Capstone Project #1: Yelp Health Rating Predictor

Objective

The application "Yelp" was established as a platform for users to leave their experiences on businesses and their individual ratings for others to see. As a Yelp reviewer and user, it would be great if Yelp provided a health rating score that was user defined. Yelp currently utilizes a score from a privately owned company, HDScore, which pulls data from local inspections and presents a health score. However, local inspection data can be slow to release and businesses can skip over guidelines in between inspections. So, my objective is to utilize user reviews to create predictions of the current health grade.

Target Audience

My target audience would initially be the Yelp user base, but I believe that it can extend to users of other review sites as well as the establishment owners. Current Yelp users and other viewers would be able to utilize user-defined health ratings to determine where they dine. Business owners will have an understanding of what patrons think about their food hygiene and decide on how to improve themselves.

Data

The data will be pulled from Yelp Challenge through Kaggle. This data consists of the user text reviews and business data that I will tie with a dataset of local health inspections available via the state or county's public inspection database.

Approach

1. Pull the Yelp data, clean/analyze and determine states or counties where there is an abundance of data.
2. Pull local inspection data based on location(s) determined in #1. Clean/analyze the data.
3. Combine user review data, business data and health ratings.
4. Test multiple machine learning algorithms, tune parameters and select the best algorithm.
5. Run the selected algorithm to predict health ratings.
6. Provide insight and recommendations based on the results.
7. Identify challenges and provide recommendations on how to improve for future use.

Deliverables

1. Code to show the data, algorithm comparisons and prediction results
2. Paper to discuss specific approach, challenges and results
3. Presentation to target audience

Data Wrangling

For my first capstone project, I will begin by extracting Yelp data from Kaggle to determine the city where I should pull my local inspection data.

Going through the information on the Yelp review data, I see that there are 5,261,688 entries but there is no data column to determine the city the business is located. However, there is a business id that I can use to query the data I need from the business data. The business data has 192,609 rows with 14 columns that cover the location, rating, attributes and hours. I begin by subsets of each data to merge on the business id only. This step resulted in a data frame with 5,111,225 entries which is 150,463 or about 2.86% reduction from the initial review data. The data that have been dropped were reviews with missing business ids and businesses without reviews. As I am trying to determine the city with the largest amount of reviews, I will leave these data dropped.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5261668 entries, 0 to 5261667
Data columns (total 9 columns):
review_id      object
user_id        object
business_id    object
stars          int64
date           object
text           object
useful         int64
funny          int64
cool           int64
dtypes: int64(4), object(5)
memory usage: 361.3+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5111255 entries, 0 to 5111254
Data columns (total 2 columns):
business_id    object
city           object
dtypes: object(2)
memory usage: 117.0+ MB
```

After merging the table and sorting the city values by its unique counts, it was determined that the highest review counts went to Las Vegas with a count of 1,593,922 which represents about 30% of the original review data. Next, I pulled the Las Vegas inspection data from the City of Las Vegas Open Data Portal. The establishment data contained 24,448 entries and contained information on the business locations as well as date/time and the result of the health inspection. Now the challenge of combining the establishment data with the yelp review data required a lot of processing power and time.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 24448 entries, 0 to 26492
Data columns (total 21 columns):
permit_number       24448 non-null object
facility_id         24448 non-null object
owner_id            0 non-null float64
PE                  24448 non-null int64
restaurant_name     24446 non-null object
location_name       24448 non-null object
address             24442 non-null object
latitude            24448 non-null float64
longitude           24448 non-null float64
city_id             24448 non-null int64
city_name           24448 non-null object
zip_code            24447 non-null object
nciaa               17411 non-null object
plan_review         0 non-null float64
record_status       24448 non-null int64
current_grade       24448 non-null object
current_demerits    24448 non-null float64
date_current        24448 non-null object
previous_grade      24360 non-null object
date_previous       24360 non-null object
search_text         24448 non-null object
dtypes: float64(5), int64(3), object(13)
memory usage: 4.1+ MB
```

