A Beginner's Journey to Data Science: Attempt on Yelp Dataset Challenge
Angela Truong August 24, 2015

Background information

Assuming not everybody are yelpers, Yelp is a website for users to share their experiences at different venues and provide information/recommendations other users would find helpful. Yelp started in 2004 and has been growing ever since by selling ads on their site for featured businesses. With mass amount of participants known as yelpers, Yelp had to create their own automation software that scans reviews and determine if a review should be featured based on credibility of the user. Many businesses rely on Yelp for potential customers to come visit their establishment. By showing only reliable reviews for the business, it helps the whole yelper community from being influenced by fake users who are only trying to self-benefit. Examples of self-benefiting yelpers are those who write positive reviews for friends' businesses or leave negative reviews for competitors. With reviews in mind, Yelp accumulates a lot of reviews and other site wide data that are useful for predictive business strategies. All this data welcomes data scientists to analyze what they can extract out of this mass of potential knowledge. This leads to the Yelp Dataset Challenge.

Getting Data from Yelp

The original project was to find the best tacos in the bay area by scraping data from Yelp. A lot of time was spent trying to find ways to attain the data. Yelp has an API available for use but was very limited in the data allowed. The API only allowed pulling one review for one business for each request. This action would be easier if I used the user interface on the website instead of actually using the API. This lead me to the Google group where API users can ask questions on how to use Yelp's API. A Yelp associate will answer some of the topics and I found one where a user is also trying to attain mass amount of reviews. Luckily, an associate decided to answer this specific topic and stated there is no way to attain all reviews for specific businesses. After the Google group, I found the FAQs page stating they were not providing special academic research access at this time. The FAQs page then directs to academic datasets where there were preset data from businesses near 30 local schools. These datasets would not work for my project since there are only 2 schools out of the 30 located in the bay area. The next attempt was to manually grab reviews from the yelp website by looking into their JavaScript code; an error code popped up stating blocked by client. Finally after spending a good amount of time trying to attain data, Ramesh recommended the Yelp Dataset Challenge with data ready for analysis. The datasets are easy to download and all came in json file format.

The Yelp Dataset Challenge datasets consisted of:

The Challenge Dataset:

- **1.6M** reviews and **500K** tips by **366K** users for **61K** businesses
- **481K** business attributes, e.g., hours, parking availability, ambience.
- Social network of **366K** users for a total of **2.9M** social edges.
- Aggregated check-ins over time for each of the **61K** businesses

Cities:

- U.K.: Edinburgh
- Germany: Karlsruhe
- Canada: Montreal and Waterloo
- U.S.: Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison

The challenge from Yelp was very open ended and gave participants free range of what they would like to discover from the datasets provided. Yelp provided many questions to get participants started with reviewing their data. There were many directions to start with and I decided to go with the natural language processing. The questions chosen to answer were from Yelp and are as follows:

- How well can you guess a review's rating from its text alone?
- What are the most common positive and negative words used in our reviews?

Choosing features to use for data analysis
Before choosing any features, the datasets needed to be converted from json files into csv files. I converted the files by first reading into the json files, creating dataframes for each dataset, and saving these dataframes into a csv file. The purpose of converting to csv files is for easier review and navigation of the data. There were many examples from the Yelp github on how to convert the datasets into csv files but I found my process of conversion easier to follow by using pandas.

With all files converted, the only datasets used to answer the questions above were the Review, Business, and User datasets. My approach to answering both questions was to combine all three databases and have a combined main dataset to navigate through. The main dataset needed is the Reviews dataset but Yelp had over one million reviews for different businesses. In order to have a dataset for my computer to handle, I randomly sampled 200K rows from the Reviews dataset by creating a function using the random module. The sample then was saved as a new csv dataset and viewed for what is consisted in the data. When I showed the tail() of the data, there was actually a review in a different language. This opened up the possibility of other reviews being in another language I would not be able to comprehend. Reducing the dataset even more, the TextBlob module was used to filter out all non-English reviews. This code took some time to run and python was finally able to determine 1,394 rows were not English and was dropped from the sample 200K.

After reducing the Reviews dataset, the Business dataset was next to be filtered by only Vegas businesses. Reviewing the data in the Business dataset, this dataframe was also very large for python to process at a reasonable pace. The decision of choosing to focus on Vegas located businesses was a conscious choice since Vegas had the highest amount of business information after doing a value_counts().

