Capstone

A Beer Recommender System

Ederlyn Tanangco

What is the goal of our beer recommender?

User consumes beer, user likes beer, user rates beer. Based on how the beer is rated following the criteria below (between 1 - 5):

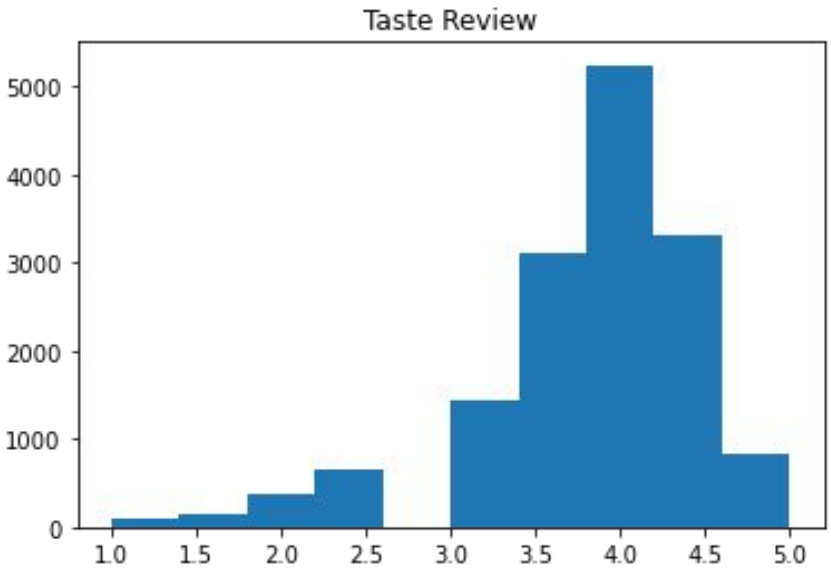
- Overall Rating
- Aroma
- Appearance
- Palate
- Taste

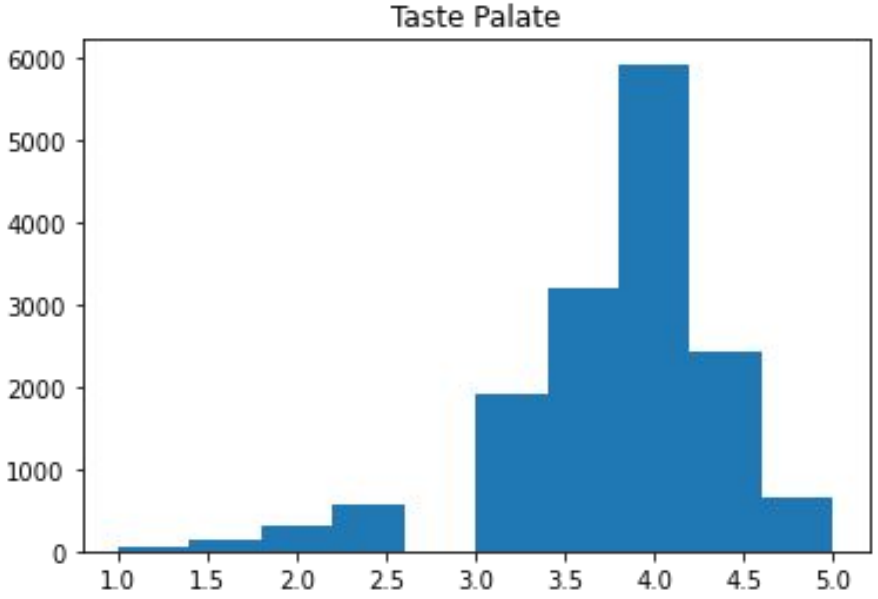
Can we recommend the top 5 similar beers based on the user's ratings?

