as political beliefs, it was extremely important for us to find some source of ethical guidelines, because what is considered to be right or wrong in handling large volumes of data, collected from automated processes in academic research can often vary and the legal guidelines do not explicitly mention all of the challenges along the way and how should one proceed in some cases (example: reinforcement of human biases). In order to rightfully keep the main principles of Big Data Ethics, we researched further what is something we should take into account that is still not officially regulated but could be in some way harmful for the sources of our data – social media users. So, we decided to act having the Code of conduct for professional Data Scientists in mind. We are aware that more of those exist e.g. Data Science Code of Professional Conduct of the Data Science Association or the IEEE Code of Ethics, but the one we chose was in our view the most suitable one for our case. Our project is not a subject of all the principles of the Code of Conduct, but still it adheres to the lawfulness, transparency, objectivity and truth, because we did not act against the law in any point of our analysis, we protected the privacy and confidentiality of the users as much as we could, secured equality, removing the biases against sensitive attributes and were critical with our own findings, meaning we accepted unfavorable data and constantly questioned the result, searching for possible influencing factors that could affect it.

2.3. Concrete legal and ethical challenges and their overcoming

2.3.1. Minimizing harm and anonymizing data

According to the GDPR data should not be used for purposes other than its original one, which for the Twitter users for example is not to be included in a research, but rather just expressing an opinion on a social platform, unless stated otherwise. Gladly in Twitter's Terms and Conditions it is stated that with their consent, the users also provide the API with agreement for their tweets to occur in the scraping. However, it is often the case that customers don't read these Terms and Conditions, so we as team should do whatever we can to minimize the possible harm on the content creators. We will not only restrain from publishing our project, but we also tried our best to anonymize the data, so the users can not be identified by future dissemination audiences and judged for their expressed opinion, if the project were to be posted by a third party. We did it by covering the username and @handle of the poster and by removing certain words from the tweet that were not crucial for the sentiment analysis afterwards. However, according to the Twitter's User Development Policy any alternation of the original tweets by publication is considered a breach and would be in a contradiction with our anonymizing practice, but since we are not going to publish our results and findings we figured we can ignore this contradiction for now. (Ursula Garzcarek, 2019)

2.3.2. Lack of transparency

Another challenge was the lack of transparency, regarding the prediction if a tweet is positive or negative, because we used predefined libraries like TextBlob, and Vader and the results were often contradicting. To overcome this challenge, we decided to use both and compare the results.

2.3.3. Unfair discrimination and biases

We also considered the problem of unfair discrimination and biases of our model against people from certain race, gender (clear gender differences in communication styles on the social media), or age group and against other sensitive attributes. For example, in a research from the Northwestern University it was proven that when it comes to older people the model falsely reports more negative attitudes toward political issues. (CHI, 2018) In our case such biases would mean affected conclusions, that would need correction before they could be considered significant.

As stated in the investigation of our model, unfair discrimination of people is not an issue with our model but there is still a bias involved regarding voicing an opinion vs stating facts. This comes from using a specific subreddit as singular data source, which mainly reports news and facts rather than opinions (as it is on Twitter). This means that our model would automatically rather predict a post to be from Reddit if no opinion is included in the post – obviously this can be the case too for Twitter posts and for Reddit posts from other subreddits.

To improve our model, one could further search for other subreddits and make a reddit dataset that combines several subreddits e.g. subreddit r/politics, with subreddit r/political discussion. The goal is to have a similar structure as the analyzed Twitter data.

Another possible bias of the data could be the “group think” phenomenon, which occurs when people in your personal network support strongly one of the candidates, and you feel obliged to support them too, sharing highly emotional and radicalized opinions on social media, without really believing in them.

A bias we could in fact do something about is the botnet - fake accounts with predefined scripts or real people who fake an opinion. With our code we were able to identify those, but it was already too late to implement this finding and exclude them from our analysis. Still it was an useful learning for future projects.

3. Experience Gained

3.1. Challenges discussion

During our project we faced a lot of challenges and obstacles. In general, it was already harder to work together on a project completely remote, but with the help of MS Teams, shared docs, DeepNote and Jupyter we could meet virtually and write & code in real time together.

Talking about DeepNote while we started coding together, we realized it is pretty unpractical to use Sparks in DeepNote because it is not set up already, so we had to move all of our codes and data files to Jupyter after a couple of days.

