The Effect of Donald Trump's Tweets on USDX

Barış Sermet
Sabancı University

Yuşa Ergüven
Sabancı University

Abstract

Since the social media can accessible to everyone, politicians use it more and more each day. In this paper we are going to analyze Donal Trump's tweets after he has been selected as President of the United States and their effects on USDX, which is and index of the value of the US dollar, to observe whether there is a correlation or not.

1. Introduction

Means of propaganda has always been important for the economy since it affects the behaviours of people in investment. Over time with the emergence of social media, social media platforms also found places in the propaganda field. Government officials and leaders are able to touch each individual easier than ever before with the help of social media. Due to this intense and reactive communication channel, people tend to show immediate reactions. The impacts of this reactive communication on economics can be observed. In addition to these, there has been research on the effect of presidential social media actions to stock market (Ge, Kurov and Wolfe, 2019) and more specifically Donald Trump's tweets(Rayarel, 2018). On the other hand there is also a research on investors' approach to market according to social media activity of governmental level users(Ge, Kurov and Wolfe, 2019).

In addition to the afore-mentioned research, we are going to analyze Donald Trump's tweets after he has been selected and their effects on USDX, which is an index of the value of the US dollar, to observe whether there is a correlation or not.

The U.S. Dollar Index is calculated with this formula[4]: $USDX = 50.14348112 \times EURUSD^{-0.576} \times USDJPY^{0.136} \times GBPUSD^{-0.119} \times USDCAD^{0.091} \times USDSEK^{0.042} \times USDCHF^{0.036}$

2. Data Collection

For this project we used 2 dataset; The first one is Donald Trump's tweets between January 20, 2017 to November 31, 2019. Second dataset is USDX changes between January 20, 2017 to

November 31, 2019. Note that 20 January 2017 is the day of Donald Trump is inaugurated as the 45th President of the United States of America.

2.1 Donald Trump's Tweets

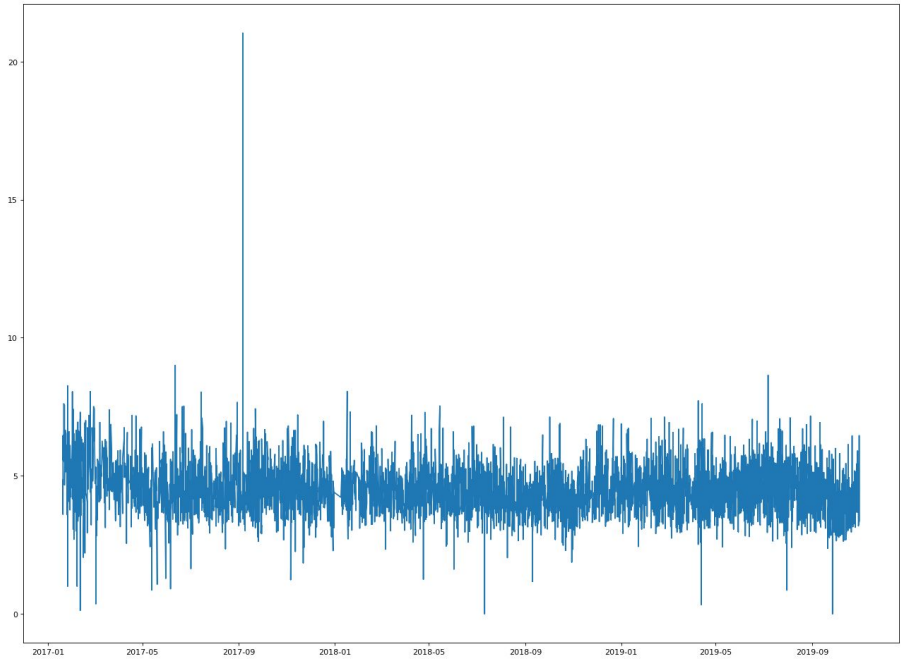
Tweets that Donald Trump posted are taken from Trump Twitter Archive[5]. The dataset contains source(e.g. iPhone, Media Studio), tweet context, data, retweet count, favorite count, is retweet.

