Capstone Project

Machine Learning Engineer Nanodegree

Ajay Shewale

Sentiment Analysis of Text Data (Tweets)

Abstract

This project addresses the problem of sentiment analysis on Twitter. The goal of this project was to predict sentiment for the given Twitter post using Python. Sentiment analysis can predict many different emotions attached to the text, but in this report, only 3 major were considered: positive, negative and neutral. The training dataset was small (just over 5900 examples) and the data within it was highly skewed, which greatly impacted on the difficulty of building a good classifier. After creating a lot of custom features, utilizing bag-of-words representations and applying the Extreme Gradient Boosting algorithm, the classification accuracy at the level of 58% was achieved.

Analysing the public sentiment as firms trying to find out the response of their products in the market, predicting political elections and predicting socioeconomic phenomena like the stock exchange.

Introduction

Domain Background

I have started my startup right after the college, Blubyn. Blubyn is voice-based travel assistant which helps the user to book flights, hotels, and events. We are providing personalized results using Machine Learning. I have built my own Recommendation Algorithm for the same. Now while dealing with this issue, we came to the problem where user feedbacks are very crucial. It helps to recommend a user what they like and what they apt to chose.

The ability to understand the public sentiment in social media is increasingly considered as an important tool for market understanding and customer segmentation.

So, from this project, I receive an opportunity to work in sentiment analysis field. Also, it will definitely be beneficial for my startup. Because while dealing with the reviews of customers, we want to interpret what user tends to portray so that we can give him best recommended results.

Apart from this, Sentiment analysis has been an interesting field of study. This is still an evolving subject, it has functions that are too complicated to understand by the machines such as sarcasm, negative emotions, hyperbole etc.

Because I am part of the industry, I know the potential in sentiment analysis. It adds a lot of value to the industry. Sentiment analysis bases its results on factors that are so inherently humane, it is bound to become one the major drivers of many business decisions in future.

Problem Statement

The goal of this project is to predict the sentiment analysis of users Tweeter data. Sentiment analysis can predict many different emotions attached to the text, but in this project, only 3 major were considered: positive, negative and neutral.

Related Work

There are many papers written on sentiment analysis for the domain.

(Pang and Lee 2008) gives a survey of sentiment analysis [5] In a paper, Jansen has analyzed the commercial impact of social mediating technology, microblogging [1]. Overall, text classification using machine learning is a well-studied field (Manning and Schuetze 1999).[3]

There is an excellent work on the effects of various machine learning techniques such as Naive Bayes, Maximum Entropy, SVM in the movie reviews domain.(Pang and Lee 2002).[4] They were able to achieve an accuracy of 82.9% using SVM and a unigram model. Work (Read, 2005) has been done in using emoticons as labels for positive and sentiment. This is very connected to Twitter because many users have emoticons in their tweets.

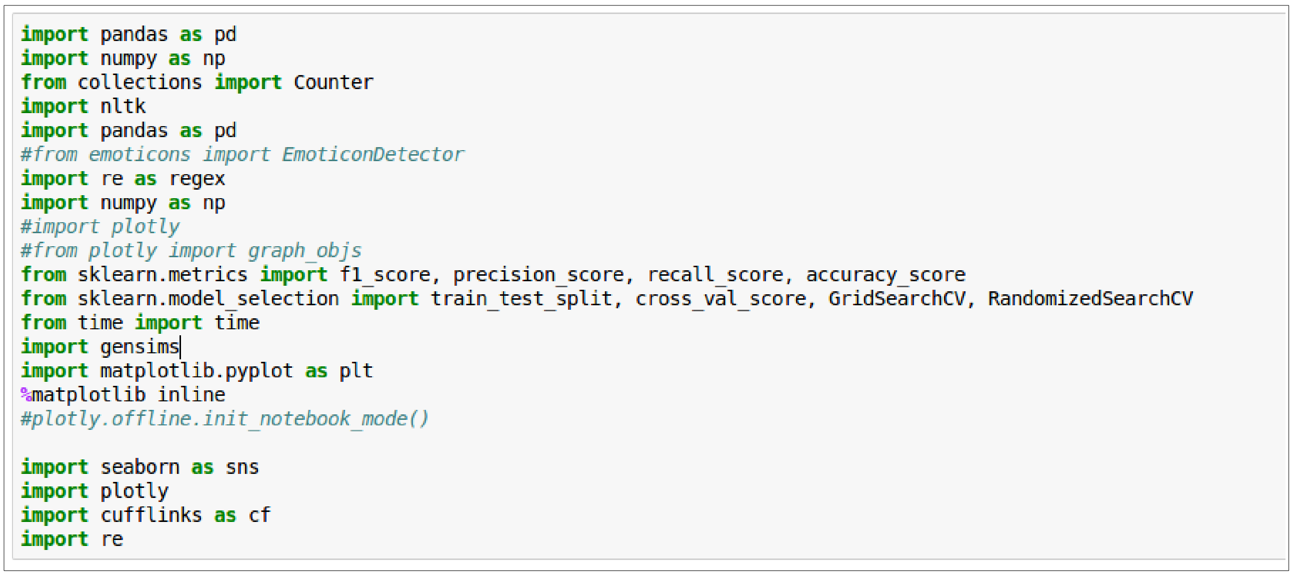
Researchers have also worked on detecting sentiment in text. (Turney 2002) presents a simple algorithm, called semantic orientation, for detecting sentiment.[8]

Rather than directly incorporating the microblogging features into sentiment classifier training, Speriosu et al. [14] constructed a graph that has some of the microblogging features such as hashtags and emoticons together with users, tweets, word unigrams and bigrams as its nodes which are connected based on the link existence among them (e.g.,users are connected to tweets they created; tweets are connected to word unigrams that they contain etc.). They then applied a label propagation method where sentiment labels

were propagated from a small set of nodes seeded with some initial label information throughout the graph. They claimed that their label propagation method outperforms MaxEnt trained from noisy labels and obtained an accuracy of 84.7% on the subset of the Twitter sentiment test set from [4].