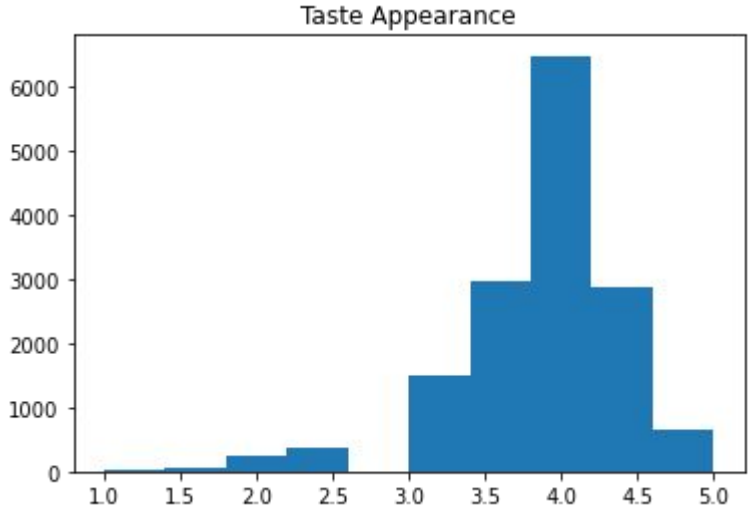
The initial process

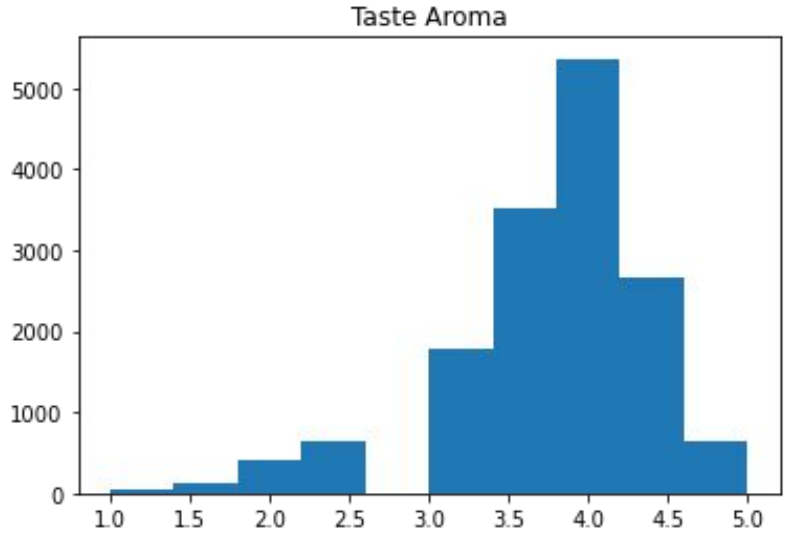
- Dataset is comprised of 1.5 million user ratings (171 MB)
  - Data is from Kaggle.com
- Dataset was too large for my personal computer & GitHub
  - Took a 1% random sample
- Cleaned dataset & completed EDA
- User ratings were between 1 - 5
  - 1 being “bad” (“I did not like”)
  - 3 being neutral
  - 5 being “good” (“I like this”)

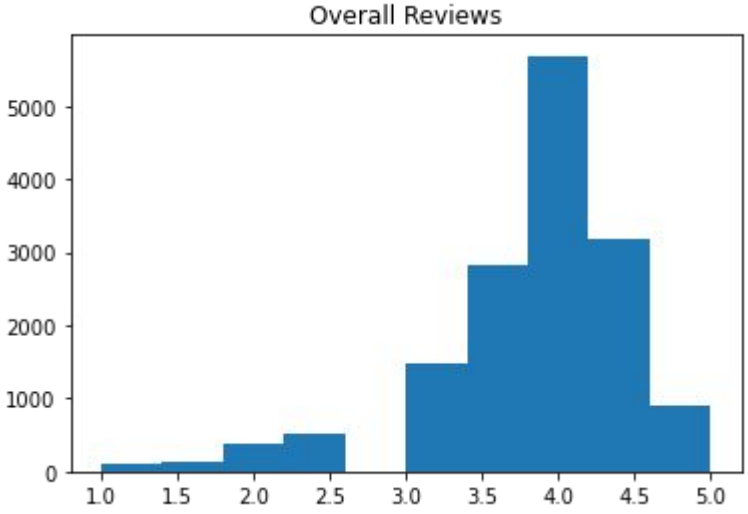
What the difference between Taste & Palate?



