Coffee Recommender

Tofer Kim

Introduction

Coffee drinkers are becoming increasingly more sophisticated in their coffee choices.[1] With so many varieties of coffees out there--new blends and methods being invented all the time--a system that could recommend new coffees to drinkers would be beneficial for the user as well as the coffee roaster. Using a recommender, coffee drinkers will be introduced to new coffees that they would like and roasters could learn what coffees interest them.

This project features a item-to-item based coffee recommender system based on data scraped from www.coffeereview.com. The system considers 4,887 unique coffees and recommends a coffee based on the cosine similarity of specified features. The system can be customized to base recommendations from three criteria: text description, ratings & types, and a combination of the two. Furthermore, the system is able to recommend the highest rated within a defined range of similar coffees, or the overall most similar coffee.

The system can only provide results for coffees present in the dataset using it's unique web page slug. The dataset includes coffees that were reviewed as far back as 1997 and, unfortunately, may no longer be available. Next steps for this project involve conducting A/B hypothesis testing to determine the usefulness of the recommender system and further tune the considered criteria--consulting with coffee-tasting experts as well as gathering feedback from users.

1. Web Scraping

Coffee Review (www.coffeereview.com) is a coffee review aggregate founded by Kenneth Davids. He and his team have expertly reviewed over 5,000 coffee roasts since 1997. This notebook below scrapes through their website, collecting data on each coffee blend.

The first step of web scraping involves retrieving every coffee's *name* and *web-page slug* in order to then collect information from each coffee's web-page. Once those *slugs* are obtained, we are able to gather all the relevant text from each coffee's page and add them to a pandas DataFrame. Because the information is not organized in a neat *html table format*, RegEx is necessary in order to extract relevant data. With the power of RegEx, we can capture the relevant information within the text. For example, every coffee has a number that follows the word "Aroma: " which can be captured using the expression “Aroma: (\d*\.*\d*)” . This will return a float like ‘8.5’ or integer like ‘7’. We apply this strategy to all the information we want, for each coffee’s web-scraped text, and create a list of lists: giving us a DataFrame of all the coffees’ associated ratings and descriptions.

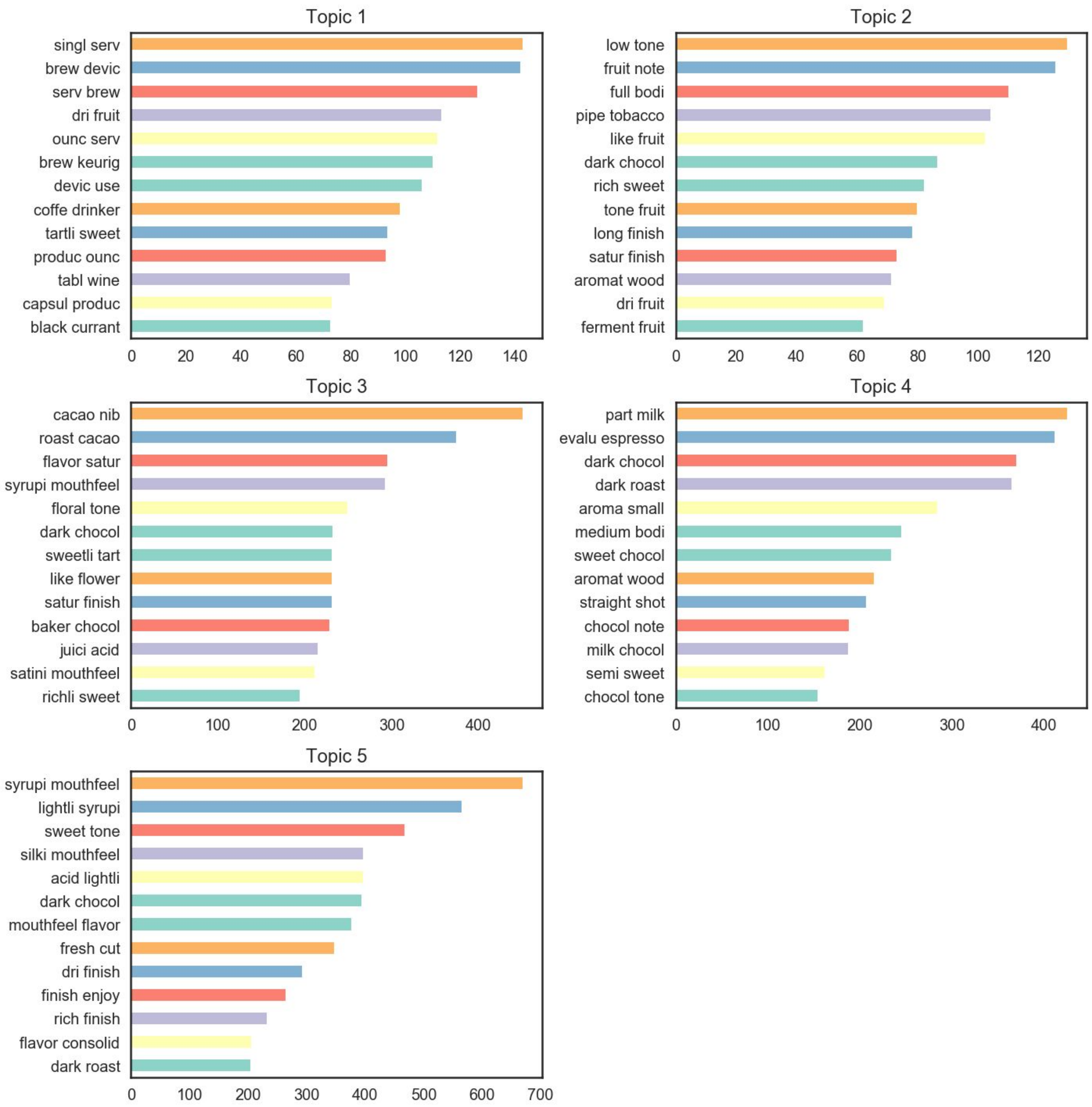
An additional scrape is used to gather more information regarding each coffee’s *type* (espresso, blend, organic, etc.) and *region* the coffee is from (Africa/Arabica, Hawaii, Central America, etc.). The advanced search feature on the website filters the coffees, and using the slugs associated with each filter, we can iterate through each search term and determine which coffees belong to each category. This information is added to the DataFrame as binary columns indicating whether or not each coffee belongs to that type or region.