Used Python Libraries

Data was pre-processed using *pandas*, *gensim* and *numpy* libraries and the learning/validating process was built with *scikit-learn*. Plots were created using *Seaborn*.



```
import pandas as pd
import numpy as np
from collections import Counter
import nltk
import pandas as pd
#from emoticons import EmoticonDetector
import re as regex
import numpy as np
#import plotly
#from plotly import graph_objs
from sklearn.metrics import f1_score, precision_score, recall_score, accuracy_score
from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV, RandomizedSearchCV
from time import time
import gensims
import matplotlib.pyplot as plt
%matplotlib inline
#plotly.offline.init_notebook_mode()

import seaborn as sns
import plotly
import cufflinks as cf
import re
```

Datasets And Input

The input data consisted of two CSV files:

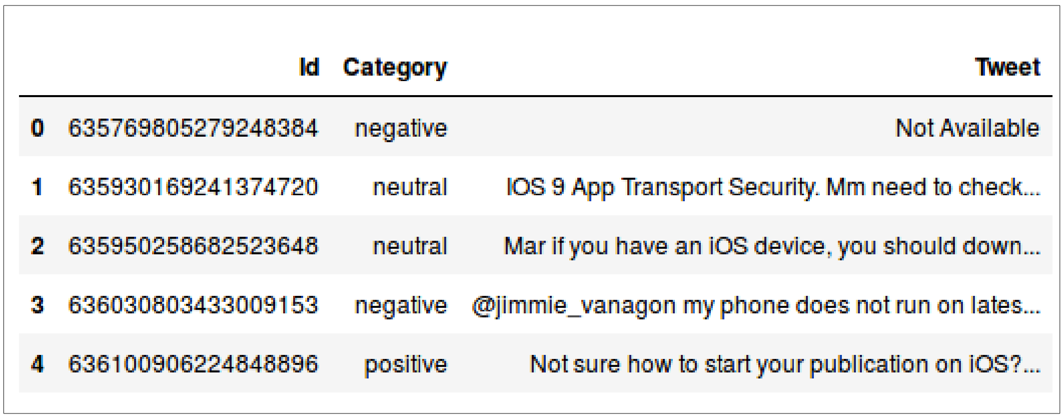
1. train.csv (5971 tweets)
2. test.csv (4000 tweets)

One for training and one for testing. The format of the data was the following (test data didn't contain Category column):

The dataset contains the following attributes:

Train data:

```
RangeIndex: 5970 entries, 0 to 5969
Data columns (total 3 columns):
Id          5970 non-null object
Category    5970 non-null object
Tweet       5970 non-null object
dtypes: object(3)
memory usage: 140.0+ KB
```



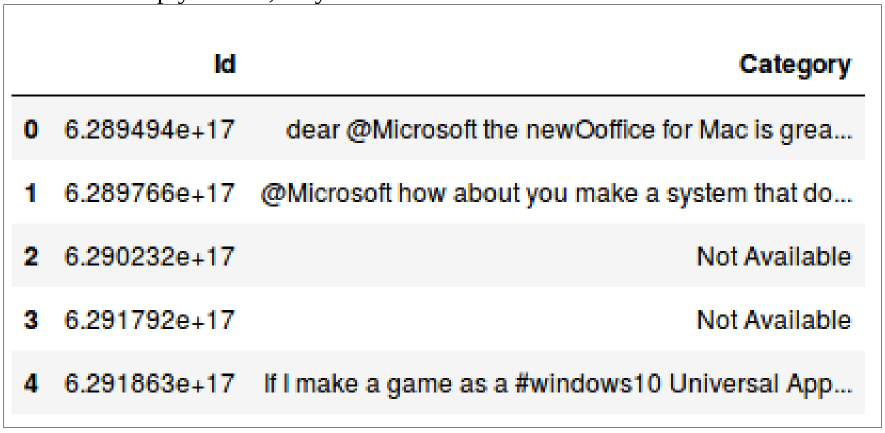
	Id	Category	Tweet
0	635769805279248384	negative	Not Available
1	635930169241374720	neutral	IOS 9 App Transport Security. Mm need to check...
2	635950258682523648	neutral	Mar if you have an iOS device, you should down...
3	636030803433009153	negative	@jimmie_vanagon my phone does not run on lates...
4	636100906224848896	positive	Not sure how to start your publication on iOS?...

Here, the' ***Category***' is the target class, given the '***Tweet***' column, '***Category***' defines whether the given user tweet is *positive, negative* or *neutral*.

Text Data:

RangeIndex: 9968 entries, 0 to 9967
Data columns (total 2 columns):
Id   4000 non-null float64
Category   4000 non-null object
dtypes: float64(1), object(1)
memory usage: 155.8+ KB

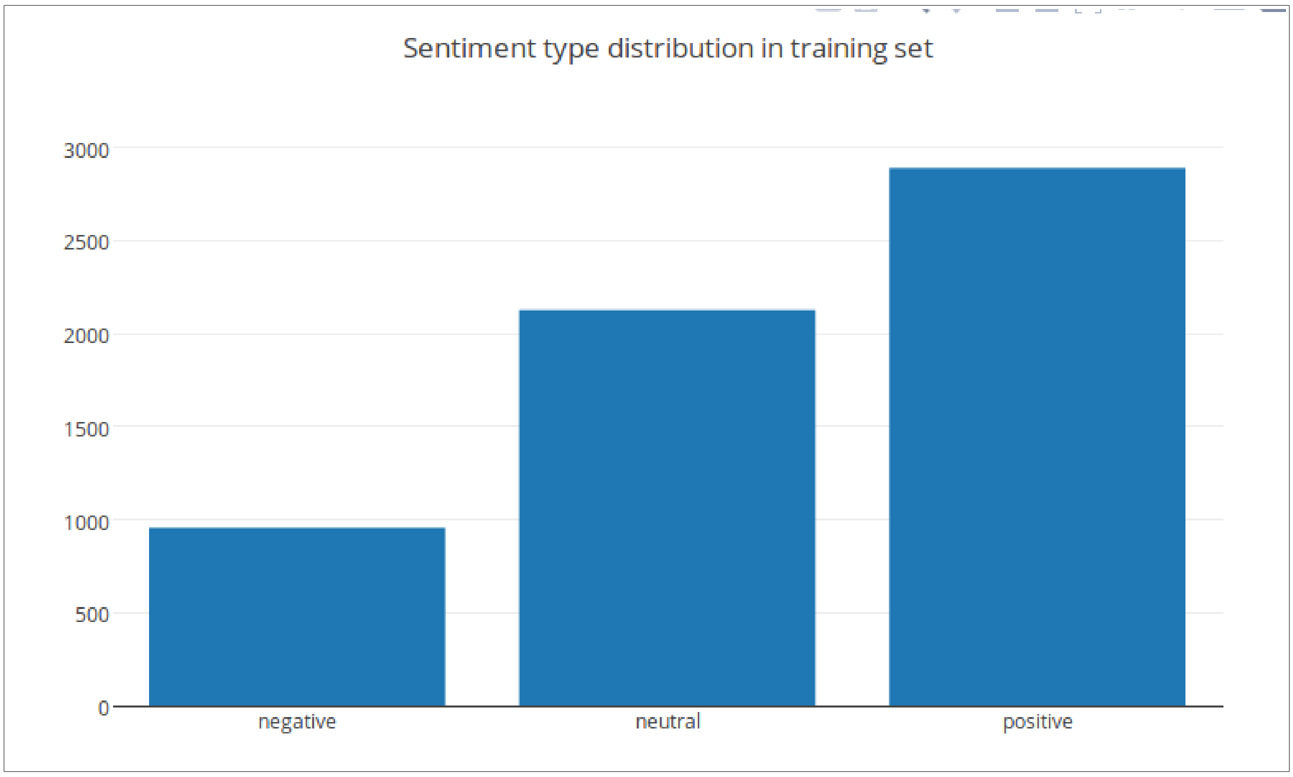
As the test.csv file was full of empty entries, they will be removed.



