Web Data Mining

In-depth Analysis of Twitter Data

Part 2

Napoleon-Christos Oikonomou
AEM: 16
onapoleon@csd.auth.gr

Maria Kouvela
AEM: 9
mvkouvela@csd.auth.gr

Introduction:

In this part of the assignment we were asked to focus on a specific topic and analyze Twitter content about it. As such, we focused on a set of hashtags that characterize it, gathered a dataset of tweets that contain those hashtags and performed various analyses on this dataset. We performed three kinds of analyses: Emerging Topic Detection, Sentiment Analysis and Geo-location Analysis.

The analysis and its results are described below.

Part 1 - Data collection from Twitter and Data storage – Build a Social Listener:

Using the Twitter Streaming API we were able to retrieve incoming tweets according to our predefined set of hashtags: *'#Tesla'*, *'#TeslaMotors'*, *'#ElonMusk'*, *'#Elon'*, *'#Model3'*, *'#ModelX'*, *'#ModelY'*, *'#TeslaRoadster'*, for our topic of interest which is Tesla Motors, an electric car company founded and directed by Elon Musk.

This was done using a script in Python, that utilizes the tweepy library. The script listens for new tweets that contain one or more of the above hashtags and saves each one of them to a MongoDB database. Noteworthy is the fact that a cluster of MongoDB shards and various indexes were created, in order to make data writing and subsequent queries require less time to execute, respectively.

Overall, we ran the script from April 1st to May 30th and managed to retrieve >154K tweets in total. Sadly, due to the real-time nature of an event listener, we faced a problem of losing the connection to API, that resulted in losing ~6 days' worth of tweets. We also decided to keep the retweets, since they provide a good indication about what Twitter users are talking about on their tweets.

Another problem that we had to deal with was that we realised that the API did not returned the tweet's full text, but only a part of it, which we realised after the dataset was completed. After doing some research we discovered that we should have used an extra parameter when using the API endpoint "tweet_mode=extended". To deal with that, we wrote a separate script that made a call to the statuses/show/:id endpoint for each of the tweets' unique twitter id and managed to gather all necessary information and save it to MongoDB under the "full_text" key. Figure 1 shows each day's tweet count.

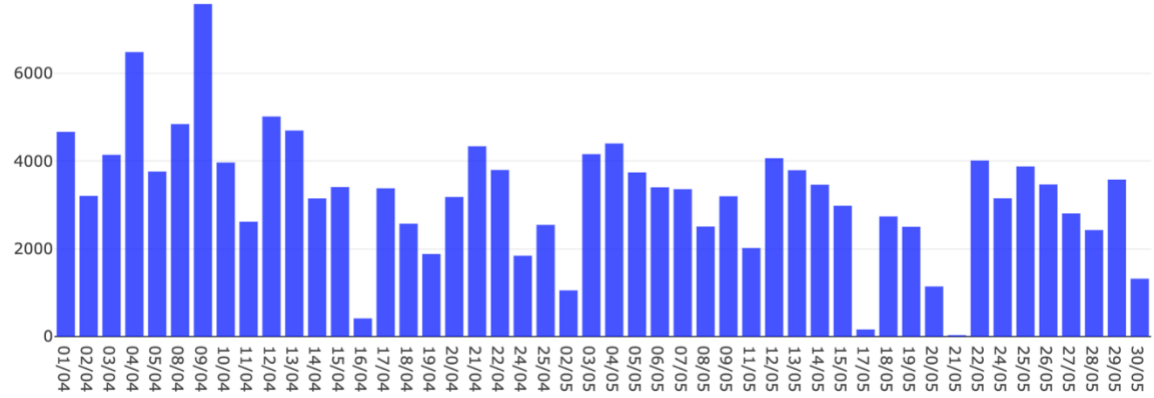

Figure 1: Tweet Activity Over Time

Relevant files: `twit2mongo.py`, `twit2full_text.py`, `plot_count.py`

Part 2 - Tweets preprocessing and modelling:

The Twitter Streaming API provided us will a very detailed json object that accompanied each tweet's text. This object included very fine-grained information such as hashtags, URLs, geo-location info, RT status, user information and more, which is explained in detail in: https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json. As such, modelling of each tweet wasn't necessary.

What was necessary, was cleaning of the text. To do that, we used a module developed by AUTH's OSWinds team, available at: https://github.com/vasisouv/tweets-preprocessor. From this module, we used methods to remove:

- URLs
- Mentions
- Hashtags
- Twitter reserved words
- punctuation
- Single-letter words
- Stopwords (where we enhanced the list with some stopwords of our own)
- Numbers

We also converted all strings to lowercase and striped multiple blank spaces.

The resulted text was saved to MongoDB under the "clean_tweet" key.

Relevant files: ` twitter_preprocessor.py`, `helpers/utils.py`, `helpers/regex.py`, `clean.py`

Part 3 - Emerging topics and information evolution in time:

In order to detect emerging events, we followed two approaches. The first approach was finding the 10 most common hashtags used throughout the dataset and then plot their normalized daily appearance count. This led to the following plot:

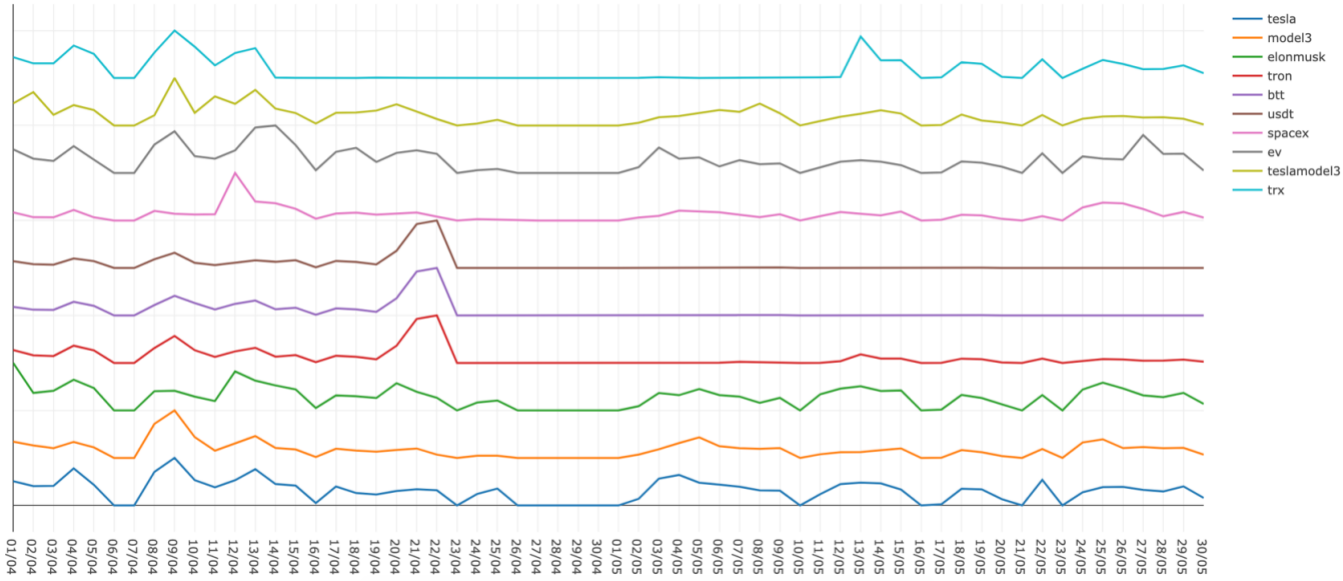

Figure 2: Normalized frequency of 10 most common hashtags

A second representation depicting the 200 most common phases (words or bigrams) can be seen below:

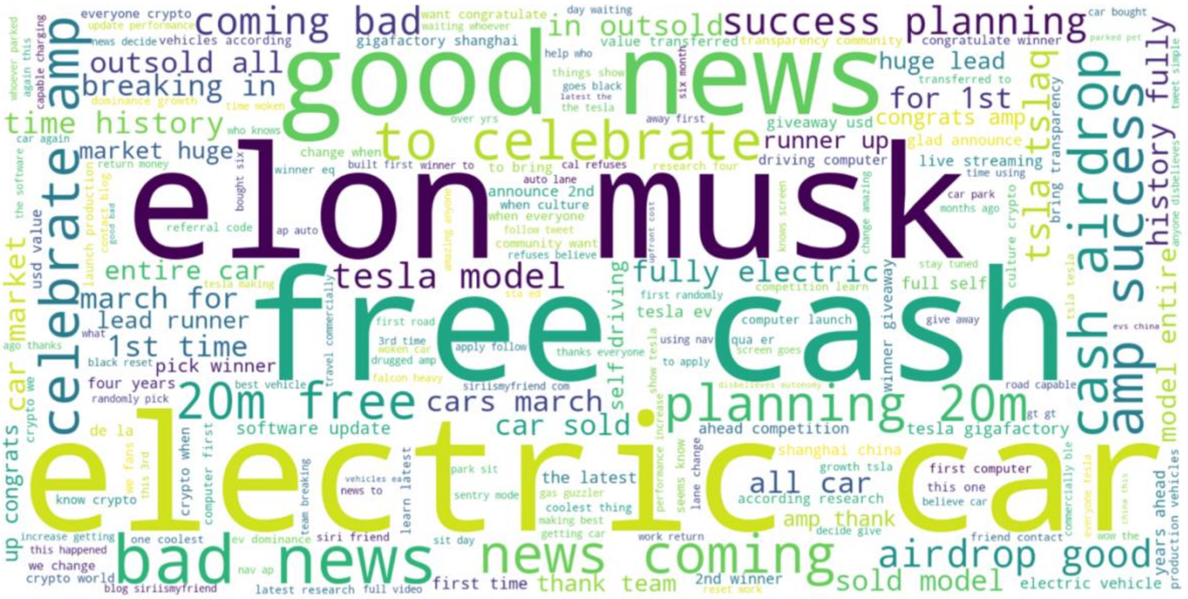

Figure 3: Most common phrases in tweets

As it can be easily seen, finding emerging events from a plot like this is a rather difficult task. Maybe one can see a topic about the TRX, TRON, USDT cryptocurrencies around the 21st of April (which, after checking online news outlets, seems to be when Elon Musk followed Tron's CEO on Twitter), and another one about SpaceX (Elon Musk's rocket & spacecraft manufacturing company), around the 12th of April, which is when they managed, for the first time, to land all three rocket boosters following a rocket launch.

As such, we tried a second approach, in an effort to get a better understanding of emerging topics. After doing some research on topic detection in the first part of this assignment, we based this approach on the notion of detecting "bursty" keywords, as explained in "Twitter Monitor", by Mathioudakis et al. (2010)[1]. The steps that were taken are as follows:

Firstly, we split the dataset into 12-hour time-windows. For each time window we gathered the clean text of all its tweets and created a vocabulary of each word (or hashtag) that was used.

Secondly, we counted the occurrence of each of these keywords and by making the assumption that the occurrence follows a Gaussian distribution, we kept only those who had $count > \mu + 2\sigma$, because these are the keywords that exhibit "abnormally" high occurrence.

Thirdly, we first created groups of keywords, based on their co-occurrence in tweets, in order to create topics and then merged groups that had more than 50% similarity.

Lastly, through manual inspection and online research, we grouped again those topics, in order to create events that make sense. This resulted in the topics presented below:

- 01/04/19 - 02/04/19: **['model3 car tsla', 'tesla elonmusk harambe', 'amp']** *Elon Musk shares link to his rap song as a tribute to the death of gorilla Harambe.*
- 03/04/19 - 05/04/19: **['car tsla', 'goes money work reset knows who help black screen back return']** *Tweet of singer @SherylCrow to @Tesla about a problem with her car, which became trending because a lot of people were experiencing it.*
- 07/04/19 - 08/04/19: **['10 free cost', 'thank entire sold time lead march cars for team 37 model3 electric market model outsold huge car runner amp all congrats 1st in up**

[1] Mathioudakis, Michael & Koudas, Nick. (2010). TwitterMonitor: Trend Detection over the Twitter Stream. Twitter monitor: Trend detection over the twitter stream. 1155-1158. 10.1145/1807167.1807306.

breaking history fully'] *In March Tesla was 1st in car sales, which marked the first time in history that an Electric Vehicle company achieved that.*

