SAINT PETERSBURG NATIONAL RESEARCH UNIVERSITY

OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS”

(ITMO UNIVERSITY)

|  |  |
| --- | --- |
| Faculty | Institute of Urban Studies and Design |
|  |  |
| Field of education | Applied Mathematics and Informatics |

**R E P O R T**

**On Scientific Research Work**

**The task title:** **Prediction of reactions to informational messages in social networks**

**Student Shamsi Goushki, Satar C4111c**

**ABSTRACT**

This paper contains 18 pages, 3 figures, 5 table**.**

**Prediction of reactions to informational messages in social networks.**

As developing in social network huge amounts of informational messages are published daily. Prediction of people reaction to such data is extremely important. In this paper, we will introduce different methods and machine learning techniques to predict people reaction.

Firstly, we will introduce different realm of prediction via social network and different method for predicting. For our research we have chosen Twitter for our data set which include 3 data sets with different sizes. Also, we have implemented and predicted people reaction with 5 methods including Bayes classifier, K-nearest neighbor classifier decision tree and random forest. Eventually, results illustrate performance of each method for our prediction.

**Introduction**

Social media allow people to publish their contents. Both Facebook and Twitter are one of the top 10 most-visited websites in the world according to Alexa ranking. The amount of information pieces created and transformed via Twitter is 140 million per day by Mach 2011. Since there are many users sharing their opinions and experiences via social media, there is aggregation of personal wisdom and different viewpoints. Such aggregation has limitations as viewpoints are subject to change with time. Most of the prediction utilized time series to predict future base on the past events. Such prediction has great benefits in many realms, such as finance, product marketing and politics, which has attracted increasing number of researchers to this subject.

But another way of prediction is about people reaction to informational message. In such cases usually we do not use time series and our end is predicting reaction base on the previous data. In rest of our we introduce some usage of social media prediction, methods for modeling, implementation and results.

**Realms of prediction subjects via social networks:**

* Marketing
* Movie box-office
* Information dissemination
* Elections
* Macroeconomic
* Miscellaneaon

Since our end is implementing and focusing on implementation for reaction, I just mentioned several subjects in social media prediction.

**Data set**

Firs of all for every prediction we need one or several data sets, so it is our initial task to collect data. there are several API to collect data from social networks. Another way is using open source data sets which are published for different purposes including educational activity. We have used several data sets from Standford University which totally include 1.6 twits and their reaction.

**Prediction methods**

There are several methods for prediction including:

* Regression method
* Bayes classifier
* K-nearest neighbor classifier
* Decision tree
* Random forest
* Model based prediction
* Artificial Neural network

Within our implementation we have focused on five first methods including logestic regression, Bayes classifier, K-nearest neighbor classifier decision tree and random forest. In next pars we observe how to implement these methods and analyze result obtained from different methods.

**Implementation**

In order to predict reaction to new informational message based on the previous history we need to split our implementation into 2 different parts, first is preprocessing data which clean our data set and remove unnecessary data. next step is sentiment analysis which perform a model for our prediction.

**Data set:**

For our implementation, we have utilized an open source dataset from Twitter which can be found within following link:

<http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip>

By looking at the description of the dataset from the link, the information on each field can be found.

0 — the polarity of the tweet (0 = negative, 4 = positive)

1 — the id of the tweet (2087)

2 — the date of the tweet (Sat May 16 23:58:44 UTC 2009)

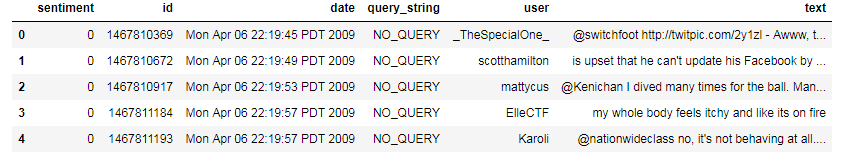
3 — the query (lyx). If there is no query, then this value is NO\_QUERY.

4 — the user that tweeted (robotickilldozr)

5 — the text of the tweet (Lyx is cool) Preprocessing data

Our data set include 3 data set with different sizes and totally 800000 twits with label of 0. Also, 800000 twits with 1 label that mean totally our data sets include 1.6 million twits with their reaction. 50% of the data is with negative label, and another 50% with positive label. Table 1 shows the top 5 rows of our first data set:

Table 1: data set overview



We just need text and sentiment column, so firstly we drop other columns.

**Data Preparation 1: HTML decoding**

It looks like HTML encoding has not been converted to text, and ended up in text field as ‘&amp’,’&quot’,etc. Decoding HTML to general text will be my first step of data preparation. I will use BeautifulSoup for this.

