Trump Twitting Identification

Nitzan Guetta

Abstract

The main purpose of this assignment is to distinguish between Trump and his staffers tweeting. In this assignment, several supervised machine and deep learning models were explored, in addition, Trumps' Twitter account data was analyzed and processed by different methods. All of these were performed in order to find the best classification model.

1 Data Exploration

In order to understand the data better, I performed several analyses such as clustering and sentiment. First, I wanted to understand the given tweets semantic. Therefore, I performed clustering with `KMeans`[1], on processed tweets of the training data. The processing of these tweets was averaging of the embedding vectors for each word in a tweet. The embedding vectors for this mission were taken from `google news`[2] embedding vectors, and were extracted by `Gensim`[3] package (more details in section 2). Then I used `PCA`[4] algorithm for dimensionality reduction. The results presented in Fig 1. While three of the clusters (red, blue and purple) captured Trump, the other cluster (green) has captured Trump staffers (mostly). So, we can clearly see from the graph that most of the staffers semantic, is different from Trumps' semantic, even though it is expected that Trump and his staffers will refer to the same occasions. Therefore, such semantic difference is an interesting result.

Second, I explored the sentiment of Trump and his staffers over time, expecting for insights. In Fig 2 we can see the staffers tweets sentiment.

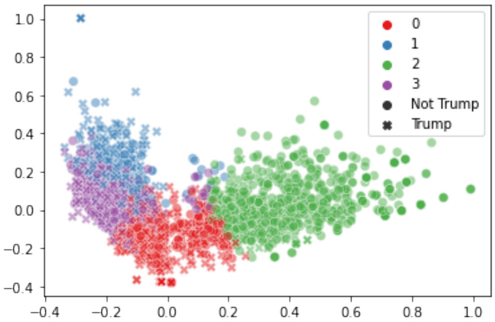

Figure 1: Kmeans clustering and PCA dimensionality reduction on processed tweets.

During the presidential campaign (June 16, 2015 - November 8,2016) the positive tweets were dramatically high in comparison to non-campaign time, which it is reasonable from marketing point of view. The natural and negative tweets were basically the same, but few months before the elections day we can see reduction in both of the types too.

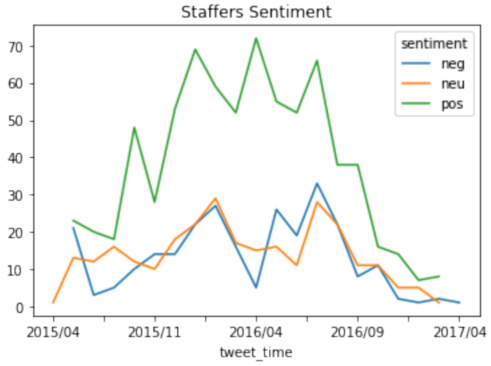

Figure 2: Staffers sentiment analysis over time.

About Trump tweets sentiment, which presented in Fig 3, we can see that since the presidential campaign started: (a) The positive tweets of trump reduced. (b) The negative tweets increased. (c) During his campaign time, and especially few months before the elections day, we can see reduction in all tweeting sentiment types.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`

[2] `https://code.google.com/archive/p/word2vec/`

[3] `https://radimrehurek.com/gensim/`

[4] `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

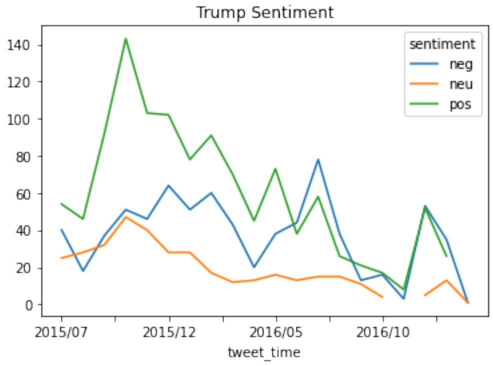

Figure 3: Trump sentiment analysis over time.

This is interesting, because when exploring the data distribution (tweets of Trump and staffers over time), as presented in Fig 4, we can see that during the presidential campaign Trump activity was reduced. This can verify the claim that Trump was kept away from his Twitter account during the campaign. Another thing that can explain this claim, is that Trump negative sentiment was increased during that time.

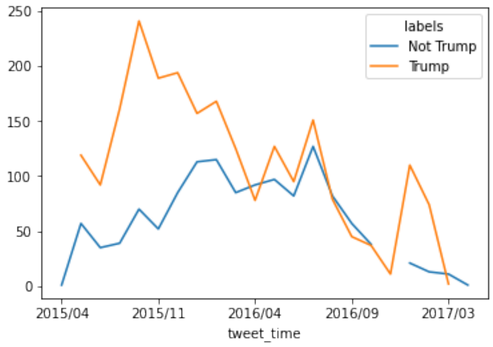

Figure 4: Trump and staffers tweets over time.

