CS584 – MACHINE LEARNING

FALL 2016

PRESIDENTIAL CAMPAIGN ANALYSER

Group Members: Himanshu Singh, Harshit Singh, Sagar Arora

Table of Contents

PRESIDENTIAL CAMPAIGN ANALYSER

Group Members: Himanshu Singh, Harshit Singh, Sagar Arora

Task

On the basis of selected Hashtags which are related to the presidential Campaign tweets are collected.After the collection of tweets we manually classified the tweets into four classes namely Hillary_negative, Hillary_positive, Trump_positive, Trump_negative. After the collection of tweets, we did sentiment analysis and then predict who out of the two candidates namely Hillary Clinton or Donald Trump will most likely win. This is useful because if we have a such model, we can use it to predict the result of the election.

Dataset

We collected around 7000 tweets after cleaning and manually labelling it we were left with around 1100 tweets which were classified into the four distinct classes. The tweets were collected based on selected hashtags which were specific to the presidential candidates and election 2016 respectively.

Data source

Data source was twitter we used twython api to interact with twitter and get the desired data that is tweets and yes, we labelled all the tweets into the four classes manually.

Target variable

We divided the tweets into the four classes that is Hillary_negative, Hillary_positive, Trump_positive, Trump_negative.

Features

Input features were the tokens that we created through the tokenizer function and then fed that into the countvectorizer which created a feature matrix of the tokens which were created from the text of the tweets. And we had around 3578 features to work with.

Data size

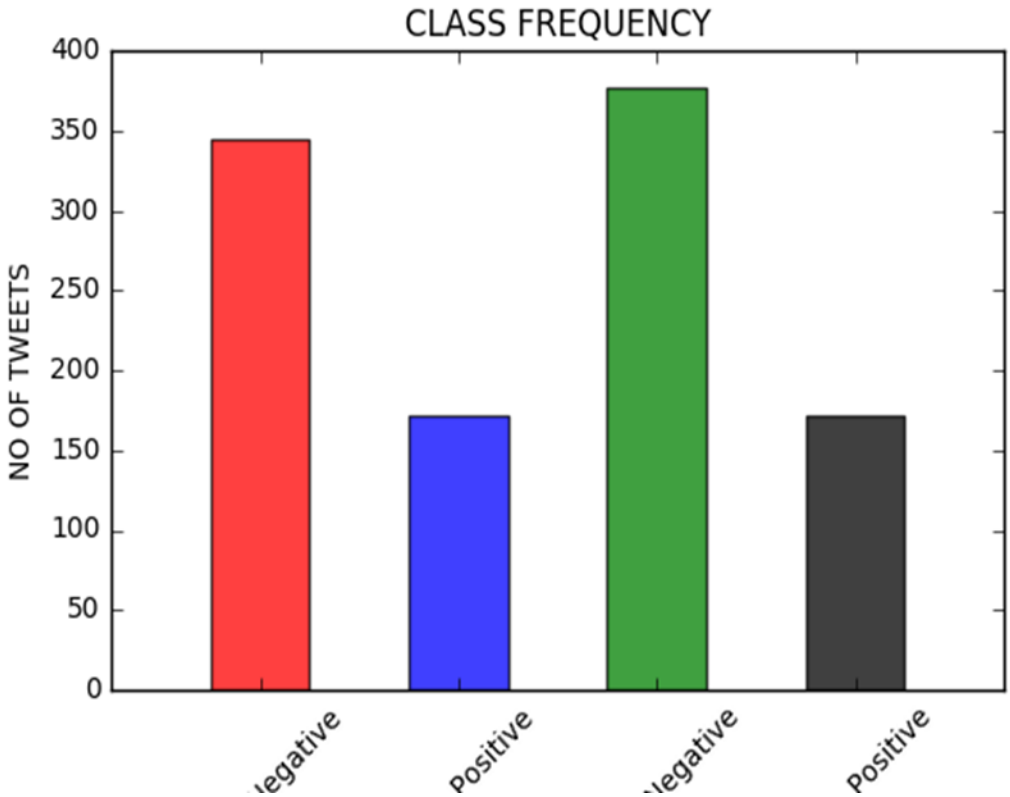
We had 1045 instances.

Preprocessing

The main process of preprocessing was to collect the data then clean the data and then classify the data into the target variables before fitting the data into the matrix. Because we wanted to get the best result so we chose to manually label the data so we can avoid the cases like if somebody has used sarcasm in their tweets and also to make sure that misclassification was minimum.