- 09/04/19: [**'car china', 'this 2019', 'best increase months everyone software thanks getting wow new time the update performance']** *Tweets about Tesla increasing power and speed performance of Model3 car due to software update.*
- 10/04/19: **['best this everyone updates thanks vehicles new wow']** *Tweet thanking Elon Musk for the software update on model3 car.*
- 11/04/19**: ['park spacex', 'believe waiting whoever pet parked day refuses sit 100 cal']** *"Funny" tweet by @idiocyafoot, detailing who a car can park all by itself, that got viral.*
- 12/04/19 - 13/04/19**: ['money happiness crying buy better lot whole', 'miles 27 over thanks love new thxelon year 000']** *Tweets from Tesla owners detailing who good a decision was buying a Tesla car, financially speaking.*
- 16/04/19 - 17/04/19: **['repo impact']** *Probably referring to Susan Repo leaving Tesla.*
- 19/04/19 - 20/04/19: **['amp tron it usdt cash btt 20m coming news success celebrate airdrop planning to free bad good i']** *Tweet by @justinsuntron, TRON cryptocurrency's CEO on giveaway, that got RTd by Elon Musk.*
- 21/04/19 - 22/04/19: **['this bad shanghai happened evs china anything chi positive i today post to negative good']** *Reports about a Tesla car that spontaneously caught fire in a parking lot in China.*
- 25/04/19: **['q1', 'tsla insurance']** *Tweets about Tesla introducing its own insurance plan after Q1 of 2019.*
- 03/05/19 - 04/05/19*:* **['first full computer four vehicles ahead learn competition latest research years self-according launch production driving']** *Tweets commenting on the results of independent analyses that Tesla is 4 years ahead competition in self-driving technologies.*
- 05/05/19: **['police cruiser accelerate policing model3']** *Tesla Model 3 police car makes an appearance at law enforcement tech conference*
- 06/05/19 - 07/05/19*:* **['monumental years made ed achievements video 17 ago here spacex founded celebrating sta']** *Elon Musk sharing video link of cargo craft from @Space_Station*
- 11/05/19 - 12/05/19*:* **['please finger request work saturation undos multiple picker add painting color adjustment', 'just world software check come']** *People talking about a new feature appearing in Tesla's car, allowing users to create hand-drawn paintings.*

Representative online content of the above topics can be seen at the website accompanying this report.

Relevant files: `find_topics.py`, ` plot_most_common.py`, ` wordcloud_most_common_words.py`

Part 4 - Sentiment and emotion information extraction:

In this part of the assignment, we are interested to identify and categorize opinions expressed in tweets, and better understand the users' emotions over Tesla. We followed two approaches; one to classify tweets in the scale of positive, neutral and negative, and a second one to match tweets to specific emotions, such as anger, joy, disgust, fear, sadness, surprise.

For the first part we used a commonly known python library for processing textual data, called textblob (https://textblob.readthedocs.io/en/dev/index.html). This library uses pre-trained models from Stanford's NTLK platform[2], in order to predict a given text's polarity (a number in [-1, 1], where 1: positive and -1: negative). After passing each tweet's through textblob and calculating each polarity and sentiment (saved under the 'overall_polarity' and 'overall_sentiment' MongoDB keys) we created the following plot.

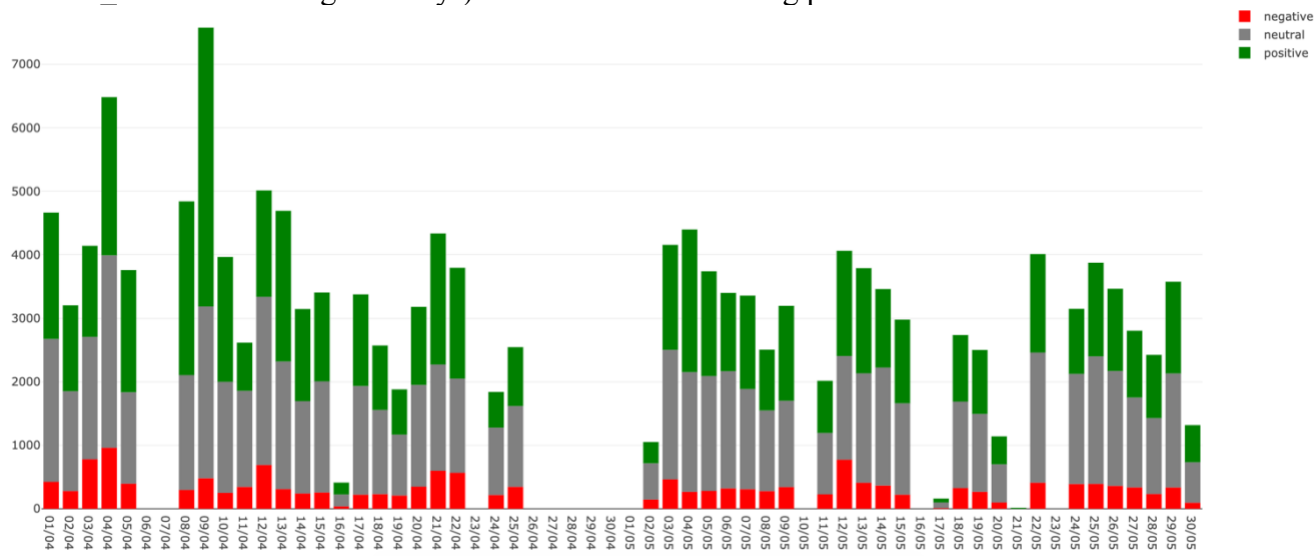

Figure 4: Overall daily sentiment of tweets

As it can be seen, Tesla is a company that enjoys public acceptance and the number of people that tweet positively about it is consistently at least 3 times higher than the ones that tweet negatively. Also, the relatively large number of neutral tweets (those whose polarity equals to 0), suggest that the people talking about tesla aren't only fans or haters. A plot of the numeric value of the polarity is shown below:

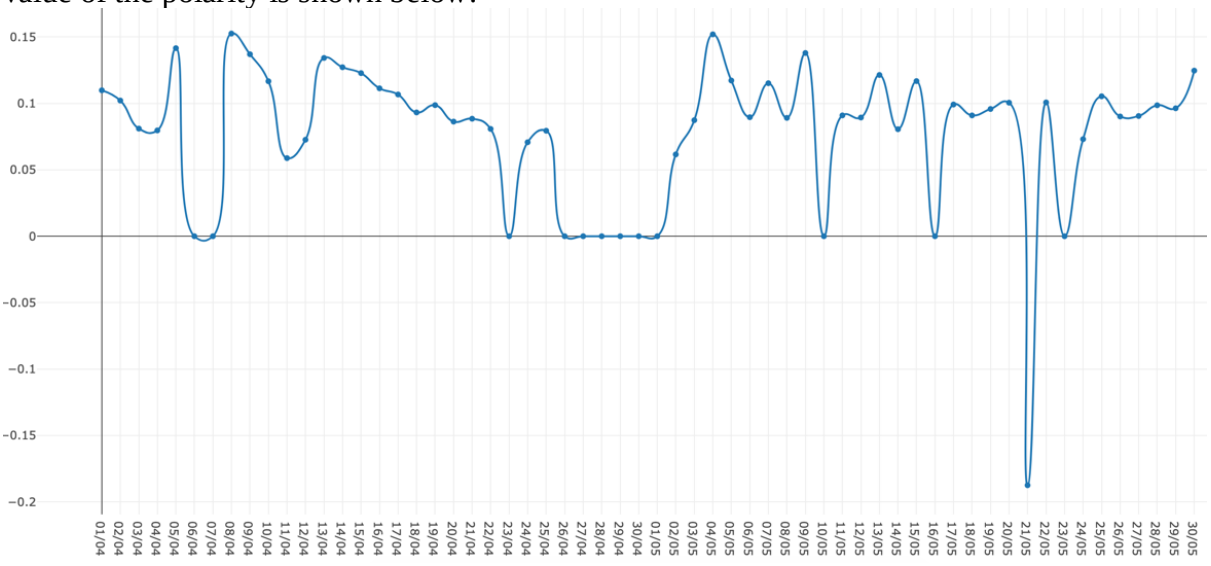

Figure 5: Daily polarity of tweets

As it can be seen, the overall sentiment about Tesla is (highly) positive almost every day. The only time in our two-month period that it appeared negative was on the 22nd of May, following Tesla's demands for special treatment from the US government in its Chinese exports. Noteworthy is the fact that this observation differs from Figure 4. This happened because even

[2] https://www.nltk.org/

though more tweets were positive on that day, those that weren't, were a lot more offensive towards the company and its CEO.

For the second part of our sentiment analysis, that is matching tweets to specific emotions, we created a multilabel classifier using the training set of tweets that was given by the instructors of the course. We experimented with different kinds of classifiers and features but settled to a bag-of-words representation[3] and a Random Forest classifier[4], with Binary Relevance to transform the input dataset[5]. As it is obvious, to create the bag-of-words representation, we first pre-processed the input dataset using the same technique as with the gathered tweets, as described in Part 2 and then, concatenated it with every collected tweet to create the full vocabulary. To evaluate our classifier's performance, we used the Hamming Loss metric[6] and managed to produce a fairly accurate model, with Hamming Loss ~18%. We then used our model to predict the emotions of each tweet and saved the results in the 'six_sentiments' MongoDB key. Our results are shown below:

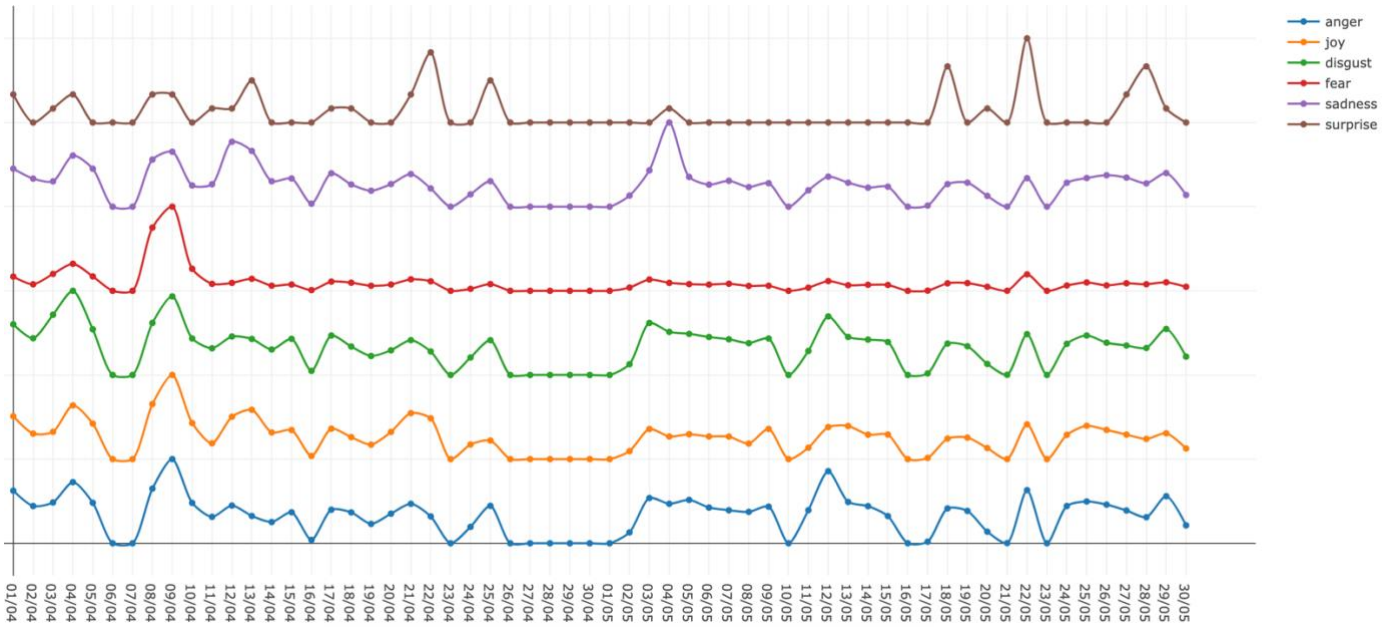

Figure 6: Sentiment value per day

From the above figure, we can extract the information that tweets about Tesla aren't characterized by fear. This could mean that Tesla, its products and its future are trusted upon by the general public. Weirdly enough, though, there are a lot of tweets that characterized by the emotion of disgust. To grasp a better understanding of the specific emotions throughout our dataset, we processed to create word-clouds about the most common phrases (words and bigrams) for each specific emotion. These are presented below:

[3] https://en.wikipedia.org/wiki/Bag-of-words_model
[4] https://en.wikipedia.org/wiki/Random_forest
[5] https://en.wikipedia.org/wiki/Multi-label_classification
[6] Grigorios Tsoumakas, Ioannis Katakis. Multi-Label Classification: An Overview. International Journal of Data Warehousing & Mining, 3(3), 1-13, July-September 2007.