**Data Preparation 2: ‘@’mention**

The second part of the preparation is dealing with @mention.

Even though @mention carries a certain information (which another user that the tweet mentioned), this information doesn’t add value to build sentiment analysis model.

**Data Preparation 3: URL links**

The third part of the cleaning is dealing with URL links, same with @mention, even though it carries some information, for sentiment analysis purpose, this can be ignored.

**Data Preparation 4: UTF-8 BOM (Byte Order Mark)**

By looking at the above entry, I can see strange patterns of characters “\xef\xbf\xbd”. After some researching, I found that these are UTF-8 BOM. “The UTF-8 BOM is a sequence of bytes (EF BB BF) that allows the reader to identify a file as being encoded in UTF-8.”

By decoding text with ‘utf-8-sig’, this BOM will be replaced with Unicode unrecognizable special characters, then I can process this as “?”.

**Data Preparation 5: hashtag / numbers**

Sometimes the text used with hashtag can provide useful information about the tweet. It might be a bit risky to get rid of all the text together with the hashtag. So, I decided to leave the text intact and just remove the ‘#’. I will do this in the process of cleaning all the nonletter characters including numbers.

**Defining data cleaning function**

With above five data cleaning tasks, I will first define data cleaning function, and then will be applied to the whole dataset. Tokenization, stemming/lemmatization, stop words will be dealt with later stage when creating matrix with either count vectorizer or tf–idf vectorizer.

**TFIDF**

In information retrieval, tf–idf or TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.[1] It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. tf–idf is one of the most popular term-weighting schemes today; 83% of text-based recommender systems in digital libraries use tf–idf.

**TF: Term Frequency**, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

**IDF: Inverse Document Frequency**, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

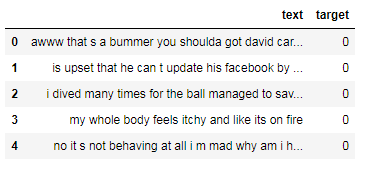
IDF(t) = log\_e(Total number of documents / Number of documents with term t in it).

**Saving cleaned data as csv**

Ultimately, our cleaned data set is ready, and we will save them as a csv file for sentiment analysis. In our new data sets label 4 is replaced by 1.

Table 2 illustrate top five rows of cleaned data:

Table 2 result of cleaned data



**Bayes classifier**

In this section we just review on Bayes classifier and other method implementation is like this just parameter of classifier must be defined. To get the frequency distribution of the words in the text, we can utilize the nltk.FreqDist() function, which lists the top words used in the text, providing a rough idea of the main topic in the text data, here is result for top 50 common word used:

['i', 'to', 'the', 'a', 'my', 'it', 'and', 'you', 'is', 'in', 'for', 's', 't' of', 'on', 'that', 'me', 'so', 'have', 'm', 'but', 'just', 'with', 'be', 'at' not', 'was', 'can', 'this', 'now', 'up', 'good', 'day', 'all', 'out', 'get', 'like', 'are', 'no', 'go', 'got', 'today', 'do', 'too', 'your', 'work', 'love', 'going', 'we', 'what']

Results after tokenizing:

['awww', 'bummer', 'shoulda', 'got', 'david', 'carr', 'third', 'day', 'upse’, 'update', 'facebook', 'texting', 'might', 'cry', 'result', 'school', 'toda’, 'also', 'blah', 'dived', 'many', 'times', 'ball', 'managed', 'save', 'rest'

'go', 'bounds', 'whole', 'body', 'feels', 'itchy', 'like', 'fire', 'behaving’ , 'mad', 'see', 'whole', 'crew', 'need', 'hug', 'hey', 'long', 'time', 'see’, 'yes', 'rains', 'bit', 'bit', 'lol']

**Building a Classifier**

After cleanup, it is time to build the classifier to identify sentiment of each twit reaction. Since our data set is huge enough, I used 80% for train and 20% for test data. (120000, 57314) (30000, 57314) training and test data set with size and features respectively for our first data set. There are many algorithms to choose from, I use a basic Naive Bayes Classifier and train the model on the training set. Our classifier accuracy is about 77% for almost data sets.

I have implemented the text processing techniques used in NLP in detail. I also demonstrated the use of text processing and build a Sentiment Analyzer with classical ML approach achieved fairly good results.

Bayes Classifier is one of the best classifiers for sentiment analysis also I have tested our prediction by other famous methods including random forest classifier, K nearest neighbor classifier, decision tree classifier and logistic regression.