	Id	Category
0	6.289494e+17	dear @Microsoft the newOoffice for Mac is grea...
1	6.289766e+17	@Microsoft how about you make a system that do...
2	6.290232e+17	Not Available
3	6.291792e+17	Not Available
4	6.291863e+17	If I make a game as a #windows10 Universal App...

Data Distibution

The below figure shows how the target class is dirtibuted.



Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

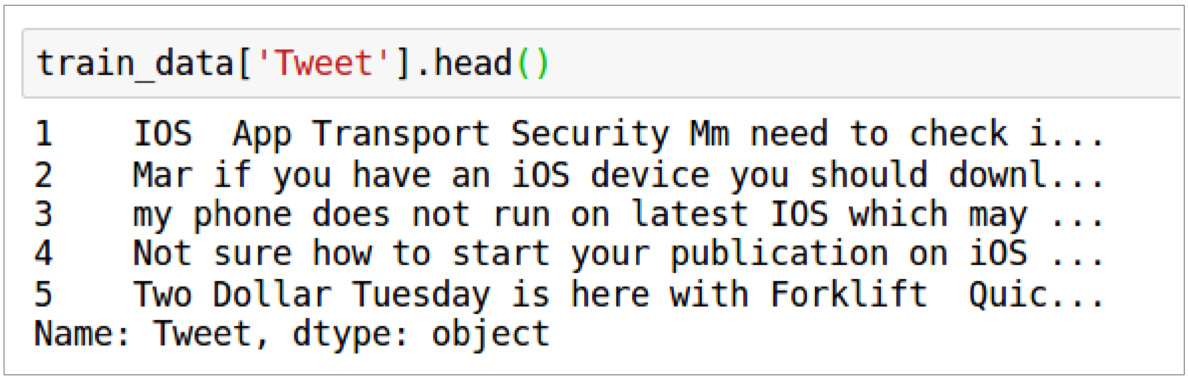
The aim of the following preprocessing is to create a **Bag-of-Words** representation of the data. The steps will execute as follows:

1. *Cleaning*
   - Remove URLs
   - Remove usernames (mentions)
   - Remove tweets with *Not Available* text
   - Remove special characters
   - Remove numbers
2. *Text processing*
   - Tokenize
   - Stemming
3. Build word list for Bag-of-Words

Data Cleanng:

Data cleaning is one of the crucial part to prepare the data for Bag-of-word representation.

After Cleaning:



```
train_data['Tweet'].head()
```

```
1     IOS  App Transport Security Mm need to check i...
2     Mar if you have an iOS device you should downl...
3     my phone does not run on latest IOS which may ...
4     Not sure how to start your publication on iOS ...
5     Two Dollar Tuesday is here with Forklift  Quic...
Name: Tweet, dtype: object
```

Tokenization & stemming

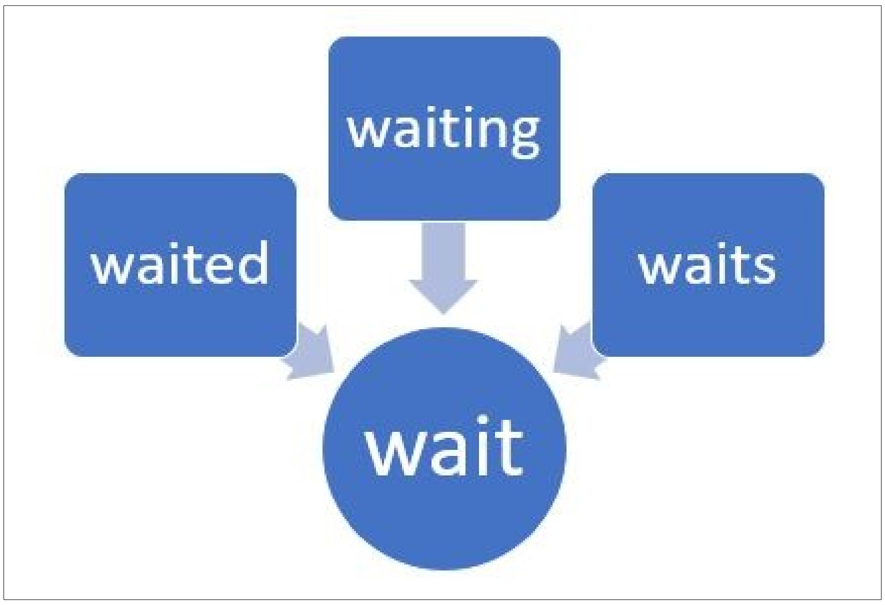
Tokenization consists of splitting the text into words, and words in a context that is with spaces, punctuation signs, cases, accents, diacritics, and so on — into standardized words. This step is critical for the whole flow accuracy

Stemming consists of preparing words expressions to find their stems. The process relies on a suffix dictionary making it possible to extract stems after analyzing the morphology of the word. According to the inflected forms identified and the defined language, it computes the most relevant stems from the grammar and syntax rules of the language.

