618 Final Project Report

Motivation

I chose yelp review dataset because I am a foodie and I find yelp reviews to be very useful. I would like to know whether there is a pattern in users' reviews.

The four specific questions that I decided to explore for this dataset are:

(1) What are the most popular words users have written in positive reviews?

(2) What are the ten most frequent part of speech tags in the review text?

(3) Some reviews are tagged as useful, funny or cool. Is there a correlation between stars, useful, funny and cool? (i.e. if a review is tagged as useful, funny or cool, is it more likely to have a very high star or low star?) If a review is tagged as useful, is it more like to be tagged as funny or cool?

(4) Can we predict whether a review is positive or negative by analyzing yelp review "text" column?

Data Source

I downloaded the yelp review csv file from Kaggle (https://www.kaggle.com/yelp-dataset/yelp-dataset/data).

Here are the important variables and their types:

(1) review_id: text

(2) user_id: text

(3) business_id: text

(4) stars: integer

(5) date: date

(6) text: text

(7) useful: integer

(8) funny: integer

(9) cool: integer

I used the first 400,000 records, instead of all of them, for my analysis, to make it run faster. The time periods the data covered are from 2004-07-22 to 2017-12-11.

Methods

Question 1

As for manipulating the data to prepare it for analysis, firstly, I chose the first 300,000 review texts instead of all review texts to make the analysis fun faster. Secondly, I filter the data to get only reviews whose star>3 because I want to only focus on positive reviews.

In terms of how to handle noisy data, I imported stopwords from wordcloud to exclude stopwords. Therefore, only words that make sense will appear in the wordcloud.

When it comes to the workflow of my source code, firstly, I imported packages needed for this project and then read the csv file into review dataframe. Then I chose the first 300,000 review texts and filter the data to get only reviews whose star>3. Then I used wordcloud and a minion image to generate a colorful minion shape which shows the most popular words in the minion.

As for the problems I encounter, word cloud was not taught in the class. So I watched several tutorials online. There is an example online using the picture of *Alice in Wonderland* (https://www.kaggle.com/poonaml/bidirectional-lstm-spacy-on-

yelp-reviews). I learned from it and created with my own example of the minion.

Question 2

As for manipulating the data to prepare it for analysis, I read strings in the ‘text’ column into “output.txt” to make it work for sc.textfile() function.

In terms of how to handle noisy data, the text was normalized to remove all the ',' and '.' using the regex expression.

When it comes to the workflow of my source code, firstly, I normalize the text and put tags in the normalize_and_extract_Pos function. The normalized text was tokenized and mapped later was reduced based on the POS tag as the key. A new map reduce task was created to compute the number of part of speech in the entire text.

As for the problems I encounter, in the very beginning, the data source I have for this question is a string list. But I do not know how to use it with the flatMap function. I tried to turn every item in the list into a readable format for flatMap, but the cell did not respond at all. In the end, I decided to write the string list into a text file and than read the text file back to Jupyter.

Question 3

As for manipulating the data to prepare it for analysis, I chose ‘useful’, ‘funny’ and ‘cool’ columns from review dataframe.

In terms of how to handle missing data, I used .isnull() function to check whether there is missing data in each column. The result turns out that there is no missing data.

When it comes to the workflow of my source code, firstly I used .isnull() function to check whether there is missing data in each column. Then I used .corr() function to calculate the correlation coefficient among stars, cool, funny and useful.

As for the problems I encounter, in the very beginning, I wanted to explore whether there is correlation between stars and cool, stars and funny and stars and useful. However, the correlation coefficients are very close to 0. Therefore, my assumption seems to be unreasonable.

Question 4

As for manipulating the data to prepare it for analysis, I brainstormed possible features and loop through the data to prepare for the features.

In terms of how to handle noisy data, the yelp review data from Kaggle is quite complete. The assumption is that there are no missing or incomplete data. Since the main data in this project is the review contents, I utilized the nltk package to for stop words removal and stemming to deal with noisy data.