**Results**

For our first data set which include 300000 twits and half of them include positive reaction and half negative reaction result is as table 3 for different classifiers. Also, our test data set include 20 percent of our data or 60000 twits.

Table 3 results of different classifiers for data set include 300000 twits(first dataset)



As we observe logistic regression, Naive Bayes and random forest are the best classifier with accuracy about 76 percent but k nearest neighbor and decision tree performance is not good. In fact, K nearest neighbor is the worst in our classifiers and figure 1 illustrate heat map of confusion matrix. Heat map shows that K nearest neighbor algorithms is useful just for positive reaction in twits and classify them in high performance while for negative reaction in twits its performance is lower than fifty percent.

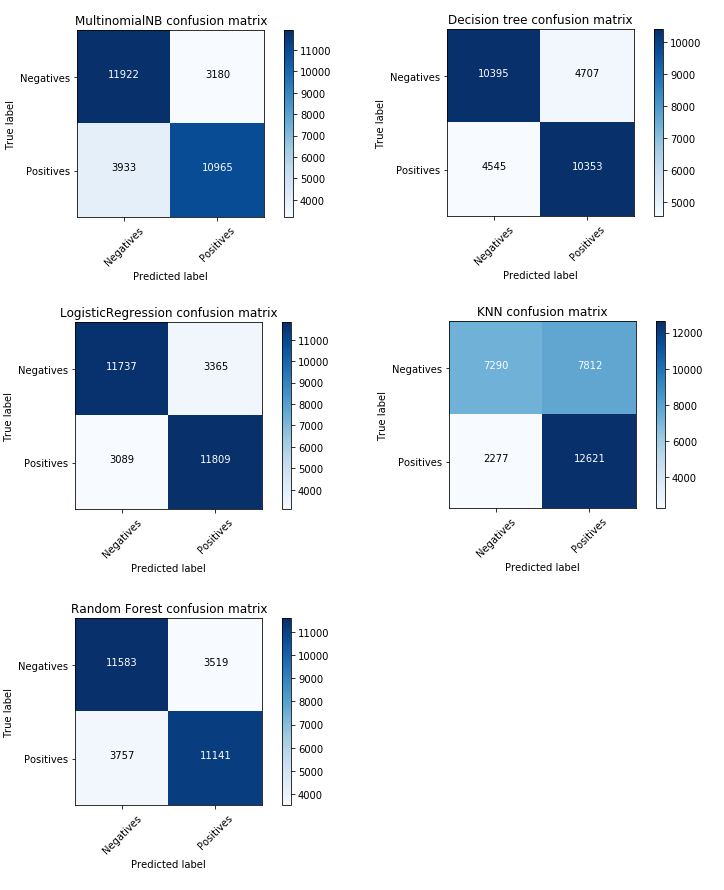


Figure 1 confusion matrix of different method for first data set including 300000 twits

Table 4 results of different classifiers for data set include 500000 twits (second dataset)



Table 4 show results for second data set which is about Twitter too, but number twits are 500000 and train set include 100000 twits. As previous data set best classifiers are logistic regression, Naive Bayes and random forest. The worst classifier is K nearest neighbor as previous modeling. Since number of twits are more than previous section results is a little better.

Figure 2 illustrate heat map of confusion matrix. Heat map shows that K nearest neighbor algorithms is useful just for positive reaction in twits and classify them in high performance while for negative reaction in twits its performance is lower than fifty percent. As previous modeling K nearest neighbor is not a good choice for our prediction. In spit of previous results negatives prediction accuracy is high about 90 per cent while positive prediction is about 35 percent.

