Report

Resources to view the project:

To view the static implementation :
https://nbviewer.jupyter.org/github/AbhishekEaswaran/Yelp-Dataset-Sentiment-Analysis-Machine-Learning/blob/master/Project/Main.ipynb

To view the interactive implementation:
https://mybinder.org/v2/gh/AbhishekEaswaran/Yelp-Dataset-Sentiment-Analysis-Machine-Learning/master

Note: The interactive implementation had to be done on a small subset of data and is for interaction purposes to see the system in action.

Source Code: https://github.com/AbhishekEaswaran/Yelp-Dataset-Sentiment-Analysis-Machine-Learning

Notes:

1. The static implementation is the nbviewer which shows the run notebook with implementations.
2. For the interactive implementation, I have created a binder project. Please run the ipython notebook by opening the main.ipynb and executing the cells.
3. The github repo contains the source code of the entire implementation of the 2 problems.

Which dataset and why I chose it:

I wanted to choose a dataset that would help me solve the 2 problems and most importantly, demonstrate my skills that match with the job description.

The dataset I chose was the yelp dataset. The dataset used is made available by Yelp for use in personal, educational and academic purposes. The dataset used is in JSON format. There are more than 4.7M Reviews written by 1.1M Users for 156K businesses with 200K pictures from 12 metropolitan areas. This data can be downloaded from → https://www.yelp.com/dataset

However, for my analysis, I have taken into consideration 200,000 reviews.

Some key points:

1. The dataset contains reviews written by actual users. This gave me a good platform to demonstrate my skills in Natural Language Processing.

2. It contains very few columns for actual visualizations. In the real world, finding the right features is scarce and I wanted to challenge myself by taking less features and coming up with the right visualizations, statistical analysis and Machine Learning solutions.
3. It aligns well with the requirements discussed during my first call with Maria. I wanted to leverage the textual data and a few features to come up with meaningful insights.

I have built two systems that tackle two different sets of problems.

The Two Problems:

Part 1 :

Providing meaningful insights based on written reviews of users for your business!

To create a system which helps business owners analyse their performance based on user ratings and reviews. Generally, established businesses have plenty of reviews and ratings and it is difficult to keep up with them as it's almost impossible to go through thousands of user reviews for an individual. This system helps making life easier when the number of reviews is very large and it is virtually impossible to go through each review.

The designed systems analyses all the reviews and ratings for a business and provides essential data about:

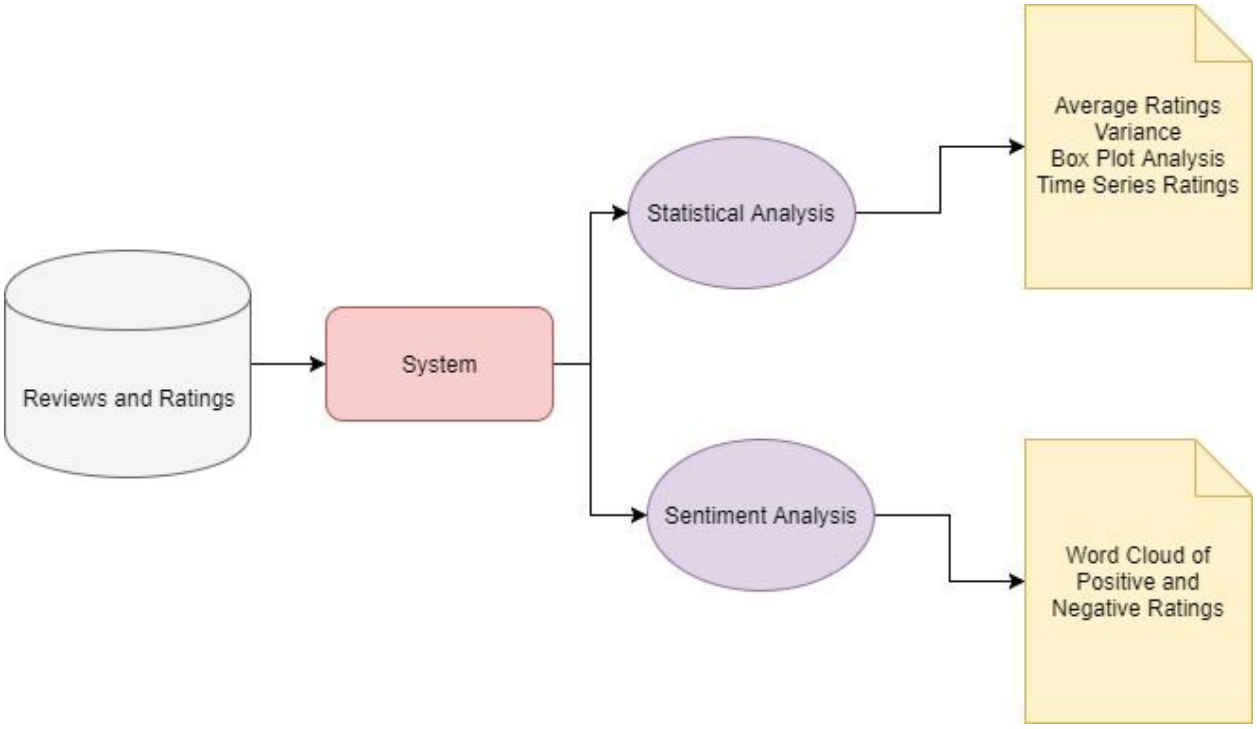
1. Average ratings from all users.
2. Statistics of ratings
3. Visualizations of ratings
4. Time series plot of average ratings.

Since reviews are being posted continuously, this system provides a good metric to judge experimental features like adding or removing a service, changing the ambience, new personnel, quality of product, etc.