2.2 USDX Changes

USDX dataset are taken from investing[6] which contains date and the change(%). We apply change a threshold on USDX. Our threshold is 0.2. So, if current change is bigger than threshold we label it as 1, if it is smaller than -0.2 than -1, else 0.

3. Data Preprocessing

Our collected data are separated, so first we need to match them by their dates. Since USDX is updated whenever U.S. Dollar markets are open, we don't have any change data for weekends. That is why we remove the weekend tweets. Also, we remove punctuation, commonly used words(stopwords) and emojis from our data and normalize the remaining words. Then we convert tweet text to vectors of vocabulary. Our features are the text vector, tweet's retweet and favorite count. We calculate the retweet/favorite ratios as a feature. Then, we calculated the favourite/retweet ratios of tweets. In the following graph, ratios of tweets are given.



Our labels are the daily USDX rate changes based on our threshold. On the first part we tried our labels as 1 (increasing) and 0 (decreasing). Since we have -1 as a label, we used different algorithms. Also we increase our features.

4. Methods

On the first part of the project, we applied Multinomial Naive Bayes and Logistic Regression algorithm. Desired output from both algorithms to predict our label, increasing or decreasing. Since we don't have 2 label classes and our retweet/favorite ratio is continuous we applied Gaussian Naive Bayes, Logistic Regression, SVM, Random Forest to our dataset.

4.1 Bag-Of-Words

A bag-of-words model, is a simple and short way to extracting features from text for using in such models, like machine learning algorithms. We can think the "bag" as a dictionary, it stores every word in a document as a key and the occurrence of that word as a value. That approach is applied for our models.

4.2 Multinomial Naive Bayes

Naive Bayes is a technique used for constructing classifiers where features are stored as vectors. Naive Bayes classifiers are based on Naive approach of Bayes theorem, in which it is assumed the features are independent. Since our task is document classification, we used Multinomial Naive Bayes model which is generally used for document classification such as ham/spam classification, positive/negative comment classification etc.

4.2.1 Maximum Likelihood Estimate (MLE)

After vectorizing our data, we used Multinomial Naive Bayes without any Prior and we achieved the following results:

For 20% of the test data we estimated 2396 predictions and there are 1312 correct, 1084 wrong predictions were made. For MLE, our accuracy was 54.75%.

4.2.2 Maximum a Posteriori (MAP)

After MLE, we also applied MAP with Dirichlet prior and the results are as follows:

For 20% of the test data we estimated 2396 predictions and there are 1329 correct, 1067 wrong predictions were made. For MAP, our accuracy was 55.46%. Since we are estimating with Dirichlet prior, our alpha is 1.

4.3 Logistic Regression

Logistic regression method is a classification method even though its name is regression. It is used in document classification tasks since it is helpful in finding relations between features and classifying the outcome accordingly. So that, we tried that also. We separated dataset to 60% training, 20% validation and 20% test sets. Then, optimized our hyperparameters: learning rate and iteration count. Since we had almost 11.000 features, due to heavy calculations we started the iteration count from 100 to 5000 with increasing the iteration count 100 at each. For the learning rate, we started it from exp(-15) to exp(-5) and split between values to 15. As conclusion, we tried 15 learning rates and 50 iteration count. Result of optimization is as follows:

```
Maximum accuracy: 0.5529583637691746 is achieved via parameters:
Learning rate: 8.571428571428571
Iteration num: 5000
```