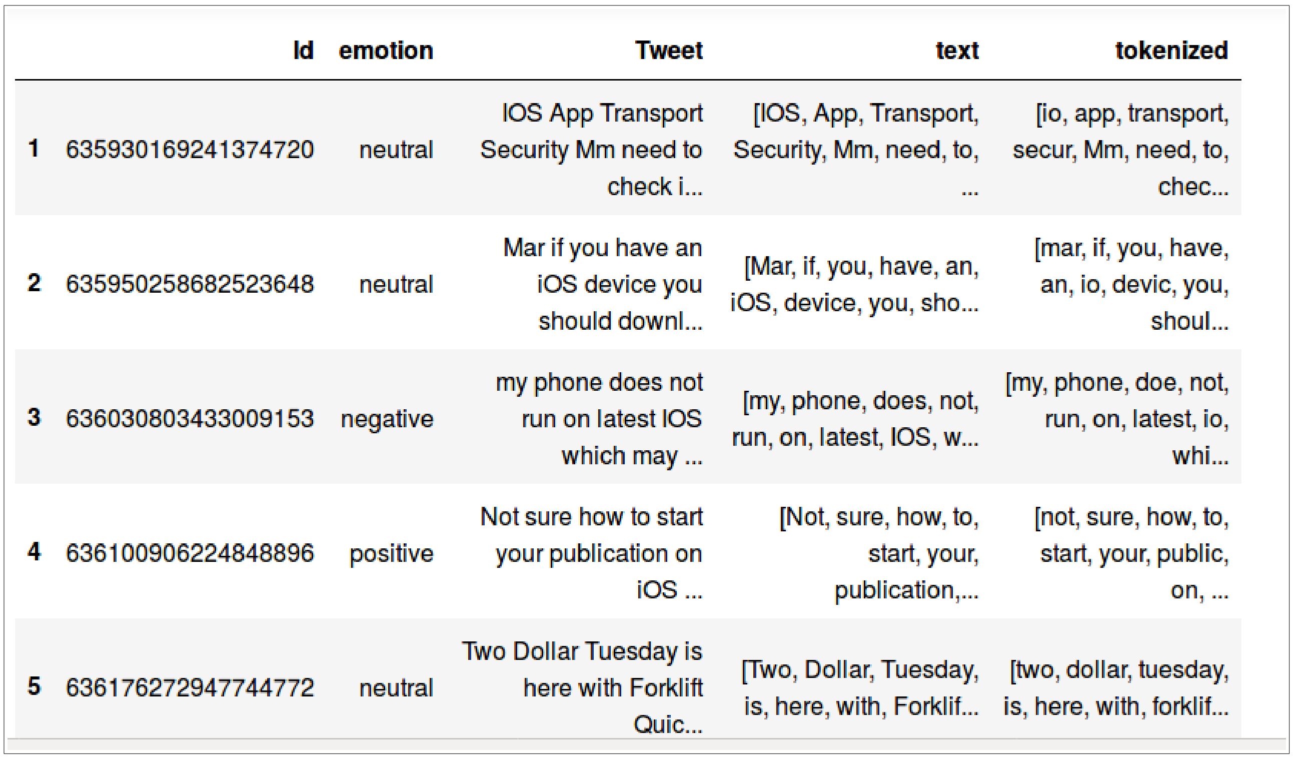
Stemming offers two main benefits:

- As it focuses on words stems, the process is quite tolerant of spelling mistakes.
- It only needs words to stem without requiring their context of use.

Below feagure[10] is the stemming example:



For the text processing, nltk library is used. First, the tweets are tokenized using nlkt.word_tokenize and then, stemming is done using **PorterStemmer** as the tweets are 100% in english.



	Id	emotion	Tweet	text	tokenized
1	635930169241374720	neutral	IOS App Transport Security Mm need to check i...	[IOS, App, Transport, Security, Mm, need, to, ...	[io, app, transport, secur, Mm, need, to, chec...
2	635950258682523648	neutral	Mar if you have an iOS device you should downl...	[Mar, if, you, have, an, iOS, device, you, sho...	[mar, if, you, have, an, io, devic, you, shoul...
3	636030803433009153	negative	my phone does not run on latest IOS which may ...	[my, phone, does, not, run, on, latest, IOS, w...	[my, phone, doe, not, run, on, latest, io, whi...
4	636100906224848896	positive	Not sure how to start your publication on iOS ...	[Not, sure, how, to, start, your, publication,...	[not, sure, how, to, start, your, public, on, ...
5	636176272947744772	neutral	Two Dollar Tuesday is here with Forklift Quic...	[Two, Dollar, Tuesday, is, here, with, Forklif...	[two, dollar, tuesday, is, here, with, forklif...

Wordlist

A bag-of-words model, or BoW for short, is a way of extracting features from the text for use in modeling, such as with machine learning algorithms.

The approach is very simple and flexible and can be used in a myriad of ways for extracting features from documents.

The wordlist (dictionary) is build by simple count of occurences of every unique word across all of the training dataset.

Additional features

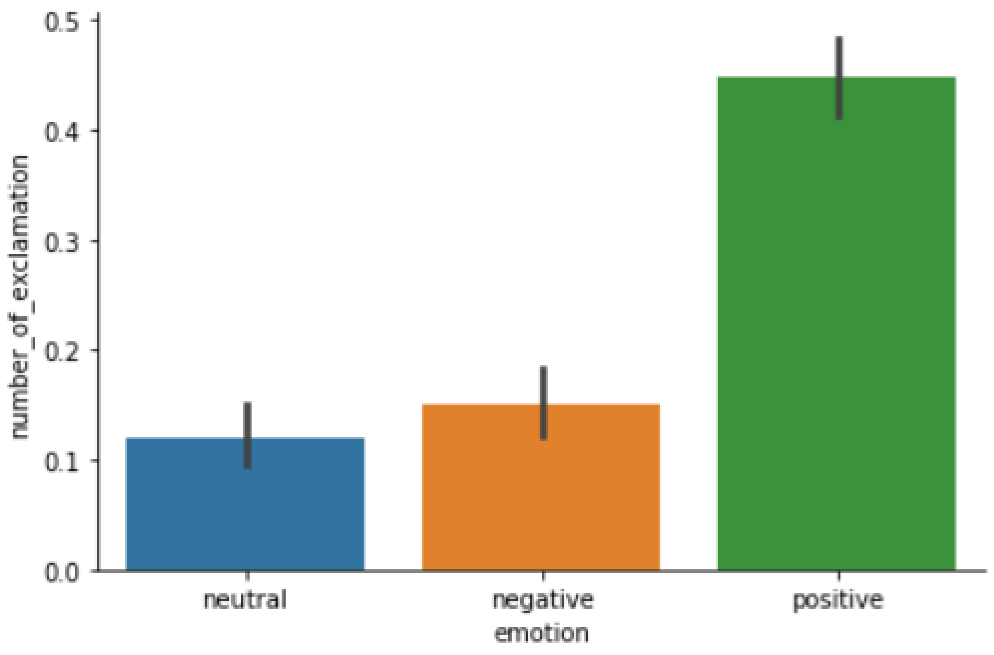
In order to **not** push any other aglorithm to the limit on the current data model, let's try to add some features that might help to classify tweets.