2 Data Processing and Classification Models

The data was sorted by time, and the training and testing procedures keep this restriction, since I want to avoid future-looking when I train the model. Therefore, Nan time values were dropped. Tweets related to Trump were labeled like this only if the device is Android, the user is 'realDonaldTrump' and the tweet is not a retweet.

The text was processed differently for each model type (machine or deep learning), and each feature for machine learning models type was also processed in unique way. Detailed explanation will be presented in the next subsections.

2.1 Machine Learning

2.1.1 Models

- **SVM**: With rbf and linear kernel types. Dimensionality reduction was tested but it was not improve the results, so eventually I did not use it.

- **Random Forest**: With the parameters $n\_jobs = 2, random\_state = 0$

- **Logistic Regression**: With the parameters $solver =' liblinear'$

2.1.2 Features

- **TfIdf**: In order to learn the language model of Trump, features that represent the way Trump express himself are important. Therefore, the $TfidfVectorizer$ of $sklearn$ package was used, with $max\_features$ of 9000 (best ngrams), and with a $tokenizer$ that performs text normalization as in previous assignment, with filtering of stop words (from Nltk).

- **Weekday**: Getting Trump schedule and habits, by the week days. I thought the most active days of week will help to get it.

- **Elections**: Distinguishing tweets that were tweeted before and after the elections day (November 8,2016), by notating each tweet. I thought the tweets nature will be changed before and after this day.

- **Hour**: Getting Trump schedule and habits, by the hours in day. Again, assuming the most active hours in day will help to get it.

- **Sentiment**: In order to learn the tweeter sentiments from tweets, the $SentimentIntensityAnalyzer$ of $Nltk$ package was used. The analyzed tweets were lower case and filtered from web links, punctuation and stop words were not filtered since I thought it is important for the sentiment.

- **Clustering on embedding vectors**: In order to learn the tweets semantic, the average word embedding for tweet was not contributing enough for the classification (I guess it is because Trump and his staffers tweet on similar occasions). Therefore, $KMeans$ clustering of $sklearn$ package was used, with

Model	Auc	F1	Precision	Recall	Accuracy
RNN	85%	**88**%	**90**%	**88**%	**88**%
CNN	79%	86%	89%	87%	87%
SVM rbf	84%	86%	89%	87%	87%
SVM linear	**86**%	**88**%	88%	**88**%	**88**%
Random Forest	84%	87%	89%	87%	87%
Logistic Regression	85%	87%	89%	**88**%	**88**%

Table 1: Results for 80:20 method

Model	Auc	F1	Precision	Recall	Accuracy
SVM rbf	80%	82%	**86**%	83%	83%
SVM linear	**83**%	**84**%	**86**%	**85**%	**85**%
Random Forest	82%	**84**%	85%	**85**%	84%
Logistic Regression	81%	83%	**86**%	84%	84%

Table 2: Results for time-cross validation method

$n\_clusters$ of 4. The embedding vectors, as presented in section 1, were taken from `google news` with *gensim* package, and contained only word vectors from the training data. The tweets were normalized, and stop words were filtered the same as in TfIdf.

2.1.3 Training Data Method

The splitting of the data matched two cases. The first, I used $TimeSeriesSplit$ of $sklearn$ package with n_splits of 2. It allowed me to perform cross validation on time data (cant be shuffled). The second, 80:20 as will explained in 2.2.3, so I could compare the ML models with the DL models (train and test on the same data).

2.2 Deep Learning

2.2.1 Models

- **Convolutional Neural Network**: Contains initialized Embedding layer (detailed in 2.2.3, three convolutional layers with 100 filters each, Dropout with probability of 0.5, followed by Linear layer with a sigmoid activation.

- **Recurrent Neural Network**: Contains initialized Embedding layer (detailed in 2.2.3, LSTM with 2 layers, Dropout with probability of 0.3, followed by Linear layer with a sigmoid activation.

2.2.2 Features

The features of these networks were the tweets, after encoding each word for a tweet and padding each tweet according to the longest tweet length.