2. Data Cleaning & EDA

Beginning with the text description of coffee, a few preprocessing steps are necessary in order to visualize and determine whether or not this text description is even useful. In order to clean the text, I removed all punctuation and stop words and stemmed each word. Stemming is a process that reduces words to their stem. For example, the words *chocolate* and *chocolatey* and *chocolates* all become *chocol*. This retains the relative meaning of each words and allows our data process to recognize them as having the same meaning, instead of representing three entirely different words.

Once cleaned, the text can be CountVectorized using n-grams of (2,2), meaning each word, instead of evaluated individually, is split into groups of two words. This is helps identify phrases such as *not acidic* instead of the individual words *not* and *acidic*, which can completely change the interpretation of the coffee’s description. CountVectorizer returns a count for how many times each n-gram appears in each description.

Latent Dirichlet Allocation is a method that essentially observes which phrases are commonly used together in a document in order to group phrases into separate topics. Adjusting the number of topics to determine, as well tuning the n-gram values, allowed us to find noise words like people’s names and *reviewer* that were not included in the original list of stop words. Settling on five topics, we visualize what they are below:



When LDA is implemented, it is crucial that the observer have deep knowledge of the documents themselves and what they represent. My interpretation of the determined topics is as follows:

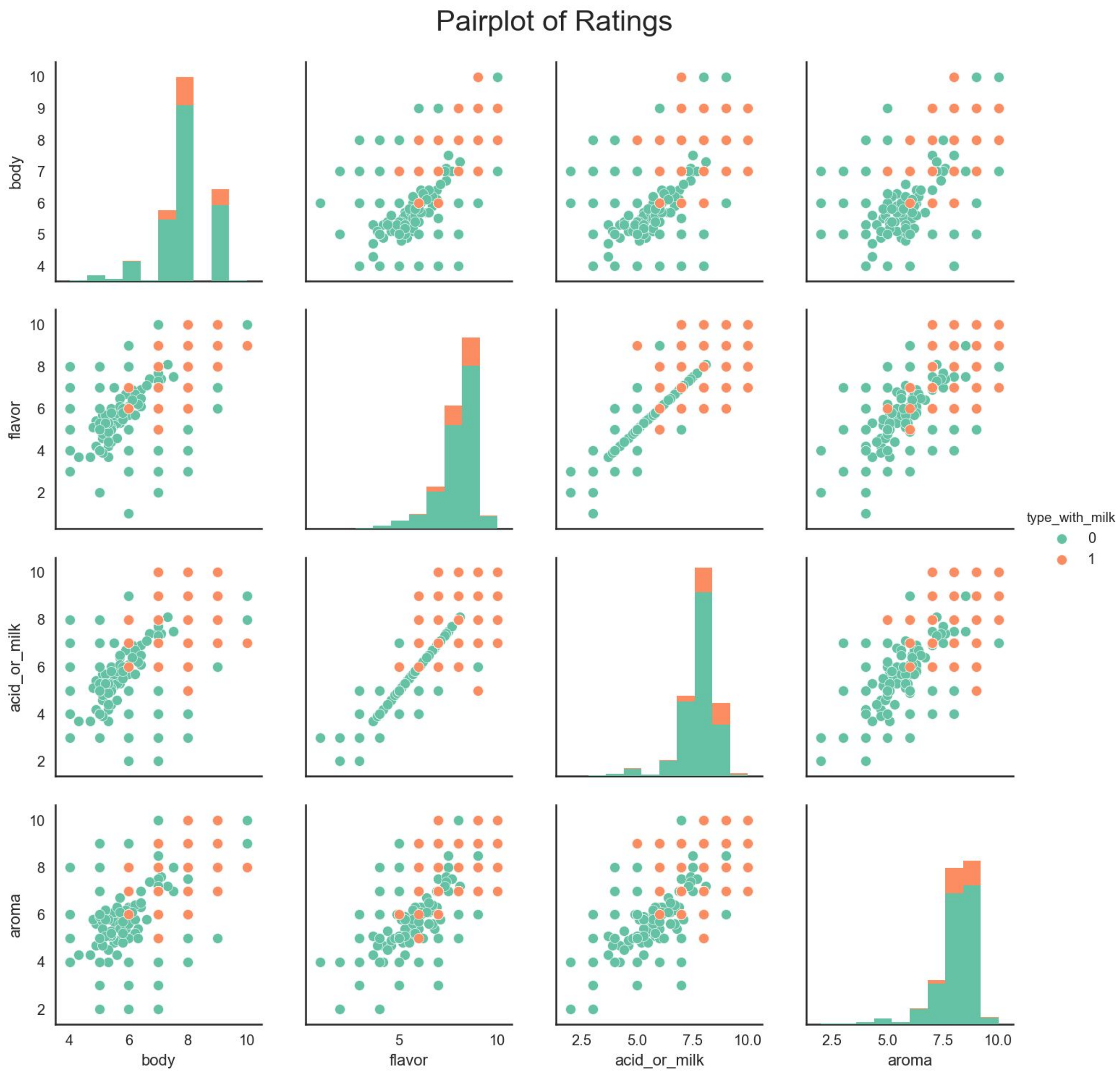
- Topic 1 - Keurig pod/capsule coffees
- Topic 2 - Fruity flavored coffees
- Topic 3 - Chocolate flavored coffees
- Topic 4 - Espressos or Lattes