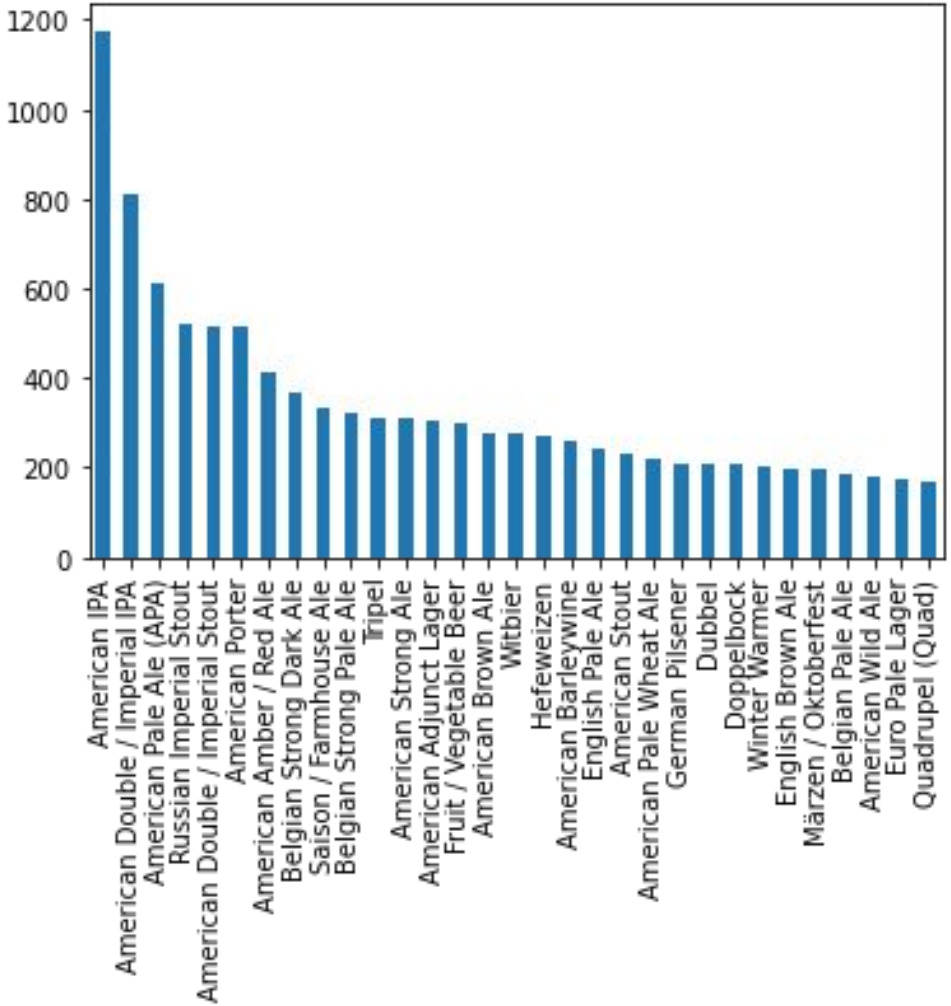



Taste Appearance
6000
5000
4000
3000
2000
1000
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Taste Aroma
5000
4000
3000
2000
1000
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Overall Reviews
5000
4000
3000
2000
1000
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