2.2.3 Training Data Method

Since neural networks need a lot of data in order to train, and as mentioned above- since I want to avoid future-looking when I train the model, I split the time-sorted data into 80% for the train set and 20% for the test set. Otherwise, by another splitting according to time methods (cross validation), the first splits will not contain enough data. Both of the architecture Embedding layers were initialized a pre-trained weight matrix, which was calculated from the $googlenews$ embedding vectors according to the training data. In addition, I split the data into train validation and test sets, by relation of 80-10-10, to find the best hyper-parameters. The different parameters were examined are learning rate, number of layers, number of epochs, descending loss on the validation set and pretrained word-embedding vs. trained from scratch.

3 Algorithms Comparison

The data was split as mentioned in 2.2.3, 2.1.3, according to the type of the model. In case of machine learning models both method have been tested, one for comparing with NN models, and the other to verify, and compare the performance between the different ML models.

The results of all models were trained in 80:20 method presented in 1. The results of ML models using CV presented in 2.

As detailed above, the best models are RNN and SVM with linear kernel. I chose to SVM as my best model, although both models are equivalent.

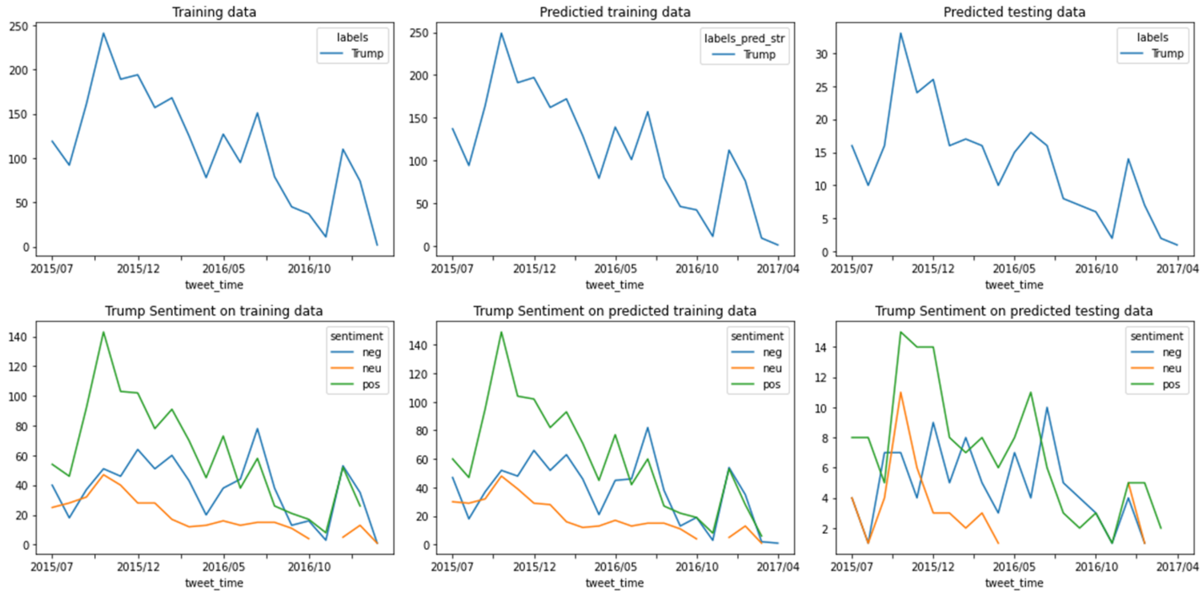

Figure 5: Best model analysis

3.1 Was Trump Kept away from his campaign?

In order to asses whether Trump was kept away during the campaign I predicted the testing data and training data (again), then I performed sentiment analysis, for Trump only. The results are presented in Fig 5. The predictions of the model behave the same as in the training data, i.e. we can see reduction in trump Twitter activity during the campaign time. We also can see in that time a reduction in the positive sentiment tweets, and an increasing in negative sentiment tweets. Moreover, all of the graphs show reduction in Trump activity few months before the election day. So, as i explained in section 1, I assumes that trump was kept away from Twitter during his campaign time, and that my best model reflects it.

4 Conclusions

I created a model that detects given a Twitter activity whether its Trump activity or not. To do it, I explored several models algorithms, and architectures. There was not a big difference in the examined models performances. yet, the performances of two of them were better. The chosen model for this mission is SVM with linear kernel.

From the training phase my main conclusions are:

(1) **Machine Learning**: The use of indicative bias is very important to determine our features correctly. For ablation test we tried the ML models without the clustering on word-embedding features and we received lower score (- 0.2 in auc, - 0.1 in f1, - 0.2 in accuracy).

(2) **Deep Learning**: It is important to use validation set in order to obtain the best hyper parameters for the model.

In addition, from 3.1 we conclude that Trump was kept away from his campaign.