When it comes to the workflow of my source code, my source data already provides me with three features - useful, funny, and cool. I decided to focus on coming up with new features related to the content of reviews and extract them accordingly. First of all, I defined some features such as length of the review and number of words in the review. I also wanted to include some more meaningful features. For example, I wanted to find some of the most frequent words in positive reviews that are not in negative reviews, and verify how many of reviews containing such words are actually positive reviews. In my sample data, I chose five words from the frequent list to be such positive words and turned out that among all reviews containing those words, over 80% of them are positive reviews. Therefore, I added

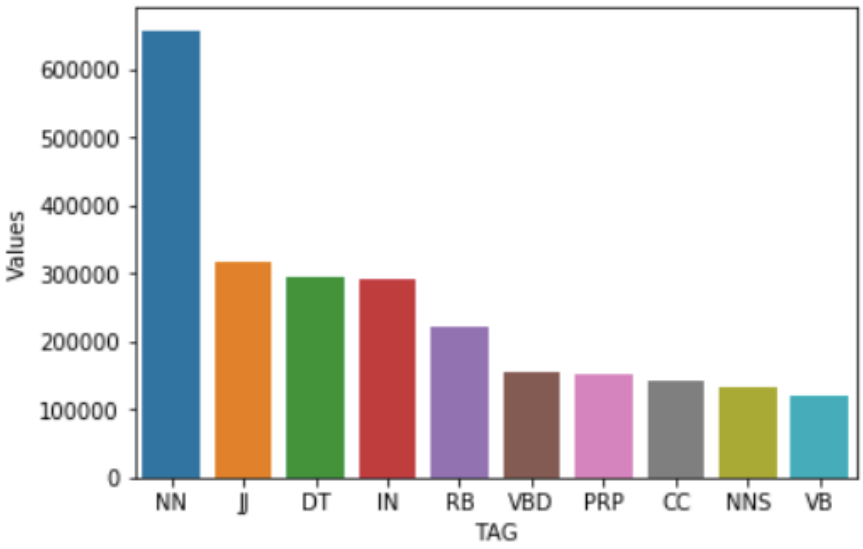
features called has_poswords and has_negwords to keep track of whether a review contain the positive and negative words.

As for the problems I encounter, it was hard for me to come up with features. As for examples given in the class, we did not have to think about features. Therefore, I search for possible types of features for text online and then choose from them. I come up with positive words and negative words list by myself.

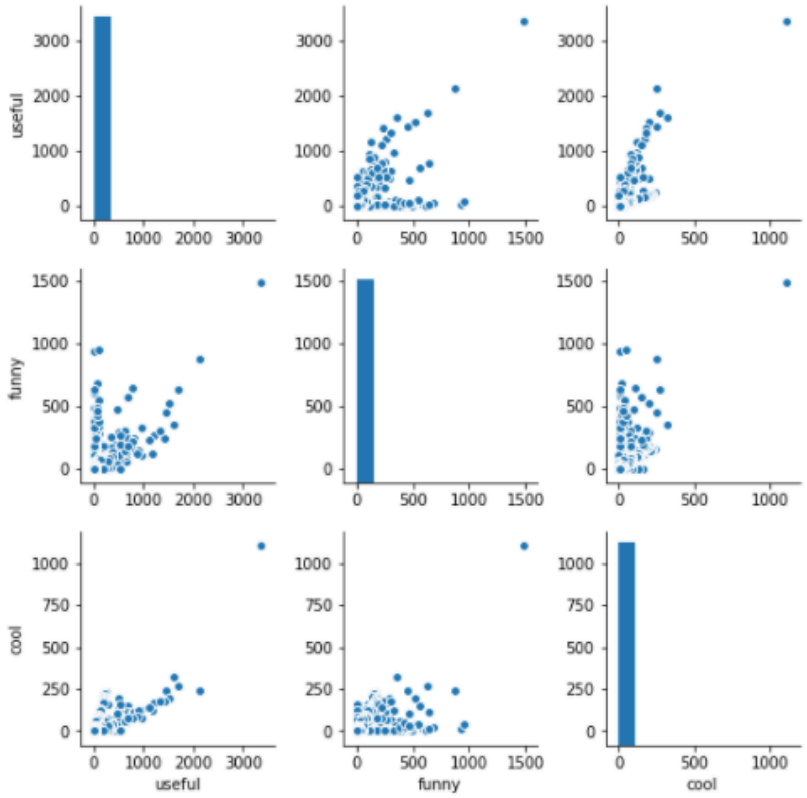
Analysis and Results

To begin with, as shown in the word cloud below, most popular words users have written in positive review are food, place, great, love, got etc.

In addition, the 10 most frequent part of speech tags in the review text are: NN, JJ, DT, IN, RB, VBD, PRP, CC, NNS, VB.



Further, there is no correlation between useful and stars, funny and stars, and cool and stars since their correlation coefficient are smaller than 0.1. Besides, there is a positive moderate correlation between useful and funny, funny and cool and useful and cool. Therefore, if a review is tagged as useful, it is more like to be tagged as funny or cool. If a review is tagged as funny, it is more likely to be tagged as cool.



Finally, the most reasonable feature importance is has_poswords. The accuracy of random forest classifier is 0.71.