Coursera Capstone Project 2015

Trung-Hieu Nguyen

November 12, 2015

Introduction

This is a capstone project report for Coursera Data Science Specialization conducted by John Hopskin University. The dataset given was Yelp Challenge Data set

During first half of the course, we have already explored dataset. Below were our stated questions and tasks of interested.

Primary question: Can we predict from a textual review whether general opinion on business is positive or negative?

The code work can be viewed and downloaded at my Github repository

Methods and codes:

Note:
The data is relatively large and takes a significant amount of time to process. Hence many codes are broken down into different R files to run seperately. Process data are then saved in appropriate format to be used throughout this report.
Hence please refer to a specific files for further details.

Raw data loading:

Please refer to *DataLoading.R* for codes.
Only business and review data was used for our analysis

```
#Raw data load Task 0
business <- readRDS("business.rds")
reviews <- readRDS("reviews.rds")
```

Data wrangling:

Please refer to files *DataWrangling.R* for codes. Details of method is explained below.

Restrains applied on our data and its purposes:
1/ Use only data from 6 cities in US to remove non-English reviews.
2/ Use only data from food services to increase homogenity in our model.

Then we combine text reviews from both business data and reviews data into a single data frame for ease of manipulation.

Firsly, we notice that there are emoticons used in reviews. These can be a good indicator of review sentiment. However we need to convert these emoticons to textual words. To do so we use package *qdap*.

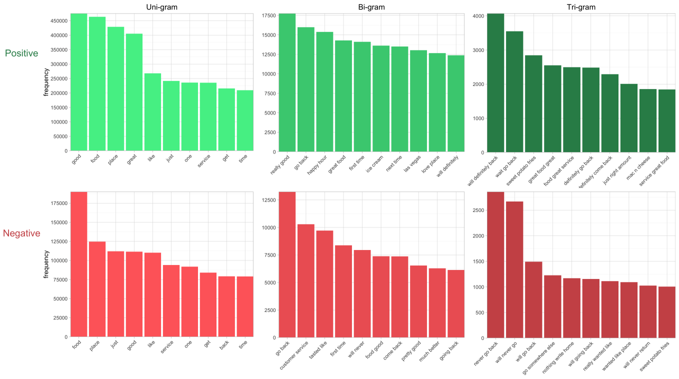
After that, we go through a usual practise of textual cleaning tasks. That includes removing punctuations, numbers, line breaks etc as well as converting everything to lower case letters. To do so we us package *tm*.

Furthermore Stopwords were also removed. This makes model constructing features easier

```
#This is the cleaned review text after all procedures described above
Yelp_reviews <- read.csv("Yelp_reviews_cleaned.csv")
```

Sentiment analysis with n-gram:

N-grams were analyzed using the *quanteda* package in R.
We use uni, bi, tri and quad gram to analyze positive reviews (4 and 5 stars) vs negative reviews (1 and 2 stars). A valid assumption that 3 stars review is neutral is made. Please refer to *NgramChart.R* for codes and details.



The uni-gram exhibit a similar frequencies of similar set of single words for both positive and negative, thus can be hardly distinguish.
The bi-gram however display a relatively significant different between positive and negative reviews.
The tri-gram is the most differetiated intuitively. ("will definitely go back" vs "never go back")

Models:

The analysis was completed using Python with *graphlab* library.

Please refer to *SentimentAnalysis.ipynb* for code lines and details are explained as below.

Binary classifier model to determine if review is positive or negative. Each review was labeled as 1 for positive (4 or 5 stars) and 0 for negative (1 or 2 stars).

The model was trained on data selected by randomly splitting the entire dataset into a 70% training and 30% testing dataset. (*line 14*)

Adding features such as n-grams (where n > 1) did not increase the accuracy on the training dataset and significantly increased computation time so they were not included in the final model.

The final model uses the 'bag of words' approach or 1-gram counts. This was created using graphlab's function 'text_analytics.count_words,' which counts words in each review.

Among random forest, naive Bayes and support vector algorithm and logistic regression, logistic regression gave the most accurate results in the training dataset.