The system processes the reviews and ratings data in the following manner:

1. Identifies the most frequent words from the ratings.
2. Runs sentiment analysis to perceive the nature of review: positive or negative.
3. Consolidates reviews in the form of a word cloud to identify positive and negative reviews along with relevant statistical data.

The figure below summarizes the system and my approach.

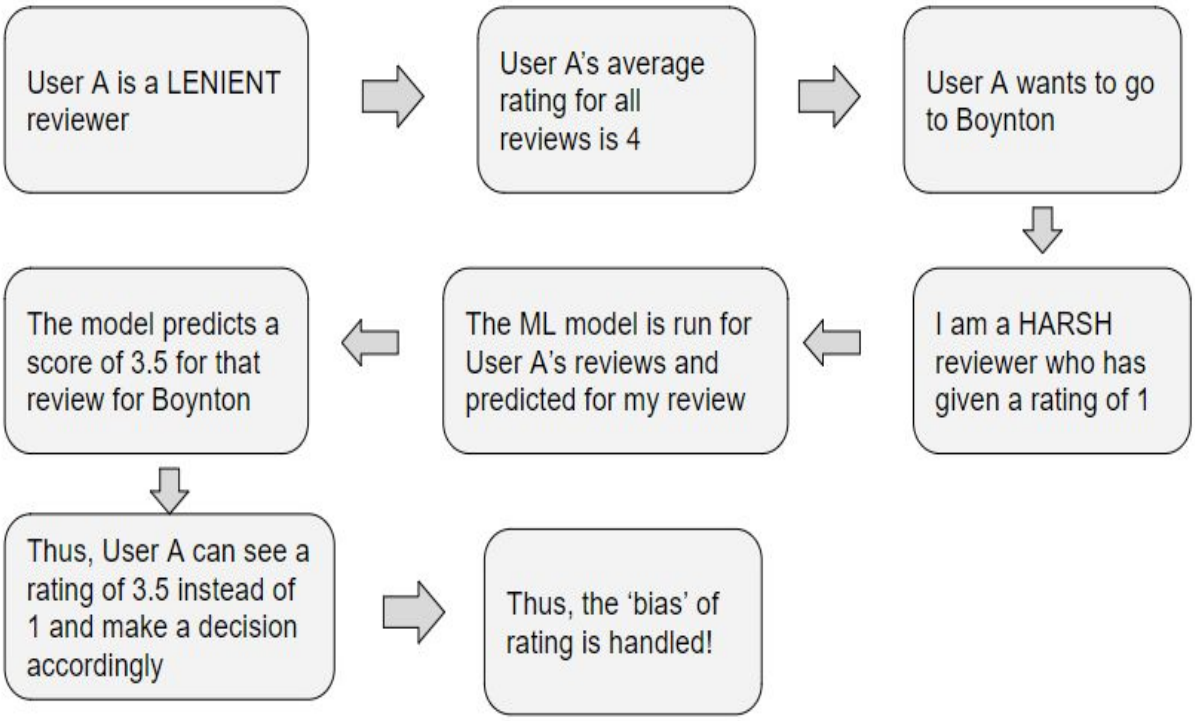


Part 2 :

Handling Biased Reviews by Using Text Mining!

The aim of this system is to perform analysis on the reviews and try to predict the possible rating for each type of user by eliminating bias. This system acts in the following way:

Build a USER SPECIFIC rating prediction model which acts as a prediction system, before the user could actually experience it. It will take all the reviews by one user and create a user-specific model and try to predict the possible rating of corresponding reviews given by other users for their texts.



Approach

Steps:

Step-1: Extract reviews for users who have written the maximum reviews. From the dataset, I have extracted the top reviewers.

Step-2:

a. **Collect reviews and ratings for one user -** This step involves collecting all reviews/ratings of one user and using that data to train the model.
b. **Apply td-idf / bag-of-words-** This step involves converting the raw text into a list of vectors. Converting this raw text to vectors can be done using these 3 techniques. tf-idf converts the words to vectors by finding out the occurance of a word in a document and offsets that by finding out the occurance of that word in the entire corpus. Thus, the obtained word's vectorized value will have more importance if it is rarer. In case of bag-of-words, the word is directly queried over the entire corpus and then the vectorization takes place.
c. **Build a Machine Learning model -** This is the step where for every user, a

machine learning model is built which can be used to train the model for that particular user and given an unseen test data of raw review words, can predict the approximate rating for that particular user whose model has been trained on. This is the main step where the bias handling takes place.

Validation of the Machine Learning model : For testing purposes, let's assume I am user A and I have another user B. I have built a model for user A that learns the patterns and the 'harshness' of user A. I input user B's 'text' and see the rating given by the model. I then compare it with the average rating that the user has given and validate my model accordingly.

Implementations:

Part 1 :

1. Text cleaning (stopwords removal, dictionary check, letters, meaningful word check)
2. Widget implementation to interactively choose from dropdown
3. Statistical analysis of individual businesses
4. Histogram, Boxplot and time-series plots for analysis
5. Word counts using Pretty Table
6. Sentiment Analysis of each review
7. WordCloud of all words and positively spoken words

Part 2:

1. Count Vectorizer to obtain vectors from words
2. TF-IDF Vectorizer to obtain vectors from words
3. Gaussian Naive Bayes
4. Linear SVM
5. Random Forest
6. Logistic Regression
7. Grid Search for hyperparameter optimization

Tools:

1. Jupyter Notebook
2. Web server hosting of the jupyter notebook

3. Python (Numpy, pandas, matplotlib, nltk, scikit)
4. Linux and Command Line