Data Science Capstone Project
Springboard Bootcamp, San Francisco
Reza Taeb
Spring 2018
Title : "Improving Restaurant Reputation Using Yelp User Reviews"
Part 4 : Machine Learning

How to improve restaurant reputation even just one star using Yelp user reviews !
" The path from being normal to good, and good to perfect !"

In this part I am trying to figure out whether we can recognize the "ratings" of reviews by just checking the "text" of reviews or not. According to the distribution of review ratings, 3, 4, and 5 star are the most common ratings that restaurants received. In each part the difference between 3 and 4 ; 4 and 5 ; and 3 and 5 stars review's texts will be investigated. The ultimate goal of this part is checking the accuracy of a model that can predict review's rating just using its text. This problem can be identified as classification problem.

1. 3 and 4 star ratings reviews classification

The task is to predict if a review is enough to predict whether a reviewers give either 3 or 4 star as rating on Yelp or not. So let's just grab reviews that are either 3 or 4 stars from the yelp dataframe.

The "text" column is independent variable (X), and the "stars" column is dependent (y) varaible.

In order to prepare the "text" column for classification algorithm we need to convert the sentences into vectors. One of the simple way to do this is **bag-of-words**[1] approach. Some text processing techniques (stopword removal, punctuations removal, lemmatization, etc.) are essential to get the better results in text classification problems.

The "CountVectorizer" in Scikit-learn is used to convert the text documents into a matrix of token counts.

After transforming the X ("text" column) into a sparse matrix, it's time to train a classification model. The classification model which is used in this project is **Multinomial Naive Bayes**[2]**.**

I used the 30-70% ratio for testing-training data sets. After fitting the training data set into the model, two tools ("confusion_matrix" , "classification_report") from Scikit-learn were used to evaluate the model. Here are the results for 3 and 4 star ratings reviews:

[1] In "bag-of-words" approach, each unique word in a text is represented by one number.
[2] Multinomial Naive Bayes is a specialised version of Naive Bayes designed more for text documents.

Classification Report (3 and 4 stars reviews)

Confusion Matrix
[[560 74]
[119 522]]

	precision	recall	f1-score	support
3	0.70	0.75	0.72	634
4	0.73	0.68	0.70	641
avg / total	0.71	0.71	0.71	1275

According to the above chart, the model has 71% precision and recall rates. It means that on average, the model is able to correctly guess “rating” of a review just based on its text in 71 percent. Also the model can find the “rating” of 71 percent of total population of 3 and 4 stars. Relatively this ratios are high enough to trust the model, however some modifications may increase this ratios and can be the subject of the other research.

2. 4 and 5 star rating reviews classification

Classification Report (4 and 5 stars reviews)

Confusion Matrix
[[804 387]
[317 885]]

	precision	recall	f1-score	support
4	0.72	0.68	0.70	1191
5	0.70	0.74	0.72	1202
avg / total	0.71	0.71	0.71	2393

According to the above table, the model works very similar to the previous part in distinguishing between 4 and 5 star reviews.

3. 3 and 5 star rating reviews classification

Classification Report (3 and 5 stars reviews)

Confusion Matrix
[[563 71]
[110 531]]

	precision	recall	f1-score	support
3	0.84	0.89	0.86	634
5	0.88	0.83	0.85	641
avg / total	0.86	0.86	0.86	1275

As you can see, 3 and 5 stars ratings are easier to be separated compared to two other scenarios that mentioned before. It is totally aligned with common belief that the more difference between ratings the more difference in language. According to the above table, both “precision” and “recall” have ratios around 86% which can be good enough for this classification problem.