- Topic 5 - Coffees with a distinct mouthfeel and texture

This analysis allows us to determine that our text is useful and indeed capable of differentiating and grouping coffees based on its text. We may even be able to use these topics as features when creating our final recommender system, comparing coffees that use phrases in the same topic as more similar.

Moving forward, we will not consider the following features:

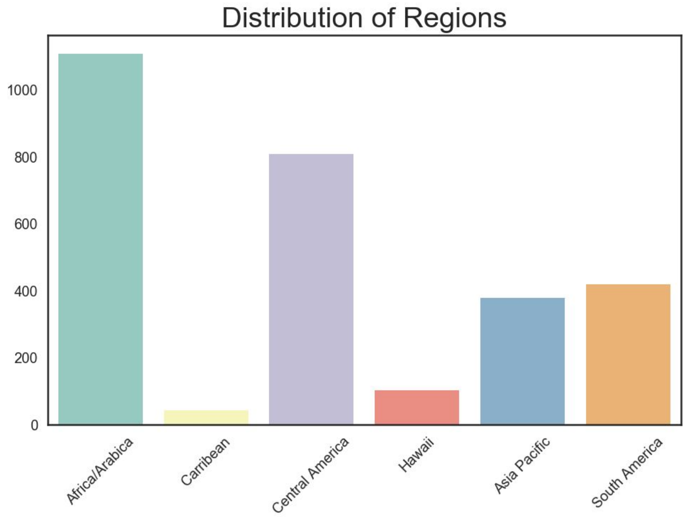
- Location: location of the coffees' roaster, which doesn't affect final recommendations
- Origin: text string that can essentially be replaced with 7 binary region_ columns
- Estimated Price: includes too many missing values, various currencies, and range over a long period of time (consider inflation). The price of each coffee, therefore, does not affect our final recommendations
- Review Date: Although a coffee may have been reviewed as far back as 1997 and may no longer be available, we will still consider older coffees as their descriptions may be useful for NLP (building our vocabulary/bag-of-words).
- Agtron: Number range of Agtron Score before and after roasting. The Agtron is a scientific measurement of roast color. The Agtron number is not relevant for our recommendations, but can be inferred from the *roast* level. More information on Agtron score can be found here: https://www.coffeereview.com/agtron-numerology/.
- Aftertaste: According to the website[2], aftertaste is graded very similarly to flavor, but may not be graded for some coffees (1008 missing observations); Instead of imputing these missing values, we can drop this column and focus on more important ratings.
- desc_3 (Notes): This section of text describes the coffees' roaster and background information on the region and roasting methods--irrelevant to our final recommendations.

