DATA301 Project Proposal

Student Name: Moses Bernard Velano
Student ID: 83396373

Summary
The proposed project applies the use of python and Apache Spark to analyze a chosen recommender/personalization data set provided by the UCSD. In my project, I have chosen the BeerAdvocate dataset which contains multi-aspect beer reviews. The research question that I will be proposing in this project investigates "what is the relationship between a customer's text reviews and a beer's features to the overall impression of the most popular beer." I will use the TF-IDF algorithm and the intended result of this should show three categories (negative, neutral, and positive) from overall impressions and the associated words from a customer's text review towards a specific beer. After that, I will use cosine similarity of a beer's features (appearance, aroma, palate, and taste) and compare it with another beer from a customer's review. The result I'm hoping to get is, to understand the relationship of a beer's features and customer's text reviews relating to the overall popularity of a beer. The significance of this result is it can help beer drinkers to easily choose the most liked beer by recommendation, promotion, and displaying the text reviews for customers to see.

Motivation
This research question is relevant to me because I do not like drinking any type of beers and someday I would like to try a beer that best fits my desire. As a student at University of Canterbury we know that there is a high volume of students who drink beers. Therefore, I want to look at the most popular beer in the BeerAdvocate data set and understand the pattern of a customer's text reviews relating to the overall impression of a beer and the score ratings of a beer based of its features. We can connect the findings from our algorithms to the factors that motivate UC students into purchasing beers. This research can help beer drinkers find the best beers that satisfy their preferences when it comes to beers. It can also help others into avoiding a very bad tasting beers, in that way buyers do not waste any money and undrunk beers. Otherwise, other UC students who do not like drinking beer and are wanting to try it in future, we can use the result from this research to help find the best beer that would certainly satisfy their preferences.

Background
The BeerAdvocate data set provides various characteristics that determine the attributes of a beer. An example of this is, the Average By Volume (ABV) which refers to the percentage of alcohol given a volume of a beer. We also get the name of the beer (name), unique beer ID (beerId), where it was made (brewerId), classification of a beer (style). The data set also provides individual score ratings on the beer such as appearance, aroma, palate, taste, and the overall impression. It also informs us about the specific point in time when a customer user submitted a review of a particular beer (time), the reviewer's user profile ID name

(profileName), and personal comment reviews (text). We can use TF-IDF in our research to help us find each term frequency of words from the customer text reviews and relate this back to the overall categorized impression of a beer. This is achievable because we know that TF-IDF algorithm weighs the significance of a specific term frequency word of single reviews from a customer user by looking at how frequent a word appears in that review (Term Frequency). We then evaluate how frequent a term word appears across all user reviews which is all document in the collection (Inverse Document Frequency). Then we just multiply TF * IDF. The cosine-similarity algorithm is necessary in our research to help us understand the relationship of a user text reviews to the associated score ratings they gave on a beer's features (appearance, aroma, palate, and taste).

Research Question or Hypothesis

The research question is "what is the relationship between a customer's text reviews and a beer's features to the overall impression of the most popular beer.". This is relevant to our data set because this can improve accuracy and effectiveness of a beer company, a liquor store, or for frequent beer buyers when trying to get the most out of a beer. By using TF-IDF this will help identify the most important term words in each user text reviews. Which will be categorized in three categories of negative, neutral, and positive coming from the score ratings of overall impression a user gave. While cosine similarity algorithm measures the similarity of a user's score reviews on a beer's features between another user's review on a beer's features. With both algorithms, this allows future people to benefit from the given information as they will likely have a faster, easier and improve their decision-making when selecting a beer that best match their preferences. Additionally, this can help promote the best beer in the market as it includes user text reviews that explains the aspects of the beer.

Design and Methods

I will be using TF-IDF and Cosine Similarity algorithms to achieve the specific objectives of my research project. Before applying these algorithms, we first need to import the BeerAdvocate dataset into our Google Collab. We then implement specific code that will decompress a file that have been compressed using the "gzip" compression into a normal JSON file format for achievable latter processes. The following steps include, setting up local Spark Cluster, load BeerAdvocate dataset into an RDD, verify attributes of the dataset to ensure accuracy, cleaning and filtering the desired dataset. Finally, we can perform our TF-IDF algorithm which starts off with computing the Term Frequency (TF) for each term words in a user's beer review. The expected output for TF should consider how frequently a word appears from a singular review in the document (where individual user review = a collection of documents). Afterwards, we take the Inverse Document Frequency (IDF) for each term word in all documents (all the user's beer reviews). This is basically trying to find out whether a specific word used in a beer review would also appear in another user beer review. Then compute TF * IDF score for each term word in BeerAdvocate dataset and collect/display the overall impression score ratings that a user gave scaling from 1-20.