Feature name	Explanation
Number of uppercase	Uppercase words help to identify the negative or positive emotions of a user
Number of !	Exclamation marks are likely to increase the strength of opinion
Number of ?	Can be used to identify neutral tweets
Number of positive emoticons	Use of positive emoticon may imply the positive sentiment
Number of negative emoticons	Use of negative emoticon may imply the negative sentiment
Number of quotations	Same as above
Number of mentions	People mention other people mostly to share something good
Number of hashtags	For experiment purpose

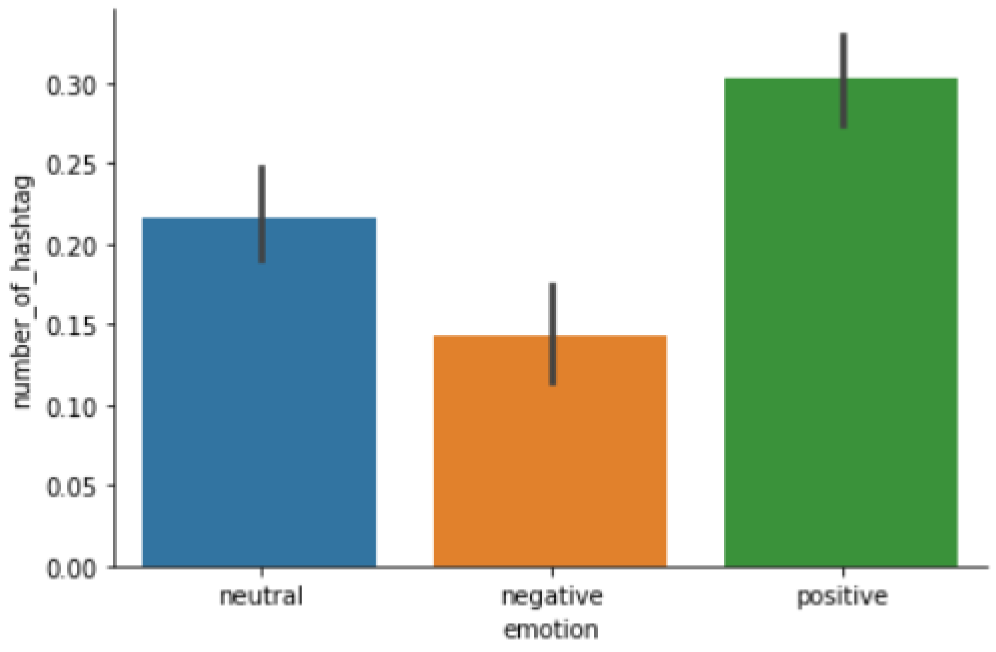
Why to use additional features (Graph representaion)

Let's see how (some) of the extra features distribute the data set. Some of them, i.e number exclamation marks, number of pos/neg emoticons do this surprisingly well. Notwithstanding the good severance, those features seldom transpire only on a small subset of the training dataset.

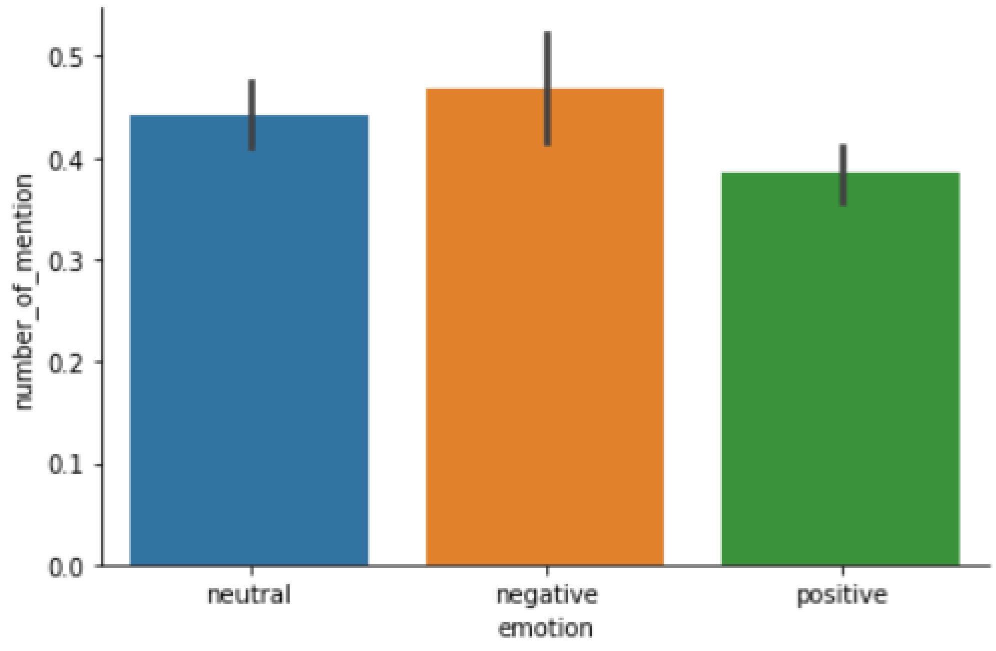
Number of Exclamation:



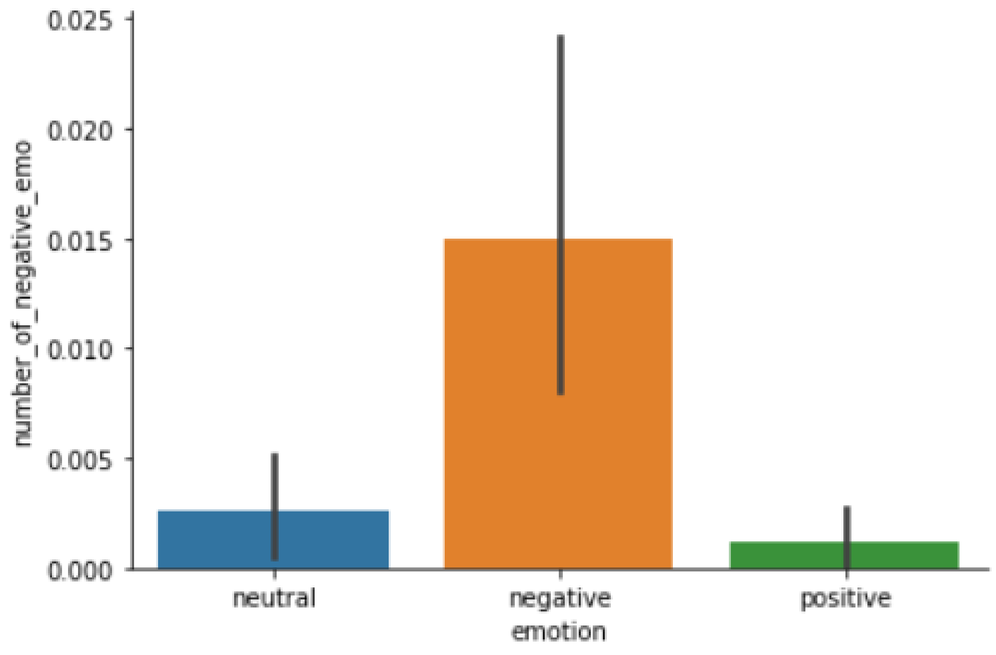
Number of Hashtag



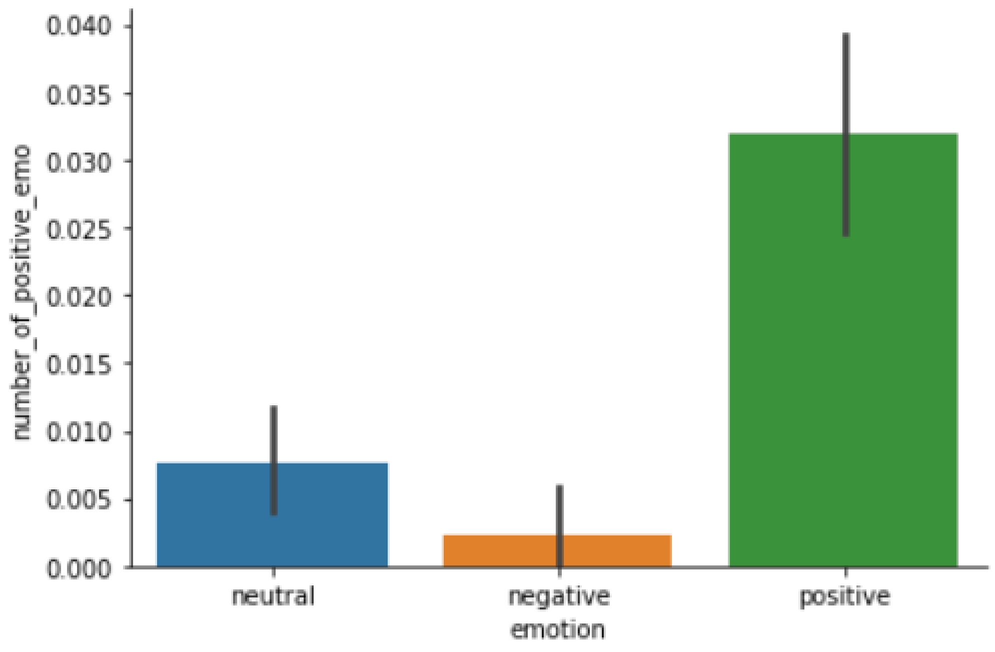
Number of Mentions



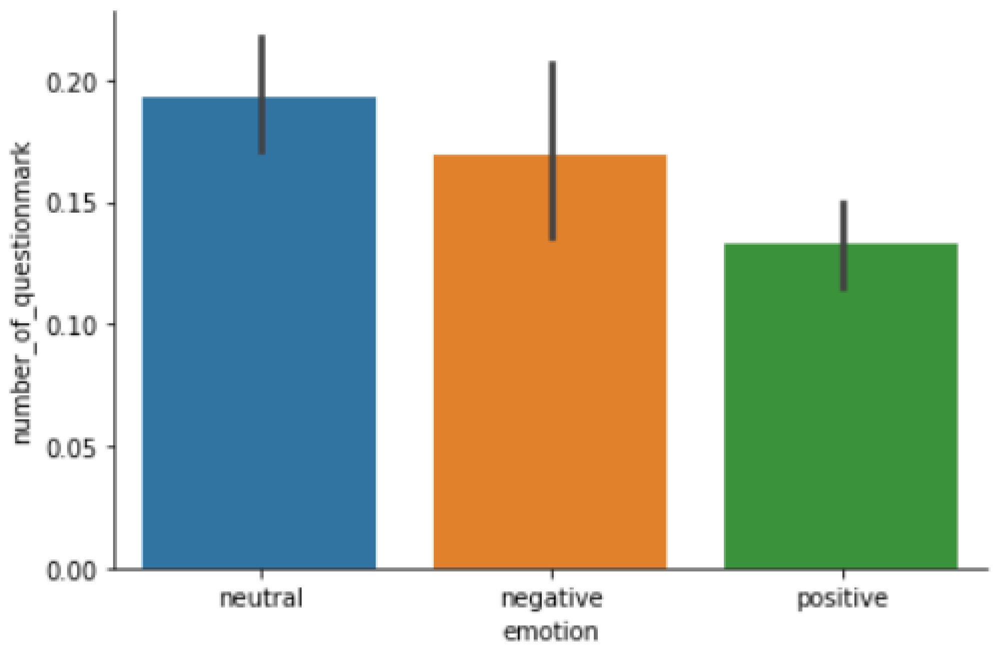
Number of Negative emoticons



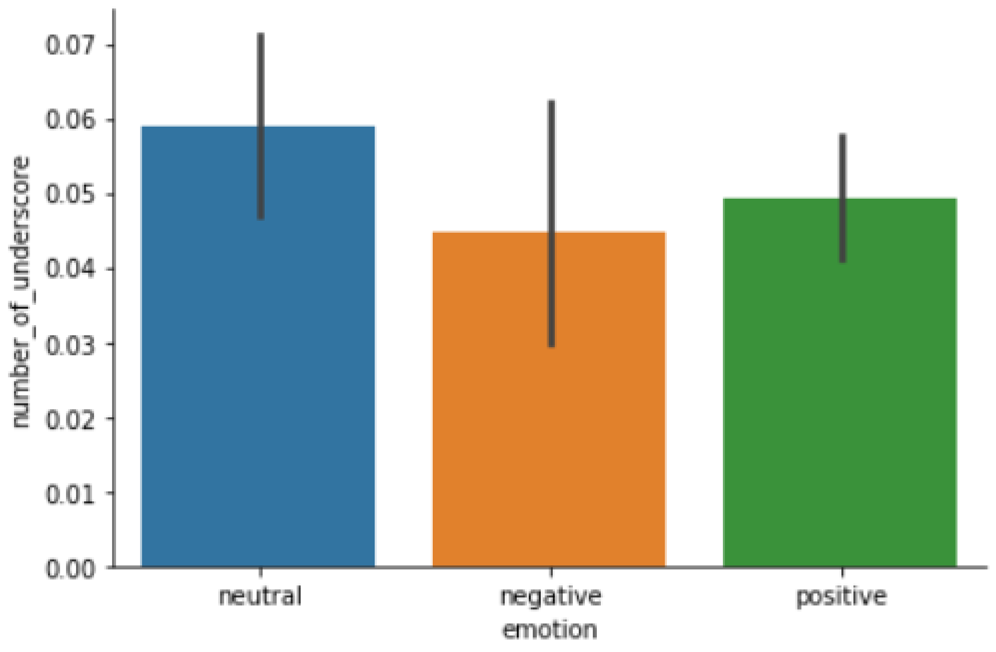
Number of Positive emoticons



Number of Questionmark



Number of Underscore:

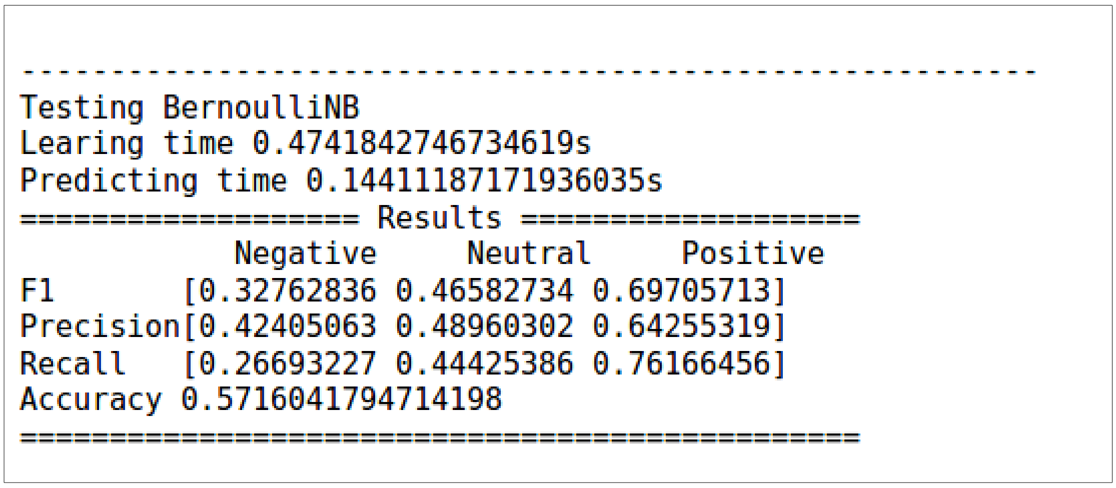


Results

Experiment 1: BOW + Naive Bayes

It is nice to see what kind of results we might get from such simple model. The bag-of-words representation is binary, so Naive Bayes Classifier seems like a nice algorithm to start the experiments.

The experiment will be based on 7:3 train:test stratified split.

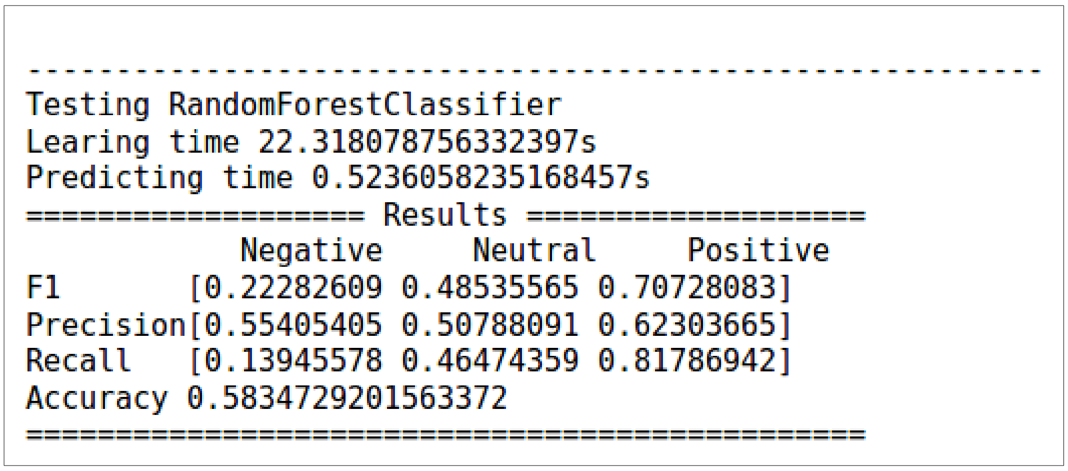


```
---------------------------------------------------------
Testing BernoulliNB
Learing time 0.4741842746734619s
Predicting time 0.14411187171936035s
=================== Results ===================
            Negative     Neutral     Positive
F1       [0.32762836 0.46582734 0.69705713]
Precision[0.42405063 0.48960302 0.64255319]
Recall   [0.26693227 0.44425386 0.76166456]
Accuracy 0.5716041794714198
===============================================
```

Experiment 2: Added fetatue + Random Forest

As a second attempt on the classification the **Random Forest** will be used.

It looks better, however it's still not much above accuracy of the random classifier and better than Naive Bayes classifier.

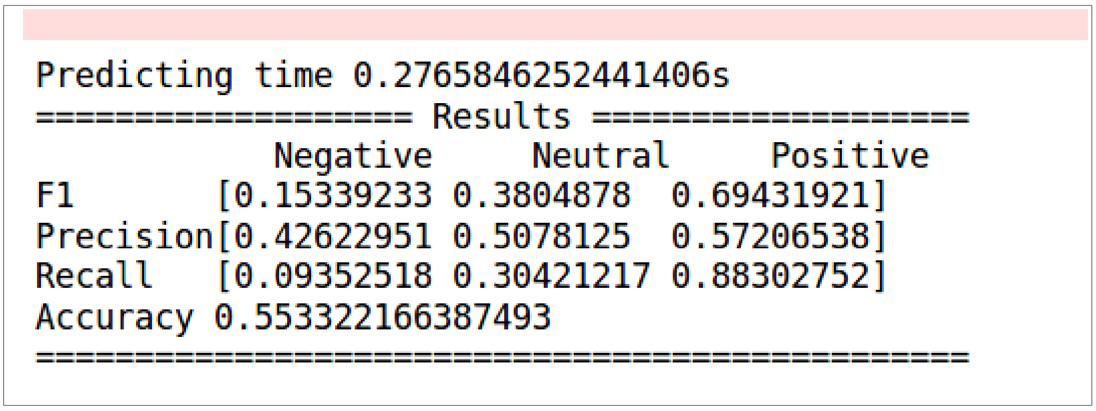


```
---------------------------------------------------------
Testing RandomForestClassifier
Learing time 22.318078756332397s
Predicting time 0.5236058235168457s
=================== Results ===================
            Negative     Neutral     Positive
F1       [0.22282609 0.48535565 0.70728083]
Precision[0.55405405 0.50788091 0.62303665]
Recall   [0.13945578 0.46474359 0.81786942]
Accuracy 0.5834729201563372
===============================================
```

We can observe a low recall level of the RandomForest classifier for the negative class, which may be caused by the data skewness.