- Anger:

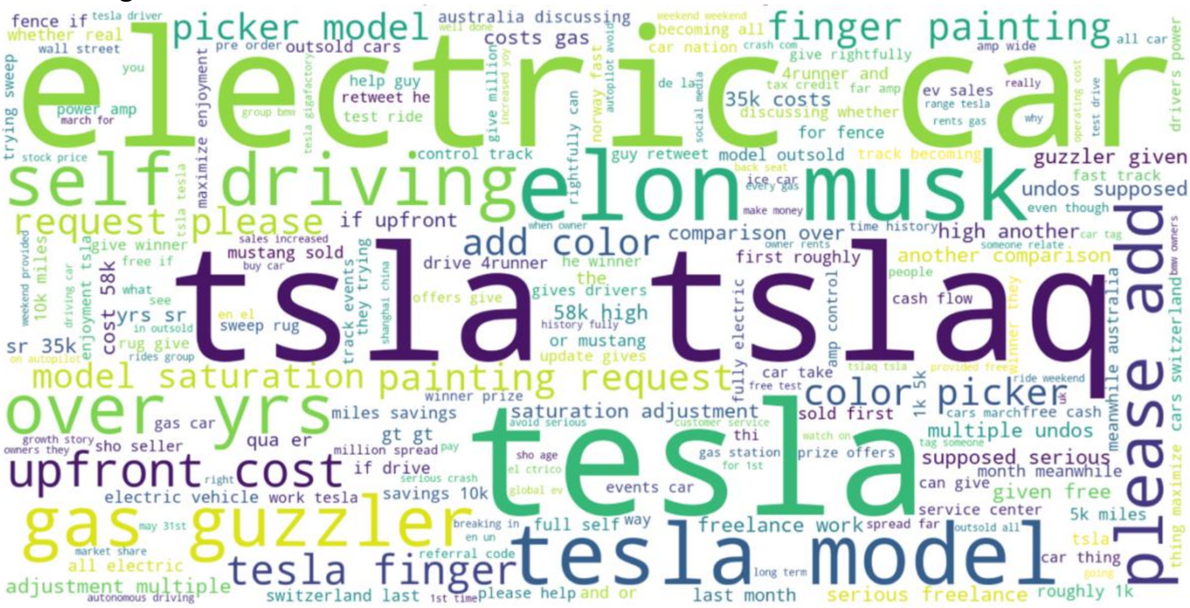

Figure 7: Most common phrases in 'anger'-tweets

In an effort to sum the above, after manual inspection of tweets, one could say that: "Tesla owners are irritated that people say Tesla cars are expensive, because they don't take into account that upfront cost is diminished throughout the years of gas savings".

- Joy:

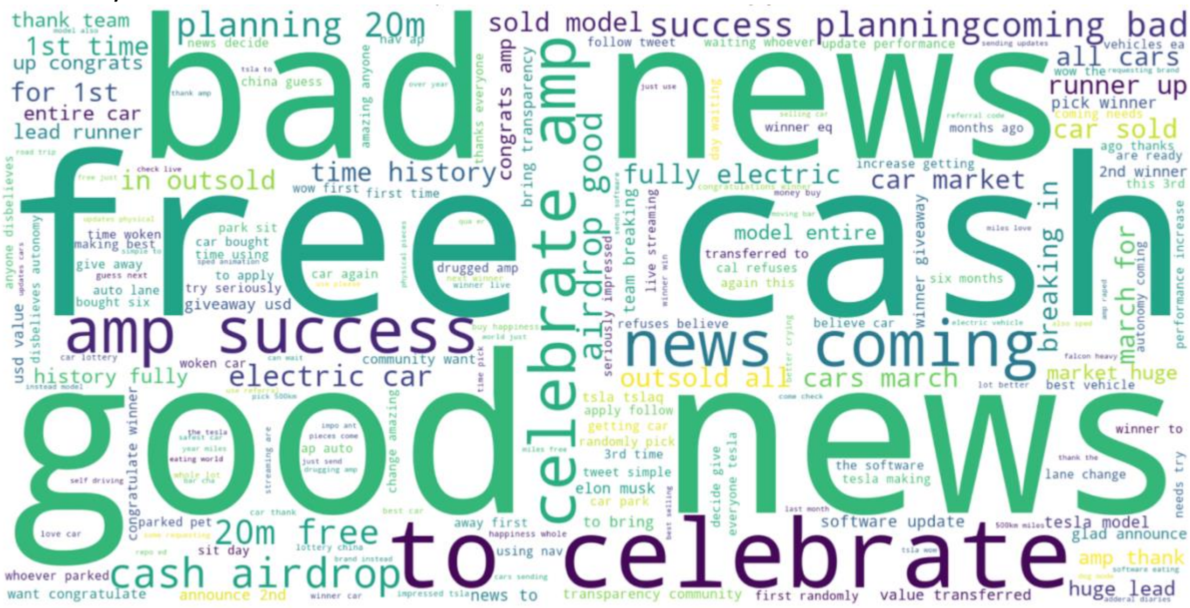

Figure 8: Most common phrases in 'joy'-tweets

In an effort to sum the above, after manual inspection of tweets, these tweets seem to be about a 20-million-dollar giveaway by TRON, whose connection with Tesla was described in Part 3. Weirdly enough, the phrase "bad news" is one of the most common. Maybe people are happy when there are new about Tesla, even bad news.

