IML’s hackathon

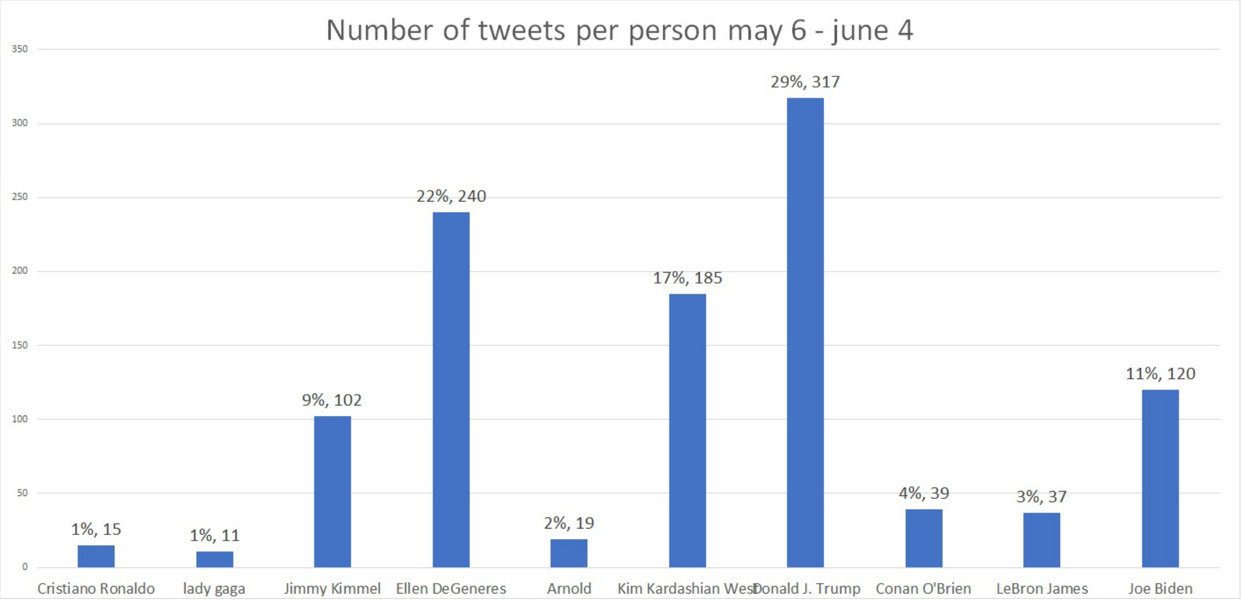
Chen Shani, Daniel Rotem, Ofir Shifman and Nitzan Luxembourg

June 7, 2019

1 Pre-processing the data

We first started by looking at the data and decided what needs to be done in the pre-processing stage.
We separated URLs, emojis from the whole tweets and retweets (kept them aside for later use). We also removed inappropriate capital letters and spacial characters as a preparation for the language models. Finally, we corrected typos, spelling mistakes and slang (kept track as well).

We noticed that all persons have different amount of tweets, which made us look at the distribution of tweets for each of them as a function of time. We measured the amount tweets in the last month (see figure below). Our assumption is that if this distribution can represent the future distribution - it can be useful to learn on a similar distribution of data.



2 Features

Since this task includes language we stated by extracting both ”trivial” and linguistics features. Trivial features are such as: number of words in a tweet, if contains URL, if a retweet, bag of words (BOG) of emojis, if the tweet starts in a capital letter, number of tags, number of punctuation and more. The linguistics features are such as: language model (LM) that represents a probability distribution of words for each person (uni-gram) to estimate the probability of label (person) given a word (for all words in a tweet), polarity and subjectivity scores for each tweet, the language itself (English, Portuguese, etc..), and more. We also had many features that were not useful enough for the final prediction such as number of nouns, number of sentences present and number of syllables in the tweet and estimated school grade level required to understand the tweet. One feature we did not manege to use properly on time is sentence embedding using BERT model (transformers based NN architecture for an holistic LM).

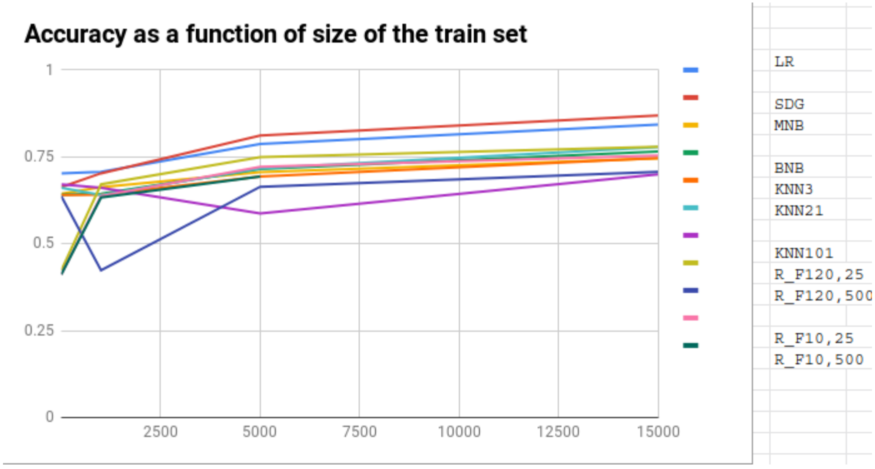
As we approached the dead-line we realized we won’t be able to fulfill our goals due to many bugs. Therefore, we did not use many of the features which we pre-made (such as uni-grams, polarity and subjectivity). We did use a statistical method of evaluating the significance of a word in given corpus where rare words are considered to be more meaningful (TF-IDF), along with the distribution of persons as a function of time mentioned before.

3 Models

We used many different models for this multi-class prediction task and measured their accuracy on the test set for fine-tuning. Finally, we used the validation set to measure the more realistic accuracy. We then used only the best models for the complete model. The models we used are: Gaussian Naive Bayes, Naive Bayes classifier for multinomial models, Naive Bayes classifier for multivariate Bernoulli models, Logistic Regression classifier, Linear classifiers (SVM, logistic regression, a.o.) with stochastic gradient descent (SGD) training, SVM (multi-class extension is handled according to a one-vs-one scheme), K-Nearest Neighbors, Random Forest (meta estimator to fit a number of Decision Trees classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting).

Important to mention, we used both under and over sampling methods in order to create train and test sets that will follow the distribution we found as a function of time. This is due to the fact that the final test set will be collected on a given period of time, and the classes will not be balanced (and hopefully - follow this distribution).

Below we can see the accuracy on the test set as a function of the size of the train set for the different models.


Accuracy as a function of size of the train set
LR
SDG
MNB
BNB
KNN3
KNN21
KNN101
R_F120,25
R_F120,500
R_F10,25
R_F10,500