Visualization

Target



STATISTICS:

Hillary_negative: 340

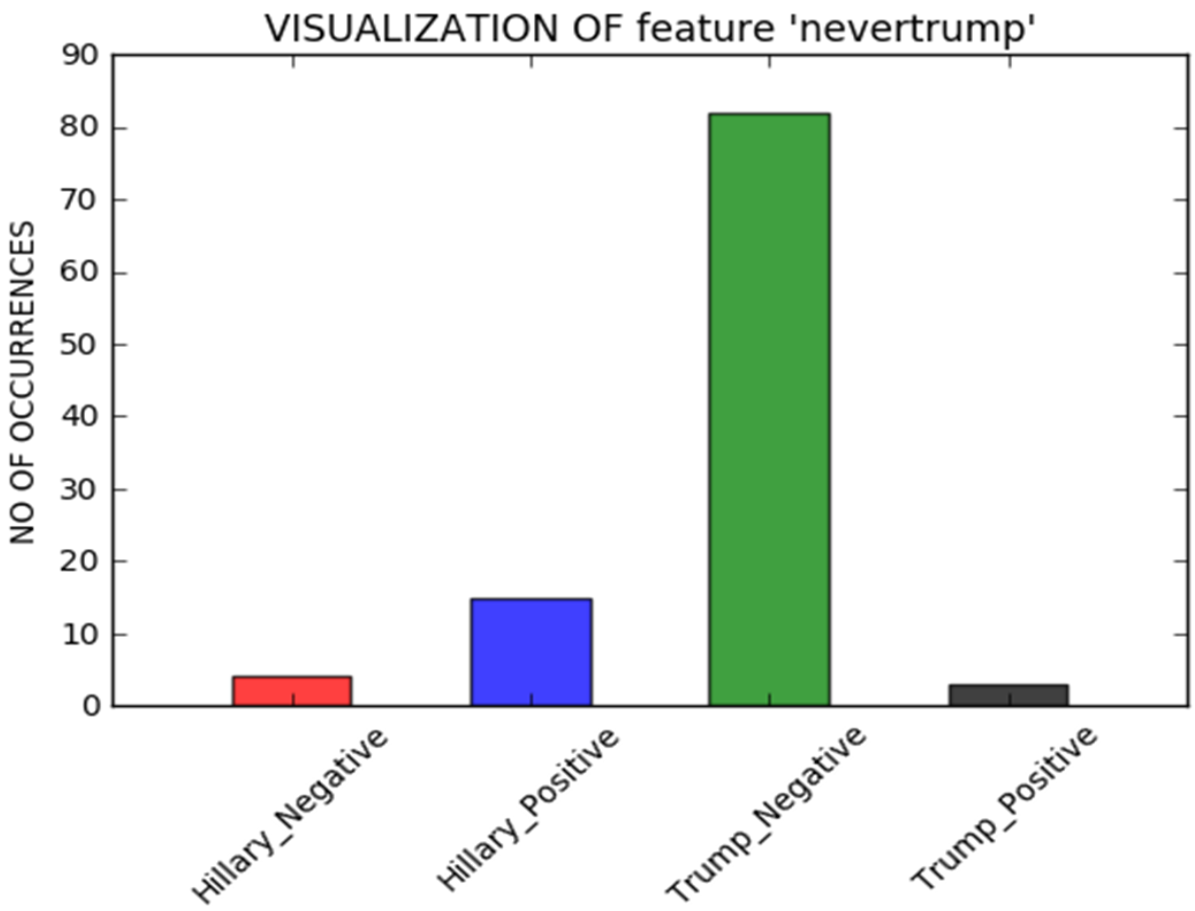
Hillary_positive: 165

Trump_positive: 150

Trump_negative: 390

Features

Visualizing one frequent feature using histogram:



List of features occurring frequently

```
Word: hillary IDF: 3.2253
Word: nevertrump     IDF: 3.2988
Word: hillaryclinton IDF: 3.3576
Word: imwithher      IDF: 3.4094
Word: dumptrump      IDF: 3.5707
Word: hillaryforprison2016  IDF: 3.5833
Word: realdonaldtrump       IDF: 3.7042
Word: antitrump      IDF: 3.8584
Word: neverhillary   IDF: 3.9097
Word: crookedhillary IDF: 4.0209
```

Evaluation

Performance Measure

We chose accuracy score as our performance measure. We chose it because we couldn't use precision and recall as it for multilabel classification and considering that we had to sentiment analysis on the text so accuracy score was probably the best measure we could've chosen.

Classifiers

We used four classifiers:

Decision Tree

Multinomial Naive Bayes

Logistic Regression

Support Vector machine

We tried over 50 combinations of the random state in test train split and for each of those 50 iterations we modified C parameter of each model with 100 different values. And after evaluating all that we got our best model that was Logistic regression.

Evaluation Strategy

We used both test train split and cross validation in our project. We used cross validation to choose which classifier was performing best. And we used train and test split to split the data for training and testing purposes as we were doing sentiment analysis and we couldn't use new data because it's not possible to get the same features from the new data so that's why we used test train split model so we can train our model and test its accuracy score using same data.