- Fear:

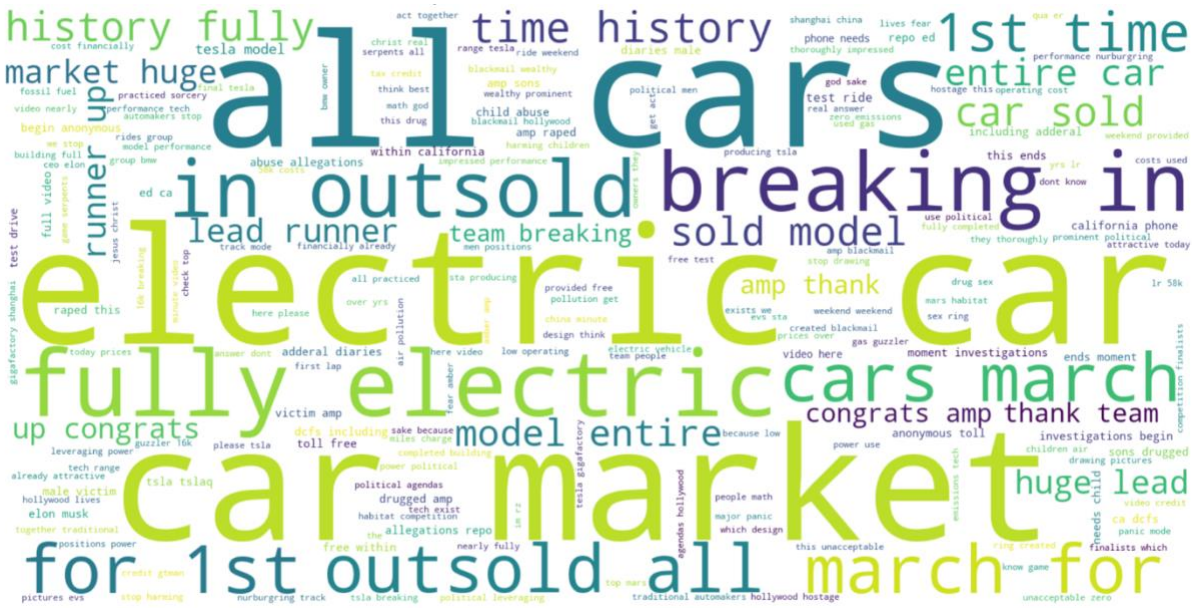

Figure 9: Most common phrases in 'fear'-tweets

In an effort to sum the above, after manual inspection of tweets, one could say that: "People express concerns about the future of traditional car manufactures and about the future car safety because of the shift to autonomously driving vehicles".

- Sadness:

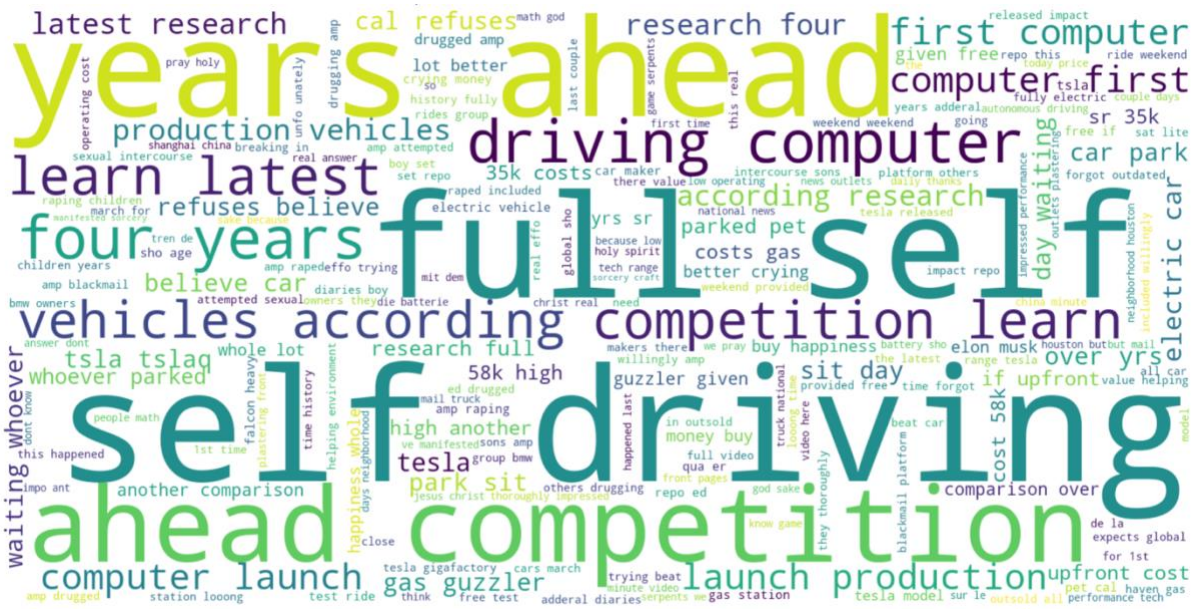

Figure 10: Most common phrases in 'sadness'-tweets

In an effort to sum the above, after manual inspection of tweets, one could say that: "People are "sad" that there aren't any other manufacturers invested in EV technologies as much as Tesla is".

- Surprise:

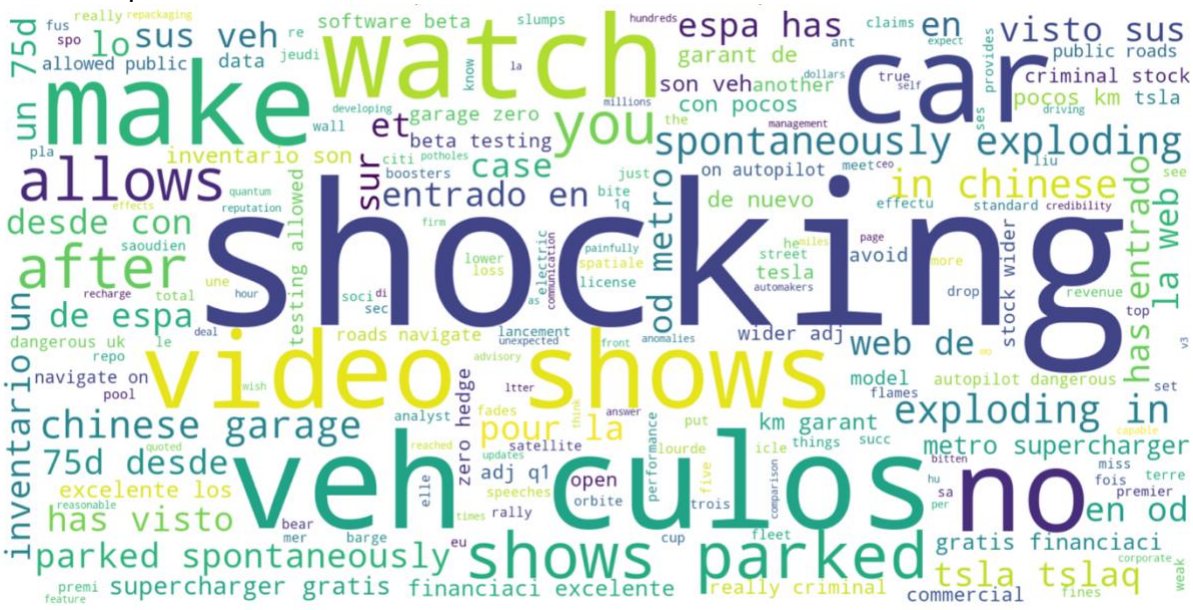

Figure 11: Most common phrases in 'surprise'-tweets

In an effort to sum the above, after manual inspection of tweets, these tweets seem to be about the shocking event of a car that spontaneously caught fire, as explained in Part 3.

- Disgust:

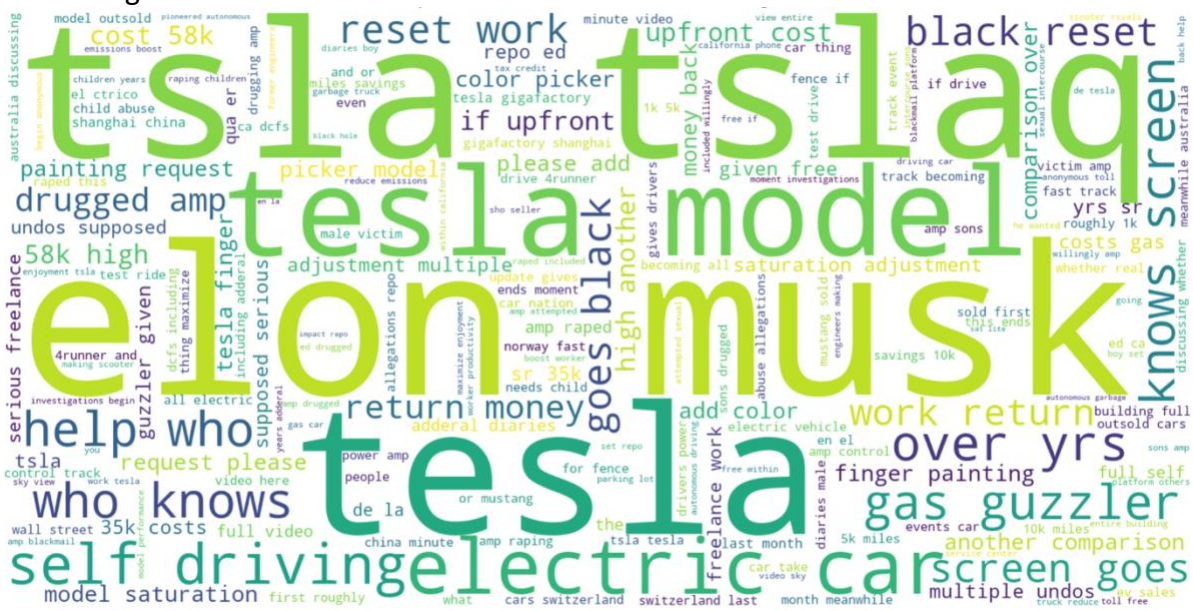

Figure 12: Most common phrases in 'disgust'-tweets

As discussed in the observations from Figure 6, above, most tweets seem to be related with the emotion of disgust. But, by looking at the word-cloud of their most common phrases and through manual inspection of tweets, we can observe that those are not 'disgust'-tweets. Rather, they appear to be tweets that don't contain any particular emotion and the classifier "learned" this absence, as the emotion of disgust.

Relevant files: `overall_polarity_plot.py`, `overall_sentiment.py`, `plot_sentiments.py`, `six_sentiments.py`, `wordcloud_most_common_words.py`

Part 5 - Geo – information extraction:

In this part of the project, our task is to extract information on users' geographic locations, based on their tweets.

Unfortunately, the number of tweets that included a specific location was very small; only 598 tweets (~0.38%). There were another 2642 (~1.7%) that included information about a known place, which we were able to pinpoint to a specific set of coordinates. Moreover, we weren't able to extract any other geographical information from tweets, except from some cases where a US state or a European country was mentioned. Overall, we manage to collect 3240 tweets, which we were able to use for the geo-location analysis.

In Figure 13, a map of these tweets is presented, where each point is coloured based on the overall sentiment of the tweet (green for positive, red for negative and grey for neutral). A more usable, interactive version of this map exists in the website that accompanies this report.

As it is evident, the topic of interest that we chose, is not tweeted about from all over the world. Tweets come mostly from the US, where Tesla is based, and from some European countries for the most part, where people own Tesla cars. Regarding the USA, most tweets come from the East-coast, but this is to be expected, as most of US citizens live there (~40%). But noteworthy is the fact that while Americans tweet mostly positive towards Tesla, European was a much more neutral stance.

Lastly, we gathered each user's reported location and created a word-cloud of the 200 most common ones. Each reported location was pre-processed, as described in Part 2. This is presented below. As we can see, most people are located in the United States.