The four most important ratings (score 1-10) to consider are: aroma, body, flavor, and acid.[2] In order to properly compare coffees based on their features, we need to ensure that there is no missing or outlier values. We will not be able to conduct necessary calculations with missing values and outliers will increase variance, potentially mischaracterizing similarities. We drop the 44 observations that having missing values for aroma, body, and flavor. Furthermore, we observe 667 missing values for acid and note that for the remaining 4870 observations, where these acid values are missing, there is a reciprocal value in the with_milk column. For the 8 observations where there is a value for both, both values are the exact same! Therefore, missing acid values we represent this disparity in a new column acid_or_milk, and create a binary column type_with_milk to signify this imputation. The remaining 187 observations where both of these values are still missing are then removed. This leaves us with a final count of 4,887 different coffees. To visualize the distribution of these features and their correlation we refer to the pairplot below:

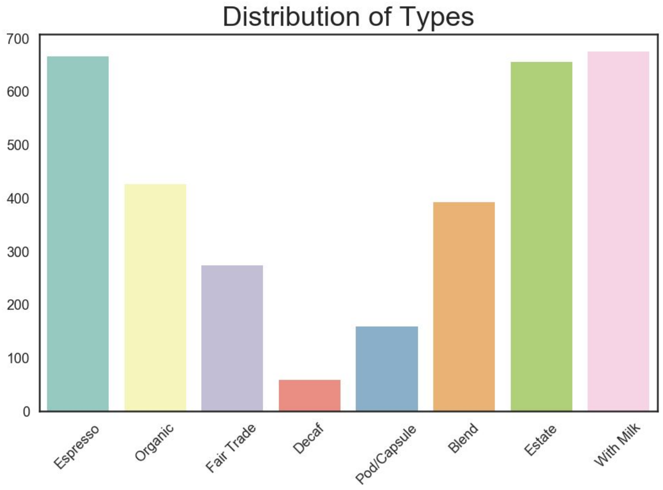


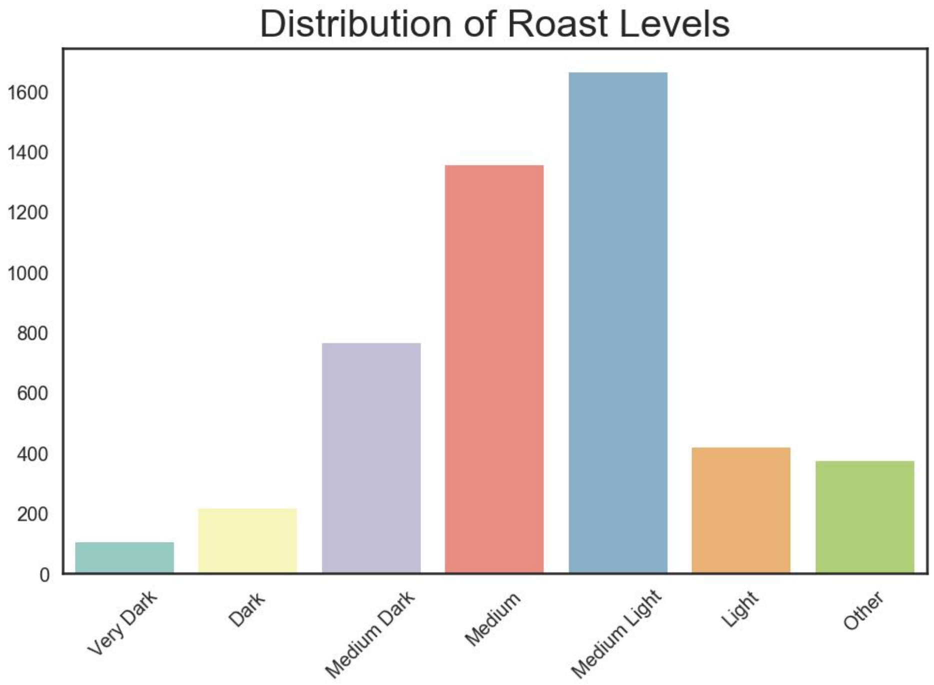
The four rating features associated with each coffee seem to be very linearly correlated with each other. That is, the higher the rating in one aspect, the higher it is in another. There is also little variance in each distribution. This will affect our recommender system in that, if we only consider these features for comparison, it will return many recommendations with very high similarity (cosine similarity close to zero).