Performance Results

Best model and parameter setting was Logistic Regression with setting C=0.927

Model	Parameters	Performance
Baseline	Majority class	0.3550
	Random class	0.2488
Logistic regression	Penalty=l2, C=0.927	0.68
	Penalty=l2, C=0.936	0.677
Multinomial NB	alpha=0.1	0.61
	alpha=0.11	0.609
Decision Tree Classifier		0.62
		0.61
Support Vector Machine	C=0.1	0.372
	C=0.11	0.369

Top Features

HILLARY_NEGATIVE

top_positive : [('fuckhillary', 1.0391729218287471), ('hillno', 1.1261598138807012), ('neverhillary', 1.2370160401848656), ('crookedhillary', 1.4957150854287564), ('hillaryforprison2016', 1.9608260015059567)]

top_negative : [('nevertrump', -1.0688001463309191), ('donaldjtrump', -1.0027422708884994), ('imwithher', -0.72634097103149919), ('dumptrump', -0.70192367014796619), ('please', -0.66488857982664495)]

HILLARY_POSITIVE

top_positive : [('clinton', 0.83556960073895281), ('hilaryclinton', 1.0981688433672858), ('hilary', 1.1110476941705234), ('hillaryclinton', 1.2703339564566791), ('imwithher', 1.9222054980580123)]

top_negative : [('fuckhillary', -0.92470246344094098), ('realdonaldtrump', -0.78001830496729418), ('trump', -0.690017547869617), ('if', -0.62635794074978202), ('fucktrump', -0.60584290854403666)]

TRUMP_NEGATIVE

top_positive : [('fucktrump', 0.74508899762659053), ('drumpf', 0.77521009022233578), ('dumptrump', 1.1964711383256146), ('antitrump', 1.4692689375159103), ('nevertrump', 1.62274653830354)]

top_negative : [('hillary', -0.94267799511891681), ('crookedhillary', -0.81752992465101693), ('hillaryforprison2016', -0.81154458096033211), ('win', -0.77991598548934749), ('clinton', -0.77637200471320189)]

TRUMP_POSITIVE

top_positive : [('donaldtrump2016', 0.91478288895550963), ('votetrump', 0.93211281908970334), ('trump2016', 0.97360376242979041), ('realdonaldtrump', 1.1776225944997827), ('donaldjtrump', 1.4849706916361471)]

top_negative : [('dumptrump', -0.81021873023040869), ('antitrump', -0.63708578791275405), ('hilaryclinton', -0.62576717178132302), ('hilary', -0.59209405065500786), ('nevertrump', -0.58418865020030153)]

Discussion

We think that the best model worked according to our expectation and we think that we tried enough combination to find the best setting but I think the reason for low accuracy score maybe because of the data that we collected wasn't enough as we selected specific hashtags so the data collected was also restricted and since in sentiment analysis we are bound to get classification error in testing phase because models don't have vast vocabulary to understand the real emotions of the people because for

the model it is just a word but people can be sarcastic or they don't mean what they write. And also, we think that the tokenizer function that we used wasn't good enough we tried to tweak that also by trying different regular expressions to make the tokens from the tweets but we got the best function from all the combinations that we used but we still think that more combinations can be there but since we had limited amount of time we could just test those combinations only. Considering all these restrictions we think that our model did fairly well

Interesting/Unexpected Results

List of top errors

'Content': "Pls America don't make the most stupid mistake of history vote for #HilaryClinton Trump is a child who needs help Trump is a symbol of hate"

'predicted': 2,

'truth': 3

'Content': "when i look at the polls and hillary's winning #HillaryForPrison2016"

'predicted': 2

'truth': 0

'Content': 'More terrorist violence from #Hillary #Clinton supporters... #ElectionDay'

'predicted': 1

'truth': 2

'Content': 'This is criminal. #VoteTrump #MAGA #NeverHillary'

'predicted': 0

'truth': 3

'Content': '#fuckstupidity'

'predicted': 0

'truth': 1

Contributions of Each Group Member

<If you are working in a group, please discuss in detail what each member did for this project.>

Phase 3		
1	Model Selection	Sagar (Logistic Regression, MultinomialNB), Harshit (Support Vector Machine Classifier) Himanshu (Decision Tree Classifier)
2	Visualization	Harshit
3	Evaluation	Sagar, Harshit, Himanshu
4	Documentation	Harshit, Himanshu
5	Final Evaluation and merging	Sagar, Himanshu
Phase 4		
1	Documentation	Harshit, Himanshu, Sagar

The project was split as follows:

Each one of us had a topic, we discussed and what all we can do in the project.

We finalized Himanshu's project at the end.

Sagar worked on two model that is logistic Regression and Multinomial Naïve Bayes

Harshit worked with Support vector machine classifier.

Himanshu worked with decision tree classifier.

We all combined our results and came up with the conclusion which the best score.

After that we tried multiple combinations in Logistic Regression and found the accuracy, precision and recall. Also the top results for all classes were reported.

Conclusion

The result of our project shows that Hillary Clinton will be the most likely candidate to win the presidential election. And in the closing remarks we would like to say that we think that this project could've been more accurate if we had some more time and more apt data for the model to learn from.

References

<Provide references if you have any.>

1. https://twython.readthedocs.io/en/latest/#twython-api-documentation
2. https://dev.twitter.com/rest/reference
3. http://scikit-learn.org/stable/supervised_learning.html#supervised-learning
4. http://scikit-learn.org/stable/modules/svm.html
5. http://scikit-learn.org/stable/modules/linear_model.html#bayesian-regression
6. http://scikit-learn.org/stable/modules/cross_validation.html.
7. http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex2/ex2.html