Experiment 3: Added Feature + XGBoost

XGBoost is relatively new machine learning algorithm based on decision trees and boosting. It is highly scalable and provides results, which are often higher than those obtained using popular algorithms such as Random Forest or SVM

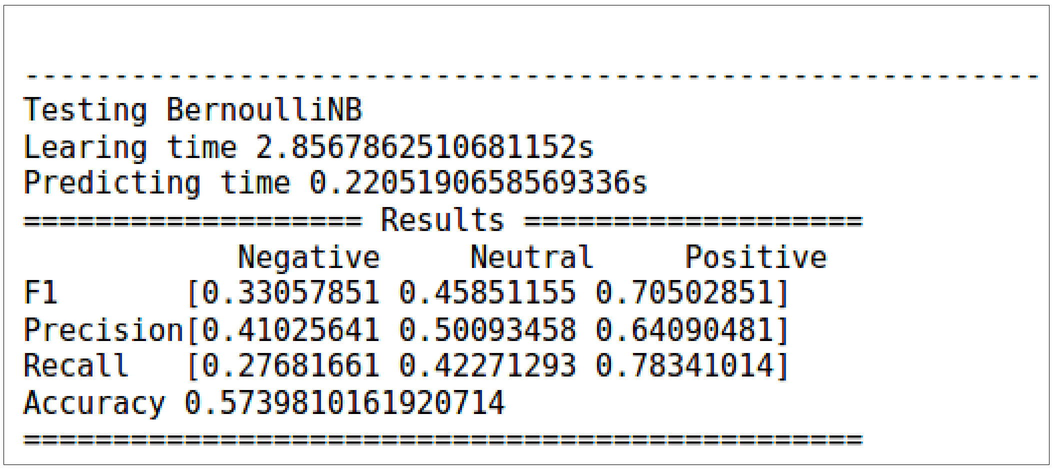


```
Predicting time 0.2765846252441406s
=================== Results ===================
             Negative     Neutral     Positive
F1        [0.15339233 0.3804878  0.69431921]
Precision[0.42622951 0.5078125  0.57206538]
Recall    [0.09352518 0.30421217 0.88302752]
Accuracy 0.553322166387493
===============================================
```

We can observe that the accuracy for XGBoost is less, which implies the accuracy depends on the data you have not the classifier you use. Each data is unique in itself and will be shine best with the proper classifier.

Experiment 4: Added Feature + Naive Bayes

After the extented features, the accuracy of the Naive Bayes increased.



```
----------------------------------------------------------
Testing BernoulliNB
Learing time 2.8567862510681152s
Predicting time 0.2205190658569336s
=================== Results ===================
            Negative     Neutral     Positive
F1        [0.33057851 0.45851155 0.70502851]
Precision[0.41025641 0.50093458 0.64090481]
Recall    [0.27681661 0.42271293 0.78341014]
Accuracy 0.5739810161920714
===============================================
```

From the above experiments we can conclude that Random Forest gives better result than other models hence we the Test Data will be classified on Random Forest.

Test data classification

After finding the best classifier, load the test data and predict sentiment for them. The data will be exported to CSV file in format containing two columns: Id, Category. There are 4000 test samples with unknown distribution of the sentiment labels.

Conclusion

The increase of microblogging sites like Twitter offers an unparalleled opening to form and employ approaches & technologies that search and mine for sentiments. The work presented in this paper specifies an approach for sentiment analysis on Twitter data. To unseal the sentiment, we extracted the relevant data from the tweets, added the features.

The overall tweet sentiment was then calculated using a model that presented in this report. This work is exploratory in nature and the prototype evaluated is a preliminary prototype.

The models showed that prediction of text sentiment is a non-trivial task for machine learning. A lot of preprocessing is needed just to be able to run an algorithm. The main problem for sentiment analysis is to craft the machine representation of the text. Simple bag-of-words was definitely not enough to obtain satisfying results, thus a lot of additional features were created basing on common sense (number of emoticons, exclamation marks, number of question mark etc). I think that a slight improvement in classification accuracy for the given training dataset could be developed, but since it included highly skewed data (small number of negative cases), the difference will be probably in the order of a few percents. The thing that could possibly enhance classification outcomes will be to add a lot of additional examples (increase training dataset), because given 5971 examples clearly do not include all sequence of words used, further - a lot of emotion-expressing information certainly is missing.

Future Work

In the classification, I have covered most of the features but emotion expressing information is missing.

Furthermore, it can be useful to analyze the viewpoints of user reviews, whether it is a good or bad review. The above method can be used in hotel, flights reviews to recommend other users about the services. I will try to consolidate the above system into my recommendation system to provide users with trustworthy suggestions regarding the flights and hotels.

Reference

[1] B. Jansen, M. Zhang, K. Sobel, A. Chowdury. The Commerical Impact of Social Mediating
[2] Technologies: Micro-blogging as Online Word-of-Mouth Branding, 2009.

[3] C. Manning and H. Schuetze. Foundations of Statistical Natural Language Processing. 1999.

[4] B. Pang, L. Lee, S. Vaithyanathan.Thumbs up?Sentiment Classification using Machine Learning Techniques, 2002.

[5] B. Pang and L. Lee. "Opinion Mining and Sentiment Analysis" in Foundations and Trends in Information Retrieval, 2008.

[6] B. Pang and L. Lee. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts" in Proceedings of ACL, 2004.

[7] J. Read. Using Emotions to Reduce Dependency in Machine Learning Techniques for Sentiment Classification, 2005.

[8] P. Turney. "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews" in Proceedings of the 40th Annual Meeting of the Association for ***Computatoinal Linguistics (ACL), 2002.***
[9] https://doc.antidot.net/reader/edTsdsfeT~k4cB6nOFOXOA/LPV1a2xWskQ4jllIUHrK4Q

[10] https://www.c-sharpcorner.com/blogs/stemming-in-natural-language-processing