I am hoping to get something like this:

(1, [(integer, 'string'), (integer, 'string')……])

(overall_impression, [(integer, ‘string’), (integer, ‘string’)……])
(overall_impression, [(integer, ‘string’), (integer, ‘string’)……])
.
.
.
.
(20, [(integer, ‘string’), (integer, ‘string’)……])

The Cosine-similarity algorithm will support the reasoning for finding the most popular beer in the BeerAdvocate dataset. But before we use this algorithm, we would need to extract the attributes of a beer’s appearance, aroma, palate, and taste and put it as a list of integers. Next, we perform the cosine-similarity algorithm to calculate the cosine similarity between two list of integers (this is the user’s score reviews on the characteristic of a beer). Finally, we can connect the Cosine-similarity and TF-IDF algorithms together. Finally, we hope to retrieve a result of the most frequent words used from a user’s text reviews and the cosine similarity of users on a beer’s characteristic which hopefully would lead us to finding the overall best beer.

I will be using Figure 1 and Table 1 to guide me through my process on accomplishing specific objectives on my project.

Figure 1: Data Flow Diagram

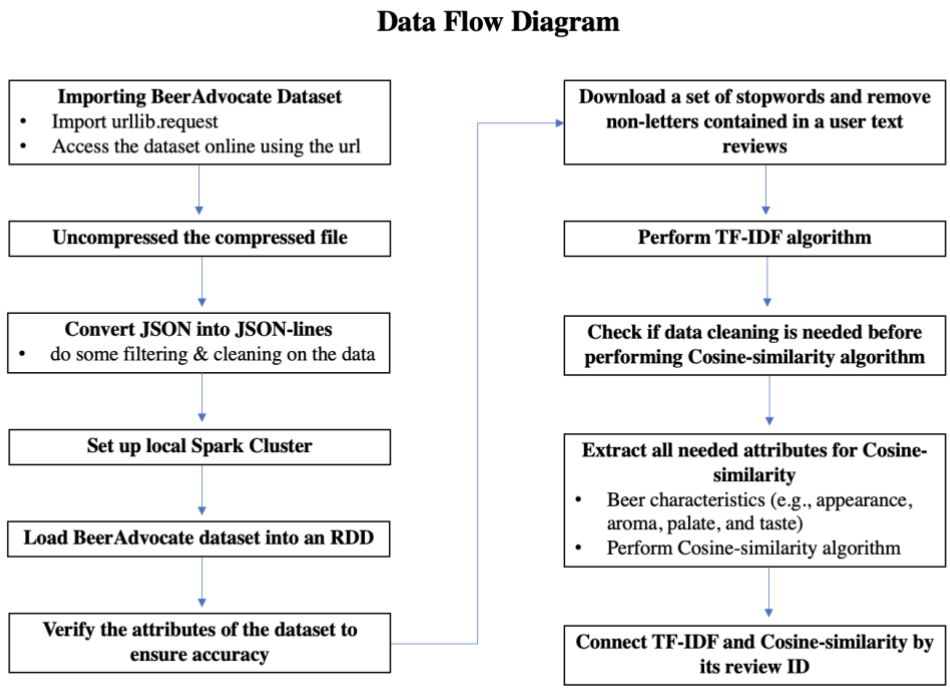

Table 1: Tentative Sequence

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Week 9	Review that the research question, goals, process, etc. is achievable and realistic.	Setting up the Google Collab environment.	Go to Lab session and verify with the tutor whether my project is proper.	Do some further research.	Review the project proposal and continue coding.	Download the data, play around with coding.	Rest day
Week 10	Prepare the dataset into a JSON-line format.	Continue yesterday's task.	If Monday's task done then move on to setting up local Spark Cluster, hence continue Monday's task. Get help.	Set up a local Spark Cluster.	Load Beer Advocate dataset into RDD and do some cleaning. Submit individual progress report.	Coding	Rest Day
Week 11	Continue coding	Coding	Coding, get help	Coding	Coding	Start writing final report	Continue doing both
Week 12	Coding & writing final report	Coding	Coding, get help, try finish coding by this time	Focus on writing final report	Submit both Software Implementation, Final Report		

However, some difficulties that I may come across is having to work with very large dataset. Since, the data format of the file is "json.gz" we would need to make sure that when downloading the data, the desired format should be in a JSON-line. It may also be difficult to extract the overall score rating on the impression of the beer per user in our data set and separate these into three categories of negative, neutral, and positive. I would also need to determine how to connect the most frequent used term words in the text reviews and categorize these in the three categories of overall impression score. Also, take all the user's text reviews (all documents) and compile the most frequent term used in the reviews in the three desired categories. Another difficulty that I may encounter is connecting the TF-IDF result to the Cosine-similarity result. Otherwise, filtering, cleaning, debugging, etc. on the data will be some of common difficulties. For limitations, there could be missing data in our BeerAdvocate dataset, and my skills in coding is limited therefore this would challenge me.

References

Laerd Dissertation. (2012). How to structure quantitative research questions. https://dissertation.laerd.com/how-to-structure-quantitative-research-questions.php

McAuley, J., Leskovec, J., Jurafsky, D. (2012). International Conference on Data Mining (ICDM). https://cseweb.ucsd.edu/~jmcauley/datasets.html#multi_aspect

Singh, A. (2015). CSE 225: *Assignment 1 Winter 2015*. Retrieved from https://cseweb.ucsd.edu/classes/wi15/cse255-a/reports/wi15/Alok_Singh.pdf

I also got my ideas from the Labs, Lectures, and the sample project code.