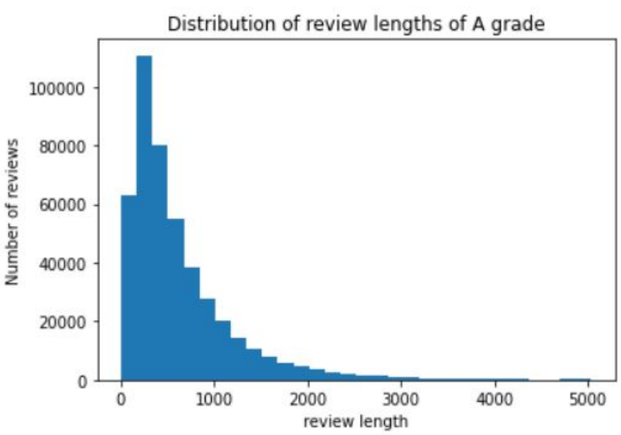
The combination of the two datasets results in 461,358 reviews that were tied with health ratings and Yelp businesses.

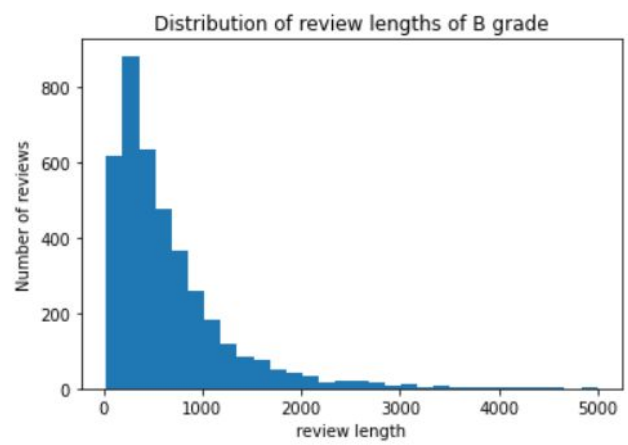
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 461358 entries, 0 to 697176
Data columns (total 42 columns):
address               461358 non-null object
attributes            453101 non-null object
business_id           461358 non-null object
categories            461346 non-null object
city                  461358 non-null object
hours                 431131 non-null object
is_open               461358 non-null int64
latitude_x            461358 non-null float64
longitude_x           461358 non-null float64
name                  461358 non-null object
postal_code           461358 non-null object
review_count          461358 non-null int64
stars_x               461358 non-null float64
state                 461358 non-null object
permit_number         461358 non-null object
facility_id           461358 non-null object
```

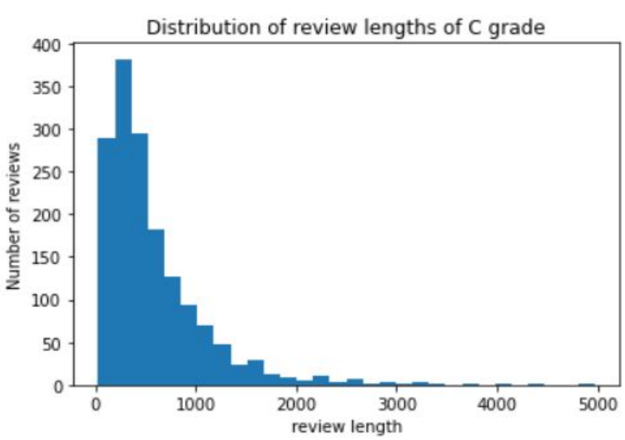
Statistical Data Analysis

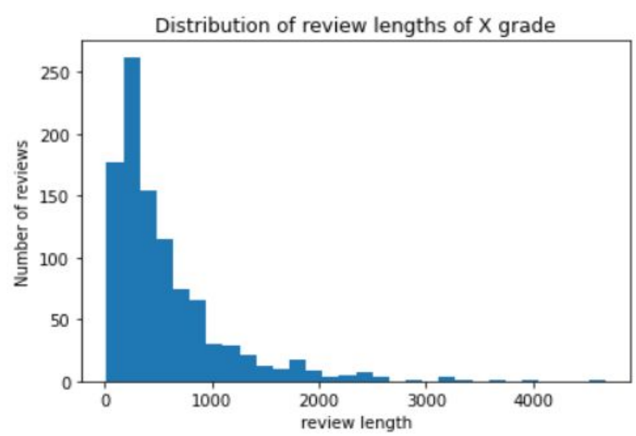
What factors can determine a health inspection grade? One would think that if a patron has a negative experience, they may leave a lengthy response detailing what they were upset with or a short response because they don't want to waste time on something that left them a bad experience. Based on these two assumptions, there could be no clear cut expectations for length of reviews in determining a grade. However, what is the point of statistical data analysis if you do not use it to prove this point?