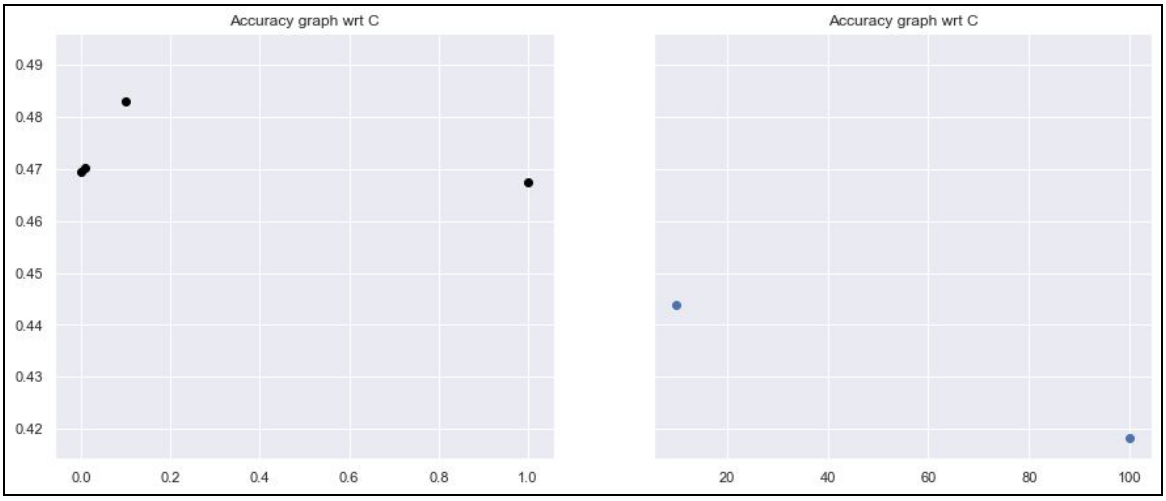
Then with the optimized hyperparameters, we tested our model. Our test results are given below:

```
Learning rate: 8.571428571428571
Iteration num: 5000
Accuracy: 0.5558802045288532
```

After changing our label system and adding new features to our dataset, we again tried Logistic Regression. We both tried LogisticRegression module of the scikit with tuning C parameter and SGDClassifier with loss parameter as log. For the LogisticRegression module, we tuned our C parameter with the following C values:

- 0.001,0.01,0.1,1,10,100

And the results are as follows:

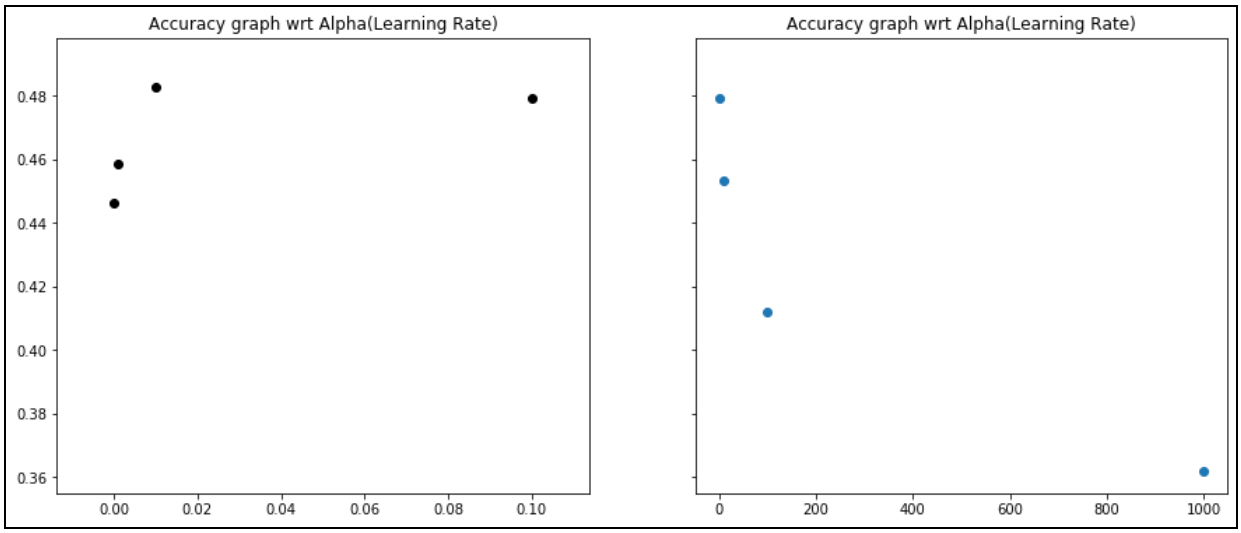


We acquired the best score: 0.483105 with C value: 0.1. After we train our model with original train dataset which includes validation set also, with the C parameter 0.1, the resulting accuracy score is: `0.4747991234477721`