Furthermore, we faced some issues with collecting the data related to our topic. Obviously, there are tons of tweets per day and only a small fraction of them is about Trump or Biden. Therefore, we had to find a way to only get the tweets relevant for our analysis. We chose the path of filtering all the tweets for at least one out of the two words “trump” and “biden” but of course there might be a lot of tweets about them not including their names. We could not find an easy manageable way include them in our analysis but in general the sentiments might look different if it would be possible to filter all the tweets about Trump, Biden, or both.

Originally, we planned to use more historical twitter tweets from the day of the capitol storming but as already mentioned earlier we only had one twitter developer account which was able to get access to historical tweets. Therefore, we only have a sample size of 3.100 tweets from the 06.01.2021 of historical twitter tweets. Generally, our results for the historical data cannot be classified as representative for the whole nation.

During our coding sessions we had some troubles with the spark data frame not loading. After trying out different approaches and reloading only parts of it we realized that the Vader lexicon was the issue. The results of the sentiment analysis with the Vader library did not load fast enough at all to process our 160k of tweets. Therefore, we mainly concentrated on the Textblob sentiments, although the Vader ones were more accurate.

Additionally, it was exhausting to find all the information about ethnical issues, licenses, and guidelines but it was absolute necessary to get familiar with that part and its importance for the data world as well.

At the beginning of our project, we had a lot of great and interesting ideas about what exactly we could do as a machine learning approach but while our coding continued, we realized most of our ideas are not workable. For example, we thought about training an algorithm to classify whether a tweet is more republican or more democratic, but we only have the sentiments towards Trump or Biden and logic wise a tweet being positive towards one of them does not mean a person is a republican or a democrat. Another problem was the fact that our data wasn’t labeled as democratic/republican and labelling our data would have taken enormous amount of time.

Therefore, we brainstormed what else we can use a classification algorithm on given our labelled data and came up with classifying whether a post came from Twitter or Reddit – as we already know that simply from downloading. This approach obviously is not as senseful or inherently interesting but still gave us a great chance to apply our learnings from class and further understand how classification of text data can be performed.

3.2. Experience gained by each team member

Group memeber 1

Starting with the Big Data topic and getting familiar with Sparks and Kafka and these huge amounts of data in class was already quite overwhelming and exciting. But then actually starting a project our own was even more crazy, super interesting, insightful but extremely challenging and hard as well.

I think the step from our last projects to work with Big Data was enormous but therefore I think we learned a lot and kind of understood the whole Data Science world a lot better, I did for sure. I am glad that we

have been in a group and could always discuss issues and tackle problems together because it is a lot of new stuff and to be honest a lot of errors occurred. But as soon as the code worked it was just an amazing feeling and a huge relief.

Additionally, it was exhausting to find all the information about ethnical issues, licenses, and guidelines but it was absolutely necessary to get familiar with that part and its importance in the data world as well.

I am extremely happy to have the opportunity to learn that much from the labs, from our teams calls, my team members, and from Stackoverflow & Github. My Python and Coding skills increased immensely, and I am happy that I could get familiar with Sparks and the extent of a Big Data project

Group memeber 2

As a person who started gaining programming skills only three months ago, I was completely overwhelmed by the endless opportunities one has when working with Big Data at first, I haven't even heard of most of the frameworks, introduced during the lectures before. These facts made the project very challenging for me, but also quite enriching. I had the opportunity to dive deeper into this section, learn a lot, also fail a lot to be honest, but with the help of my team members, the documentations and GitHub I was able to gain practical experience and do a lot of the things we discussed in class also by myself, which made everything more clear and not that scary, although the topic of Big Data and its processing still seems pretty endless to me. Nevertheless, the work on this sentiment analysis brought me more clarity and I discovered my interest in Natural Language Processing, which I would love to explore further with future projects. I also feel like my Python skills are strongly improved now and I can manage the work with Apache Spark too, which was pretty hard in the beginning, when it came naturally for me to process the data only with Pandas.

I am happy we could realize all of our initial plans of the project, because in the beginning it all seemed unmanageable. I am proud of the results and grateful for my team, their support and active participation, because the good teamwork is one of the main factors that turned the project into such a great learning experience.

Group memeber 3

I can split my experience gained into two main segments: practical coding experience and theoretical application experience. The coding experience just came from trying multiple times to preprocess the data, run through code, solve errors, and finally visualize the results. The experience gained here is crucial for me as it now puts me into the position of "whatever problem comes my way, with a fresh eye and an explorer attitude I can solve it". This was especially important while working on the machine learning part in this project as even after hours of not being able to solve an error or get the result I want, I learned to take a break and just try it again.