Results

Accuracy and confusion matrix:

The trained model was applied to the test dataset and the accuracy was 0.942. The confusion matirx is shown below. (*line 18* in python file)

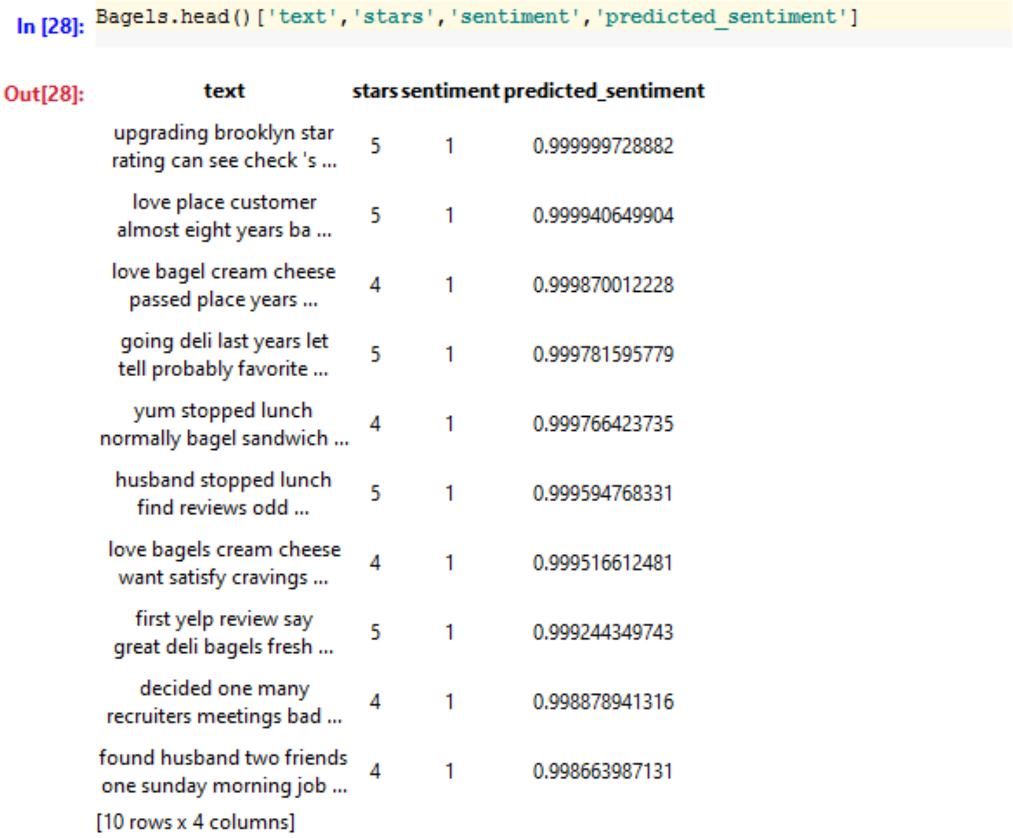
		Predicted	label
		0	1
Target	0	4871	9339
label	1	6009	201198

Validation by applied model to certain business:

For example, we apply our model to restaurant Bagel deli which has business id as "wx2EJUCNOCPrMC0DtKb98A" (*line 25 to line 32* in python file)

Then by comparing predicted sentiment with underlying sentiments using intutive judgemen, we can see that in fact the model is pretty accurate.

Below are the attached of head and tail of reviews data (sorted by predicted sentiment value from 0 to 1) (*line 28 and line 30*):