For the SGDClassifier with loss parameter “log”, we picked max iteration count as 1000 and tuned our learning rate with the following values:

- 0.001,0.01,0.1,1,10,100, 1000

Results are as follows:



We acquired the best score: 0.4824717436848637 with Alpha(Learning Rate) value: 0.01.

After we train our model with original train dataset which includes validation set also, with the Alpha parameter 0.01, the resulting accuracy score is: 0.49013878743608474.

4.4 Gaussian Naive Bayes

After we change our label to (-1, 0, +1) and added new non-binary features such as retweet/favourite ration, we no longer were able to use Multinomial Naive Bayes. So we applied Gaussian Naive Bayes to our new dataset, and yet the results weren’t as expected:

`Accuracy: 0.3689497716894977`

We don’t have any optimization to apply in Gaussian Naive Bayes so that, we directly used train and test dataset instead of creating validation set. The result above belongs to the test set.

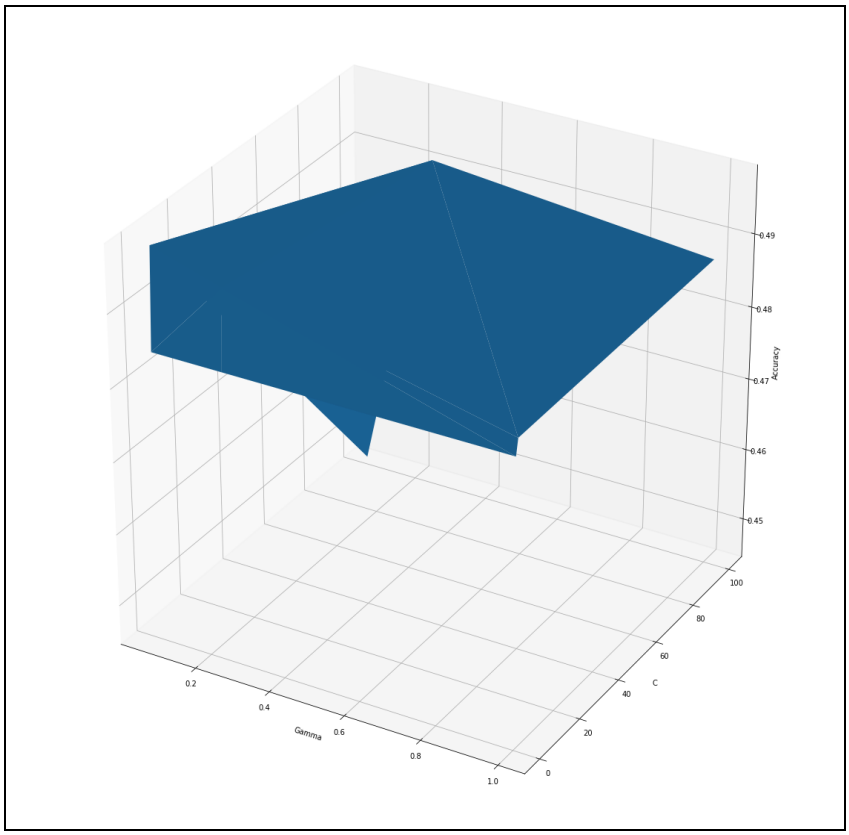
4.4 SVM

We also applied SVM with optimizing the hyperparameters. We tuned the C and gamma parameters with the following values:

- C = 0.0001, 0.01, 1, 100
- Gamma = 0.0625, 0.25, 1

	C	gamma	accuracy
0	0.0001	0.0625	0.484295
1	0.0001	0.25	0.484295
2	0.0001	1	0.484295
3	0.01	0.0625	0.484295
4	0.01	0.25	0.484295
5	0.01	1	0.484295
6	1	0.0625	0.498174
7	1	0.25	0.495252
8	1	1	0.486486
9	100	0.0625	0.445581
10	100	0.25	0.490869
11	100	1	0.487217

The accuracies can also be seen in the following graph:



Then we selected the parameters with highest accuracy and trained the model with it and tested:

```
C: 1,  Gamma: 0.0625
```

Test results are as follows:

```
Accuracy: 0.5178962746530315
```

4.5 Random Forests

Random forests are ensemble methods that are used for classification, regression problems which are based on decision trees. Since our problem is also classification problem, we applied random forests with optimizing the following parameters with the following values.

- Bootstrap = True, False
- Max_depth = 10, 60, 110
- Max_features = auto, sqrt
- Min_samples_leaf = 2, 4
- Min_samples_split = 2, 5, 10
- N_estimators = 200, 1100, 2000

We used RandomizedSearchCV with 10 iteration count and 4 folds. The results are as follows:

Best Parameters:

```
{'n_estimators': 200,
 'min_samples_split': 5,
 'min_samples_leaf': 2,
 'max_features': 'sqrt',
 'max_depth': 110,
 'bootstrap': False}
```

Best Score:

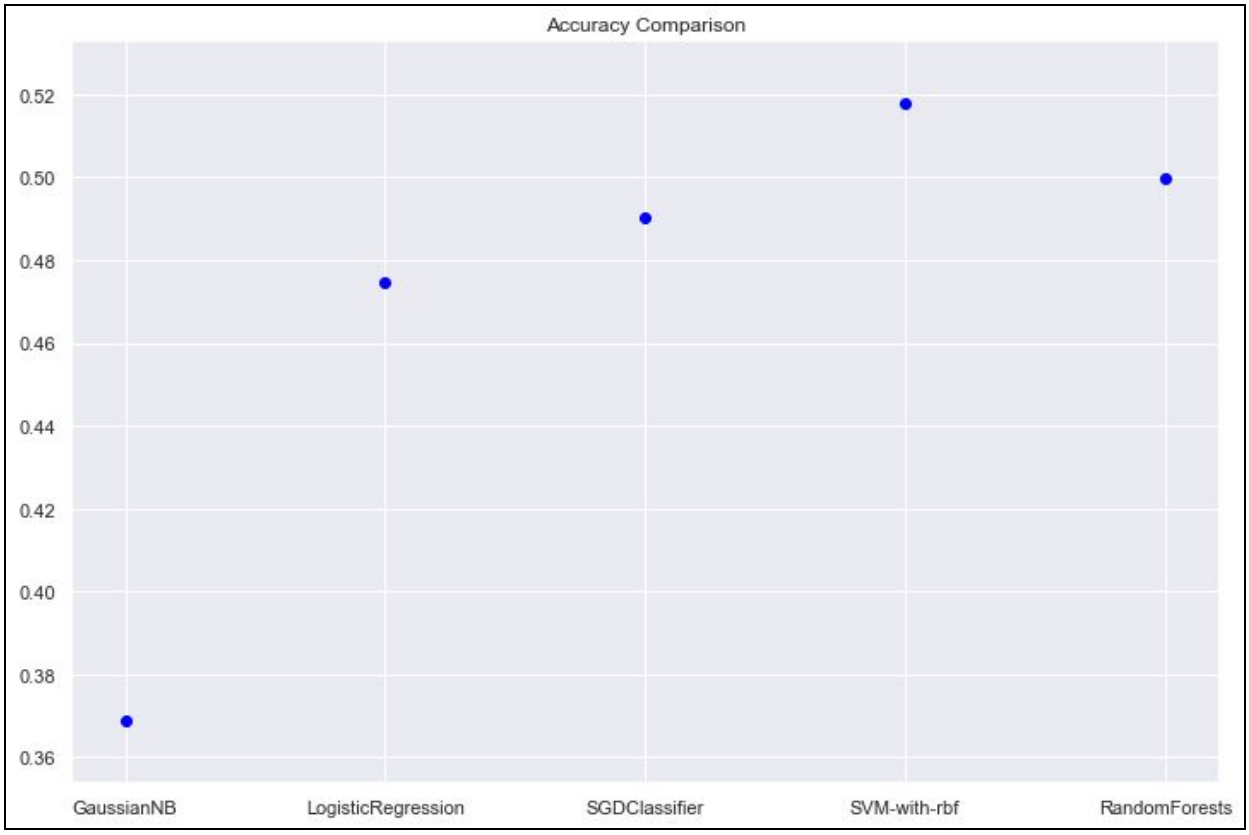
```
0.48412879652888785
```

Then we trained our model with the original train set and predicted the labels of the test set with the optimized parameters. The accuracy we acquired is as follows:

```
0.4996347699050402
```

5. Discussion

The comparison of the algorithms we tried can be seen in the following graph.



Even though our accuracy scores are not really high compared to random guess accuracy 0.33, there is a small improvement. It is clear SVM with Radial basis function kernel gives us the best accuracy among them. For the Naive Bayes method we should have re-implement the algorithm instead of just using Gaussian due to our continuous features, so that we combine both Multinomial and Gaussian accordingly to the features. The main reason GaussianNB gives such a low accuracy is related to that. SGDClassifier with “log” as loss parameter is basically Logistic Regression with Gradient Descent, and Logistic Regression is linear classifier while SVM with RBF kernel is non-linear classifier. It is possible the accuracy difference is due to the fact that our data is not linearly separable and while SVM with RBF kernel could handle non-linearities in the data, Logistic regression can’t. Since we used bag-of-word approach to extract features, our data is highly linear. Lastly for the random forest, since tuning the hyperparameters takes a lot of time, we might have not found optimum parameters. Based on the fact that the difference between random forests and SVM are not too much, it might be due to our hyperparameter selection. Yet, it doesn’t shows that if we tune the hyperparameters of random forest, we may get better results.

6. Conclusion

According to the models we trained, we haven't achieved a result that supports the correlation we wanted to look for in the first place, yet. Our training models' accuracy metrics states based on the keywords in the tweet, retweet/favorite ratio, tweet contains URL or not and tweet contains uppercase letter or not, then we can estimate the change in the USDX rate of the day with around %50 probability. Since we have 3 labels, if we randomly select the probability of guessing the correct label is %33. In this project we achieve around %50 probability which is higher than %33. Based on the features, labels we picked and the machine learning algorithms we applied, the results have been better if we choose our feature set more specific. For now, our model predicts the labels better than randomly selecting.

7. References

1. Ge, Q. and Wolfe, M. (2017). Stock Market Reactions to Presidential Social Media Usage: Evidence from Company-Specific Tweets. SSRN Electronic Journal .
2. Ge, Q., Kurov, A. and Wolfe, M. (2019). Do Investors Care About Presidential Company-Specific Tweets?. Journal of Financial Research , 42(2), pp.213-242.
3. Rayarel, K. (2018). The Impact of Donald Trump's Tweets on Financial Markets. [online] Available at: https://www.nottingham.ac.uk/economics/documents/research-first/krishan-rayarel.pdf [Accessed 3 Nov. 2019].
4. U.S. Dollar Index® Contracts . (2018). P. 2. Available at: https://www.theice.com/publicdocs/futures_us/ICE_Dollar_Index_FAQ.pdf [Accessed 3 Nov.2019]
5. Trumptwitterarchive.com. (2019). Trump Twitter Archive . [online] Available at: http://www.trumptwitterarchive.com/archive [Accessed 3 Nov. 2019].
6. investing.com. (n.d.). US Dollar Index . [online] Available at: https://www.investing.com/currencies/us-dollar-index [Accessed 3 Nov. 2019].