With User dataset, Yelp did not use their automation system to filter out the fake yelpers appearing in this dataframe. I decided to create my own classification by separating creditable yelpers from the bad yelpers. The qualifications to become a creditable yelper are to have written over 100 yelp reviews and been an Elite for at least 5 years. The step taken to filter for users who have written over 100 reviews was by creating a function and adding a feature. The feature added was the ‘Useful’ column and broken down into three categories of: Creditable >= 100 written reviews, Okay >= 50 written reviews, and Bad = less than 50 written reviews. To determine Elite status for each user, the length of each row in the ‘elite’ column would have to be longer than 25 since the data lists elite status by year (EX: “2009, 2010, 2011, 2012, 2013”). Another feature was created called ‘elite_5_years’ that was filtered by a function by having it return TRUE if the length of ‘elite’ column is longer than 25 characters. The dataset did not completely generate into a perfect dataframe format due to the ‘friends’ column listing too many ID’s. The ‘friends’ column had to be deleted in order to move forward with perfect clean data. Once both of these newly created features were set, another feature called ‘cool_people’ was added to determine which users are truly qualified to join the Reviews dataset for further filtering.

Now with both User and Business datasets filtered for creditable users and Vegas located venues, a new function was created in order to determine which reviews from the Reviews dataset are creditable users and are in Vegas. The created function is a python version of vertical lookup in excel. The function

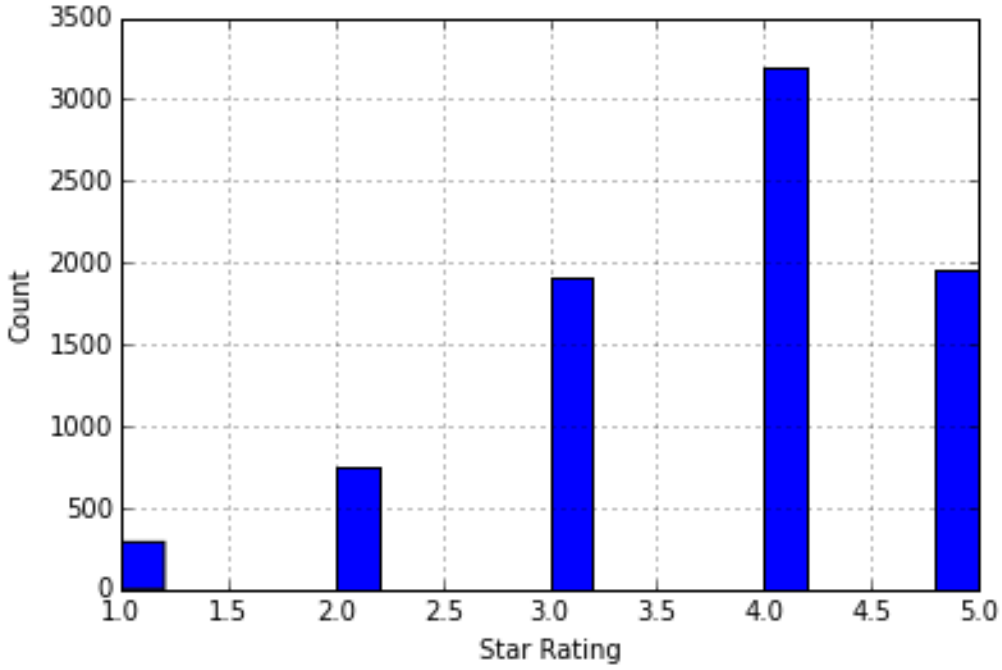
looks up the common key between two datasets and will fill in a chosen column to line up with the appropriate row. The Reviews dataset and the Business dataset have the common key of 'business_id' and Users dataset common key is 'user_id'. Now with a complete dataset, we can move on to actually analyzing the dataset to answer the questions.

```
              business_id        date               review_id  stars  \
0  DZGWM0o7GC_NYYpIX-MBfg  2014-11-15  SmbhLBXH-7ToJsDjJRIxcA      4
1  sC58XGwsZowovN1CnKktOg  2011-02-28  d5Lv61cghjvx9nAP-aTO1Q      4
2  Z0kyK8wCBNGkkUT9UrMWcg  2011-02-17  RzqQZp99OdBIzCxGCr-Cew      3
3  Cu3L4mNS_Np34wD7Af2fCw  2011-12-20  9e6B9k4gbIvfTaSxOLjoHQ      2
4  gy3WtUaSqZNzxLcMgAbJHg  2011-01-14  tGyU19fgXX9zMYwFd77a5Q      1

                                                text    type  \
0  Items Selected:\nChef Zen\nLatiya - Custard Po...  review
1  Vegas is a town that seems to have some sort o...  review
2  We had a really nice meal at Del Frisco's, wit...  review
3  Ate here twice in a week, once for breakfast, ...  review
4  The last two times I've been here the food has...  review

                  user_id                                votes English  \
0  5lq4LkrviYgQ4LJNsBYHcA  {u'funny': 5, u'useful': 9, u'cool': 10}    True
1  4p-qAdc_ZLXfieNwaZGNGA   {u'funny': 1, u'useful': 2, u'cool': 2}    True
2  ZG81nYjerp82pQas1mdjtQ   {u'funny': 0, u'useful': 0, u'cool': 0}    True
3  htJ84ka4CyUSijmfSkmVgg   {u'funny': 0, u'useful': 0, u'cool': 0}    True
4  KkEFlQAmrKf6BtPVi6EU7g   {u'funny': 1, u'useful': 5, u'cool': 2}    True

        city cool_people
0  Las Vegas        True
1  Las Vegas        True
2  Las Vegas        True
3  Las Vegas        True
4  Las Vegas        True
```