Interestingly, observations with_milk appear to have higher ratings in each category, but not as much correlation. Possibly, these may indicate espressos or lattes, which usually have stronger flavor profiles, which are enhanced with the addition of milk.



Almost half of our observations (44%) are not associated with any region and even fewer are associated with multiple regions (0.7%). This could affect our recommender as it will consider coffees from the same region as more similar; When the value is missing, it may consider coffees from the same region as less similar. If the user considers the region of the coffee of interest as important, we will need to assume the coffee's region is listed. It may be possible to parse through the origin column to determine which region each coffee is from, although there are many values missing from this column as well. As for the types of coffees, for our purposes, we will need to assume that these values are properly assigned as we are unable to verify their completeness without more data.





If the coffees' roast type is unspecified, it may be due to the fact that the coffee was a blend, or was pre-bottled, or just not disclosed by the manufacturer. We will treat the unobserved roasts as its own feature and call it Other.

3. Cosine Similarity

The final decision of our recommender system relies on calculating cosine similarities between each coffees' data and determine which are more similar on a scale from 0 to 1. A score of 0 means the two coffees are indistinguishable based on its features; 1 means they have no similarity. Each coffee observation needs to represent a vector in space where cosine similarity can then be calculated by the formula:

{\displaystyle \mathbf {A} \cdot \mathbf {B} =\left\|\mathbf {A} \right\|\left\|\mathbf {B} \right\|\cos \theta }**Image**

Scaling our numerical features allows the data to be represented based on its variability within in each feature, as opposed to the given number representation. For our text data, we use Latent Semantic Analysis. This natural language processing involves running our text through a TF-IDF vectorizer and transforming the data through TruncatedSVD. TF-IDF works similarly to CountVectorizer, but instead of returning a count, it undergoes a further calculation that weighs less frequent phrases more heavily and vice-versa. Because this data becomes so sparse, meaning there are a lot of values of 0, we transform this data with TruncatedSVD, which essentiality enables us to reduce to the feature space and minimize the loss of variance.

This project actually considers three different recommender systems: one which considers only text descriptions, one which considers only ratings, types and regions, and the third is a combination of the two. Because our LSA is limited to 225 features, and we have 25 numerical features, the combination recommender is a mix of 10% numerical and 90% text based.

The recommender function is then designed for the user to pick which recommender to use. Furthermore, they are given the option for the recommender to select the overall most similar, or compare the coffee of interest to its n-nearest coffees and return the coffee with the highest overall rating. The system can only provide results for coffees present in the dataset using it's unique web page slug. The dataset includes coffees that were reviewed as far back as 1997 and, unfortunately, may no longer be available.

Next Steps

Next steps for this project involve conducting A/B hypothesis testing to determine the usefulness of the recommender system and further tune the considered criteria--consulting with coffee-tasting experts as well as gathering feedback from users.

A/B Testing proposal:

Suppose we narrow down our options for recommenders to two, text-based and combination, we can conduct a survey and set up our hypothesis as follows

1. Choose a coffee and get recommendations from both systems.
2. Gather tasters and have them try the three coffees: The coffee of interest, the text-based recommended coffee, and the combination-based recommender system.
3. Have the tasters rate each coffee overall on a scale from 1-100. Our parameter of interest is the difference of ratings.
4. Our null hypothesis is that the recommender systems are the same, in that tasters will rate both recommendations the same or close to the coffee of interest and the difference between the ratings is the same.

5. The alternative is that they are different; one recommender is doing a better job at recommending similar coffees and will provide better recommendations.
6. If after conducting t-testing, we reject the null hypothesis and we can confirm the alternative.

Sources

1. Reuters, March 3rd, 2018; *Americans are drinking a daily cup of coffee at the highest level in six years: survey* https://www.reuters.com/article/us-coffee-conference-survey/americans-are-drinking-a-daily-cup-of-coffee-at-the-highest-level-in-six-years-survey-idUSKCN1GT0KU
2. Interpreting Coffee Reviews: https://www.coffeereview.com/interpret-coffee/