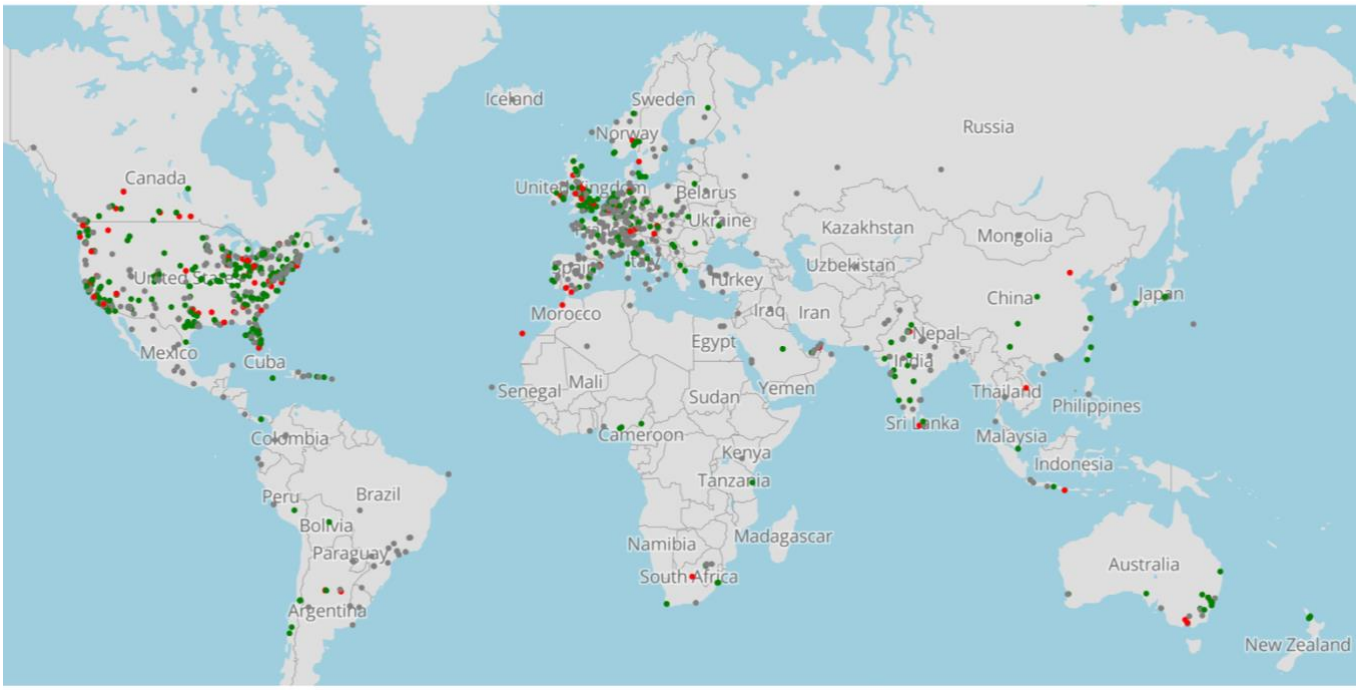

Figure 13: Map of tweets

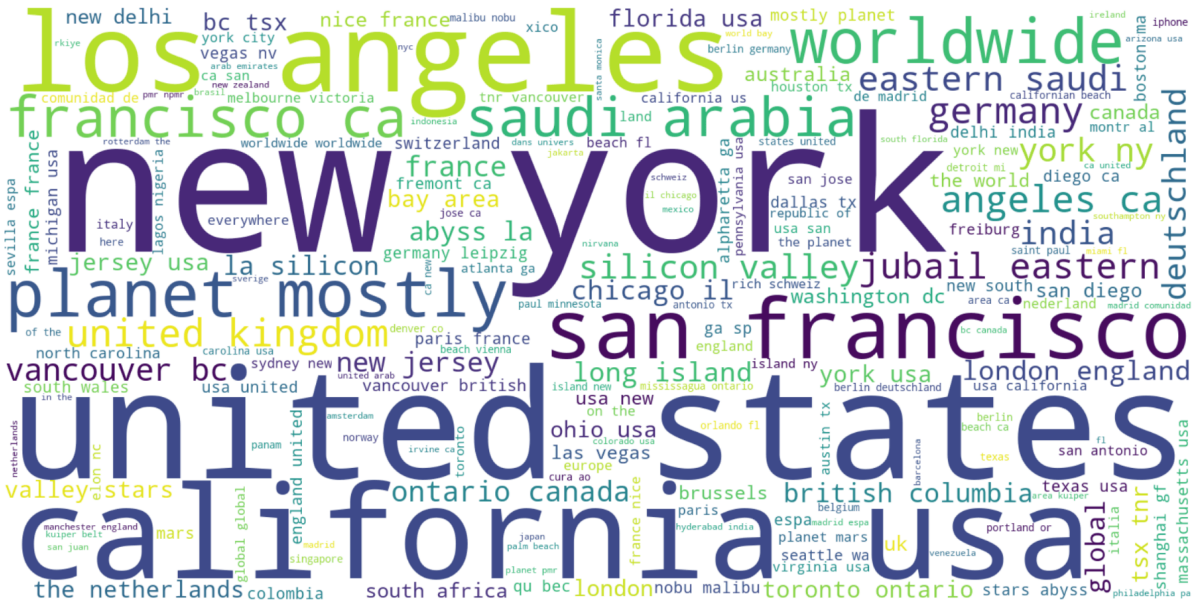

Figure 14: Most common user locations

Relevant files: `plot_location.py`, `wordcloud_most_common_locations.py`

Part 6 - Notes:

For our analyses we used the Python programming language. Notably, we used libraries like:

- tweepy: To connect with the Twitter API
- pymongo: To connect with the MongoDB Database
- wordcloud: To create word-clouds
- textblob: To extract overall sentiment polarity
- plotly: To plot our results.
- sklearn/skmultilearn: To train the multi-label classifier that we used to predict emotions.

In the README.md of the 'code' folder that is attached with this report, there is detailed information on how to install and execute all the scripts that were written, alongside the scripts themselves.

For the website that we created, we used the Node.js runtime engine, using the JavaScript programming language. Notably, we used the frameworks Gatsby.js and React.js to create the necessary HTML, CSS and JS files. The Gatsby.js starter template that we used is mentioned in the footer of our website. There is an online version of the website hosted on: https://twitter-analysis.netlify.com/. Also, in the README.md of the 'code' folder that is attached with this report, there is detailed information on how to install and run the site locally, alongside its source code.

All the aforementioned files, along with the dump of the MongoDB database that we used, which we couldn't attach due to its large size (>1GB), are available to the assignments repository, at: https://github.com/iamnapo/tesla-web-mining.