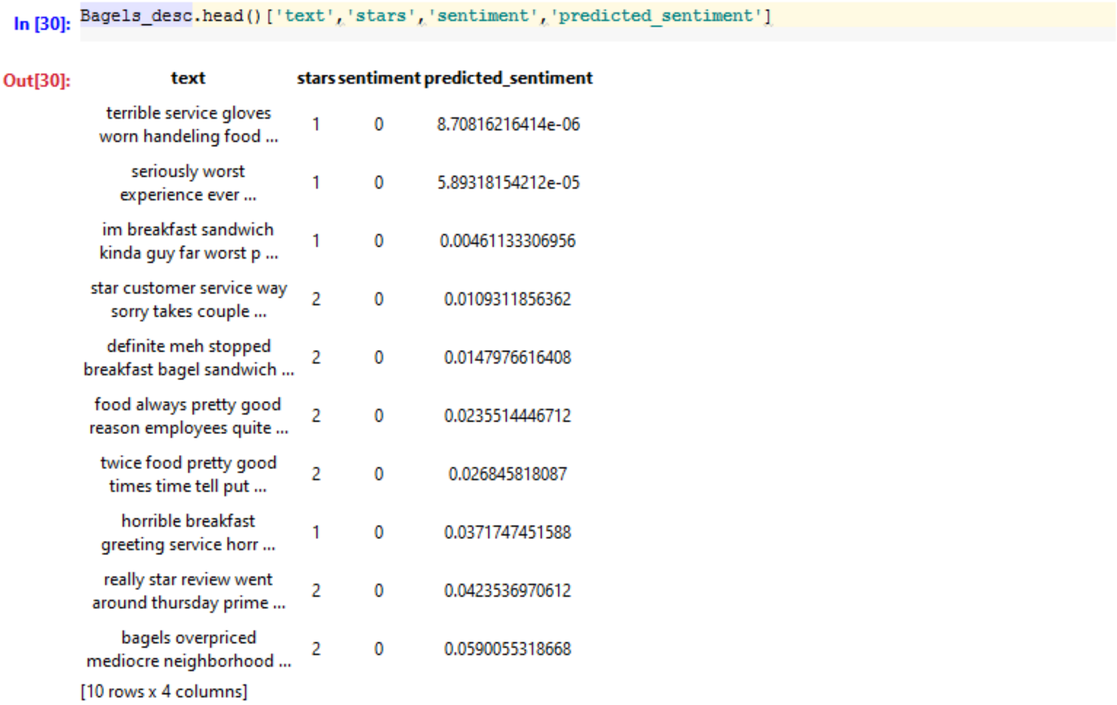


```
In [28]: Bagels.head()['text','stars','sentiment','predicted_sentiment']
```

Out[28]:

text	stars	sentiment	predicted_sentiment
upgrading brooklyn star rating can see check 's ...	5	1	0.999999728882
love place customer almost eight years ba ...	5	1	0.999940649904
love bagel cream cheese passed place years ...	4	1	0.999870012228
going deli last years let tell probably favorite ...	5	1	0.999781595779
yum stopped lunch normally bagel sandwich ...	4	1	0.999766423735
husband stopped lunch find reviews odd ...	5	1	0.999594768331
love bagels cream cheese want satisfy cravings ...	4	1	0.999516612481
first yelp review say great deli bagels fresh ...	5	1	0.999244349743
decided one many recruiters meetings bad ...	4	1	0.998878941316
found husband two friends one sunday morning job ...	4	1	0.998663987131

[10 rows x 4 columns]



In [30]:

```
Bagels_desc.head()['text','stars','sentiment','predicted_sentiment']
```

Out[30]:

text	stars	sentiment	predicted_sentiment
terrible service gloves worn handeling food ...	1	0	8.70816216414e-06
seriously worst experience ever ...	1	0	5.89318154212e-05
im breakfast sandwich kinda guy far worst p ...	1	0	0.00461133306956
star customer service way sorry takes couple ...	2	0	0.0109311856362
definite meh stopped breakfast bagel sandwich ...	2	0	0.0147976616408
food always pretty good reason employees quite ...	2	0	0.0235514446712
twice food pretty good times time tell put ...	2	0	0.026845818087
horrible breakfast greeting service horr ...	1	0	0.0371747451588
really star review went around thursday prime ...	2	0	0.0423536970612
bagels overpriced mediocre neighborhood ...	2	0	0.0590055318668

[10 rows x 4 columns]

Discussion:

Since there is restriction in both time and resources, we have limited the analysis within food servie business in US only. The model hence cannot be used to generalise all other sectors in other places.

Binary classification on sentiment is just a simple model. A further mutifactor model can also be developed to specifically indicates the star rating.

However the accuracy obtained with our bianary classifier model is quite high at 94%.