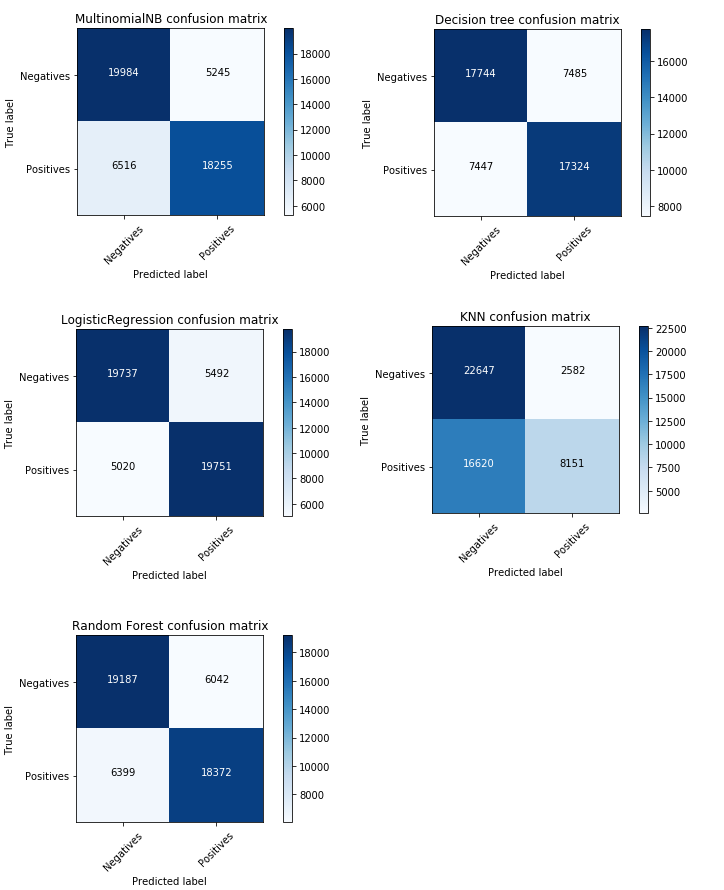


Figure 2 confusion matrix of different method for second data set including 500000 twits

Ultimately, our last data set include 800000 twits, half of them include positive reaction and half negative. We have tested and implemented all previous classifiers and result almost are similar.

Table 5 results of different classifiers for data set include 800000 twits (third dataset)



Totally, the result of different method for data sets are close to each other. And best classifier is logistic regression while K nearest neighbor is not sufficient for our datasets.

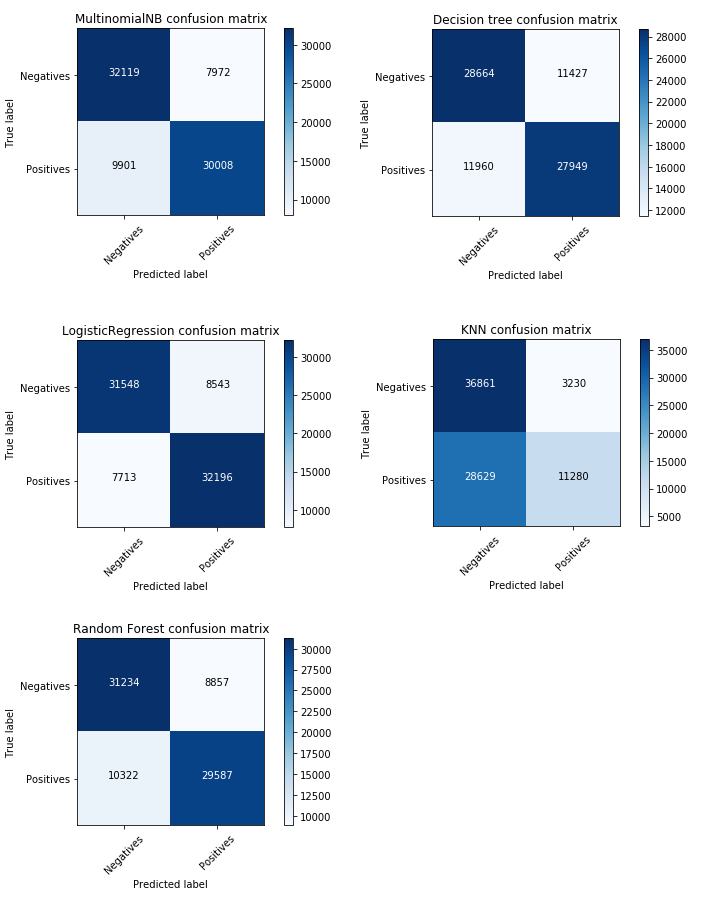


Figure 3 confusion matrix of different method for third data set including 800000 twits

**Related future work**

There are several methods for prediction of data sets. I have implemented classic machine learning methods for our prediction. Also, there are a lot of more methods which utilize neural network. In our next work we will work implement neural network for different data sets and compare results with different classical methods. Also, we have implemented our classifier by default parameter, we will work on parameter to improve our prediction in future.

**conclusion**

In this paper, we have implemented several methods for people reaction to informational message for 3 data set via Twitter which their number of twits is different. Result obviously illustrate different between methods. There three main part for implementation including gathering data, data preparation and modeling. Every part could affect our results, so it is extremely important to choose the best way for all 3 parts. One the most famous classifier for reaction prediction is Naive Bayes, but for our data sets logistic regression classifier performance is the best with accuracy about 80% while K nearest neighbor is not suitable. For some data sets KNN predict positive reaction and for others negative reaction with low accuracy about 30%,