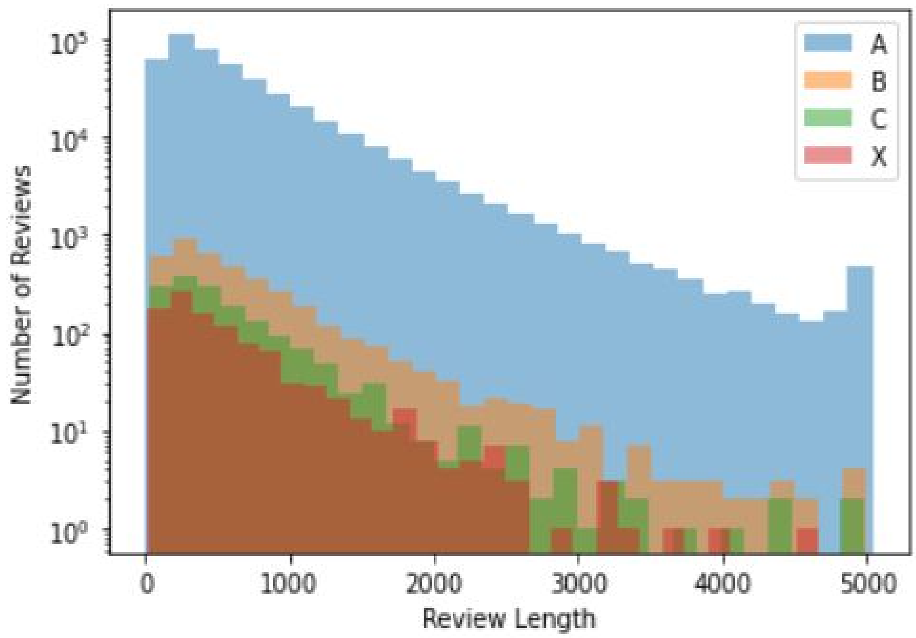
Firstly, it is important to separate the four inspection grades (A, B, C, X) to their individual dataframes. After the separation, it may be nice to view each grade's review count versus its length. Plotting each histogram reveals there is no visible difference among the grades. However, a histogram containing all grades together reveals that there are differences in quantity of review and some differences in review length distributions.
In order to better understand if each dataset's review length is truly different from the other, it is necessary to perform a two sample t-test. The following parameters are used for each two sample t-test between the letter grade combinations: alpha of 0.05, null hypothesis that both grade's mean review length are the same, and alternate hypothesis that they are not equal.











Running the 6 different combinations ($_4C_2$), it is shown that the only combination where the null hypothesis cannot be rejected is when C and X are being compared. An assumption of why this could be the case is that restaurants with low C grades are very close to receiving an X grade, so the reviews for both are similarly negative.

C vs. X

Null Hypothesis: Mean of C reviews = Mean of X reviews

Alternate Hypothesis: Mean of C reviews != Mean of X reviews

```
Ttest_indResult(statistic=0.7867981187165051, pvalue=0.4314715709457597)
```