The theoretical or conceptual understanding I learned is even more significant. My main goal was to really understand the possible applications of analyzing big data. For me the key thing I want to take out of university is to solve real world problems – and for that I need to understand exactly what the underlying technology can do and what it cannot do. In this project, I definitely understood part of it and what

questions to ask e.g. do you have labeled data? when it comes to solving a problem. Still, after applying certain machine learning algorithms I feel like there is a lot more to learn and to explore in the section.

Overall, I greatly benefitted from this project and I will take this knowledge into my work as research assistant at the International Institute for Applied System Analysis and at WU. The key parts I take with me is the time necessary to spend on data preprocessing, a "can-do" attitude, taking a break when looking at the same problem for 5 hours and finally to always have the goal of code in mind – something like: write Pseudo Code, code working solution and then make the solution elegant!

myself

Working on this project was extremely interesting and surprisingly challenging for me. I already have some experience with natural language processing, but I have never worked with the big data frameworks mentioned in class. Therefore, I needed some time to get used to the Apache Spark environment and needed quite some time to find solutions to all the errors we have gotten, but now I am proud of our results and eager to dive deeper into this fraction of the data science topic.

One key learning is my new developed ability to quickly search through the documentation of PySpark and improve my skills in working with it for such a short time period, as a lot of information was not easy to find on the internet, even on GitHub and Stackoverflow.

Our group dynamic was great, and we could complement each other's strengths and weaknesses, conduct super productive MS Teams calls and everybody was eager to invest their time and motivation.

Generally, I am glad for getting new hands-on experience with Big Data technologies and the various theoretical insights from the course.

3.3. Recommendations for future work

In general, there are probably a lot of different aspects that we could have handled better or a little different but in the following we will just point out a couple of things that could be important to think about before starting any future projects.

First, we can absolutely recommend planning the project more detailed in the early stages. If we would have started planning earlier as well as shaping our idea earlier, we could have been able to compare tweets from exactly 7 days before a specific event, the day of the event and 7 days after the event. It would be nice to have a stable framework about the exact time when getting the data as well as the exact amount of data to have more accurate results in the end. For example, three points with each 7 days apart and each 50.000 tweets. Then the reliability and accuracy of data probably will be a little higher.

Another factor we would keep in mind in our next projects is to start thinking about what is possible with Machine Learning approaches. As mentioned in our challenges, we had the issue to come up with ideas that make sense and that are possible with our data. Especially regarding data availability, we will spend more time on investigating how our available text data (on reddit/twitter or any other source) can be used to perform machine learning. Next time we will probably focus on a smaller part of ideas but form the ideas in a more detailed and structured way. A project should not be only interesting, but it should have a strong connection between each part of it.

Additionally, we thought about comparing more than just two libraries in the sentiment analysis would make a lot of sense. With the comparison as we did, we can see the difference between two libraries and

see if they produce similar or different results. But only if we compare at least 4-5 libraries we could see if there is a trend in the results and maybe two or three of them produce super similar results and might be the best option. So, in general, we would think about using more libraries for the sentiment analysis in case the resources allow it.

References

CHI, A. (2018). *Addressing Age-Related Bias in Sentiment Analysis.*

https://www.reddit.com/r/pushshift/. (n.d.).

https://www.reddit.com/r/pushshift/comments/bcxguf/new_to_pushshift_read_this_faq/. (n.d.).

LEE DE-WIT, S. V. (2019). *Are Social Media Driving Political Polarization?* https://greatergood.berkeley.edu/article/item/is_social_media_driving_political_polarization.

Mims, C. (2020). *Why Social Media Is So Good at Polarizing Us.* The Wall Street Journal.

Paulus F.M., M.-P. L.-Q. (2019). *The politics of embarrassment: considerations on how norm-transgressions of political representatives shape nation-wide communication of emotions on social media.* https://scholar.google.com/scholar_lookup?journal=Front+Commun&title=The+politics+of+embarrassment:+considerations+on+how+norm-transgressions+of+political+representatives+shape+nation-wide+communication+of+emotions+on+social+media&author=F.M.+Paulus&autho.

Sayce, D. (2020). *The Number of tweets per day in 2020.* dsayce.com.

Union, E. P. (2018). General Data Protection Regulation 2016/679 .

Ursula Garzcarek, D. (2019). *Approaching Ethical Guidelines for Data - https://arxiv.org/pdf/1901.04824.pdf.*

Weinstein, J. (2019). *How to mine data from Reddit.* medium.com.

White, B. (2020). *Sentiment Analysis: VADER or TextBlob?* towardsdatascience.com.

ZAID JILANI, J. A. (2019). *What is the true cost of polarization in America?* https://greatergood.berkeley.edu/article/item/what_is_the_true_cost_of_polarization_in_america.