Modeling Process for both Questions
How well can you guess a review's rating from its text alone?

My first approach to answering this question was using Sentiment Polarity for each review in the dataset. Sentiment Polarity comes from the TextBlob module. When applied to each review in the data, there comes a score in the range between -1 to 1 to determine if a group of words are negative or positive. After attaining the score for each review, the scores were not a good predictor for star ratings due to negative sentiment polarity for four-star rated businesses. The reviews in the data were long which created difficulty of getting accurate scores for prediction.

My second approach is to use Naïve Bayes with CountVectorizer and Term-Frequency – Inverse Document Frequency (TFIDF Vectorizer). The first step is to separate training and test sets for the data. For both Count and TFIDF Vectorizers, the parameters were set to exclude all English stopwords, group phrases up to 3 words, and using 50K max_feature to reduce variance. Count Vectorizer came up with an accuracy percentage of around 50% and TFIDF Vectorizer had an accuracy percentage of around 40%. These percentages were the highest accuracy I could get after trying different parameters. There was a moment when I resulted in 4% for Count Vectorizer and 3% for TFIDF Vectorizer. This might be easier if I split the star ratings into two sections where 4+ stars are in one class and 3 and less stars are in another class. I would be able to calculate the ROC and AUC for sensitivity and specificity.

What are the most common positive and negative words used in our reviews?

The approach used for this is the natural language toolkit and the regular expression module. First step is to tokenize the words in the review column and exclude all the non-words from the reviews. After singling each word, the words had to be chained back together and classified if the word is positive or negative. I found a dictionary of 6800 positive and negatives words from http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon and used it to determine if the tokenized words were positive or negative. The downside to this method is that the only words classified are words existing in both the data and the dictionary. With all words classified, positive and negative words are separated into different word clouds. I decided to use a white background for positive words and a black background for negative words. Additional to choosing colors, only 2,000 words would appear in the word cloud and all stopwords are excluded. The results are below:

Challenges and Successes

One of the challenges for this project was to find clean data to work with. With all the time I spent trying to find ways to scrape from Yelp, that time could have been allocated to more analysis. The planning of how to organize the dataset is easier planned than actually adjusting the data. After all the function creations, feature selecting, and combining of datasets, the main dataset was finally ready to be analyzed. If I didn't have clean data to start out with, this process would have taken longer. I fell into the trap of focusing too much on my data instead of spending more time analyzing.

The challenges of analyzing the data were the constant errors that continued to come up and the exactness of coding. A lot of my time was spent researching for python syntax on how to make python push out the results I wanted. Most of the time errors continued to come up and not give me a more constructive lead on how to fix the error. Being a beginner of trying to understand a new language is difficult. Not only was I new to coding, the challenge of learning how to use a new operating system. I've always been a windows user but I decided to purchase a mac since it is the recommended industry standard. I had a difficult

time finding out how to open a rar-formatted folder. The 6800 word dictionary came in a rar file and it took me two downloads of random software and further research before I got it open without having to use the two software programs. Once I was able to open the folder, I had to convert both negative and positive words from text files to csv. I attempted to use python for this conversion but decided to use excel instead since it was the faster route.

Conclusions and Key Learnings

The conclusion to this project from a beginners' point of view is that this class was worth learning and the Yelp Dataset Challenge is definitely a good start into the Data Sciences. I learned more about the difficulties of being a data scientist more than I did the dataset. There are many things I have to personally work on which are:

- Do not focus too much on the data and allocate more time to analysis
- Find already clean data if possible to avoid cleaning
- Do not rely on excel knowledge to analyze data

For specifically the Yelp Dataset Challenge:

- Find a better way to calculate the sentiment polarity of reviews
- Find another way to classify negative and positive words so there won't be a restricted dictionary