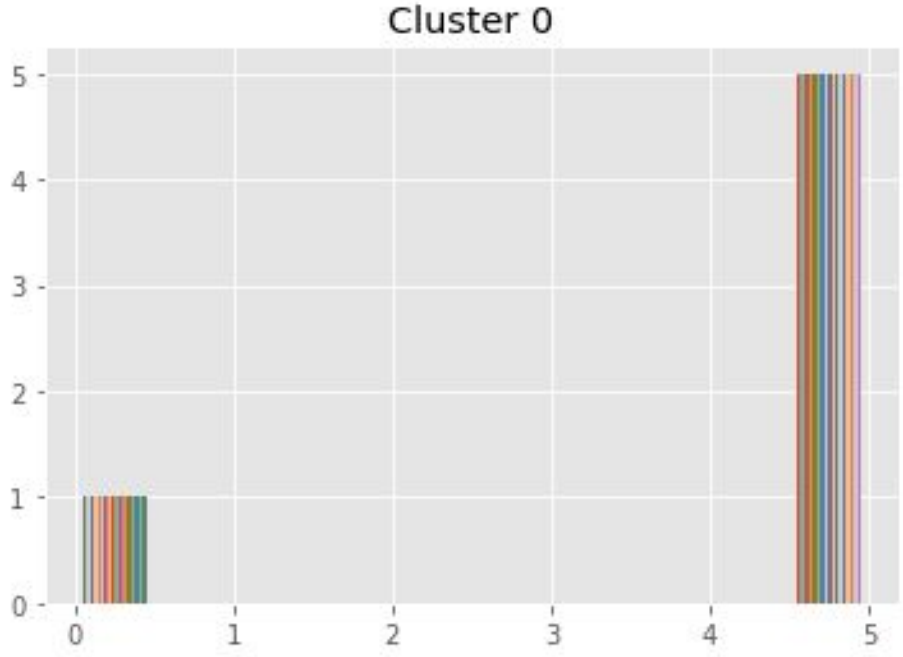
Top 30 Beers in the Dataset

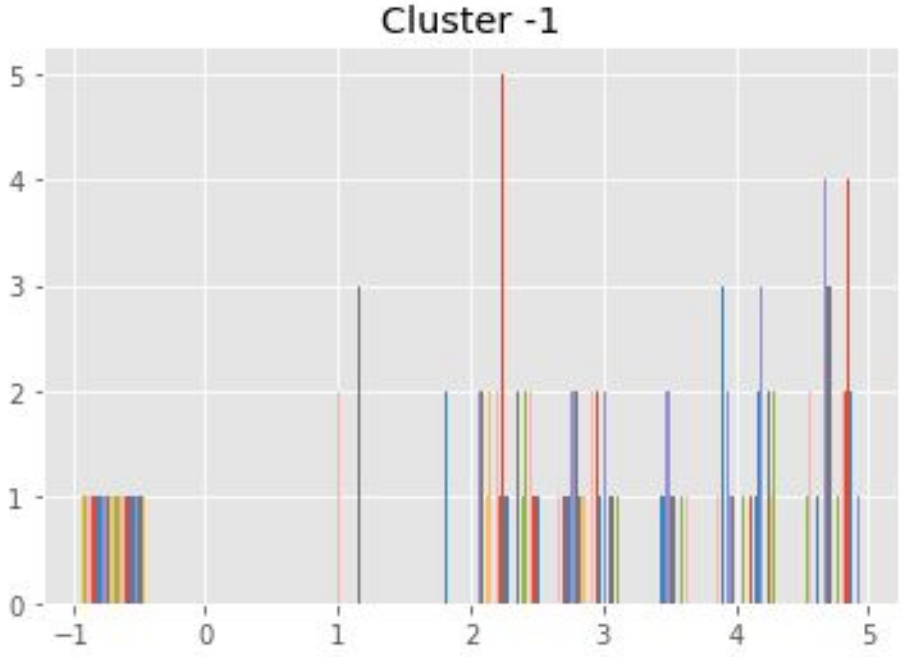


Clustering data

- KMeans
  - GridSearched
  - Silhouette score: 0.162 (bad)
- DBSCAN
  - Silhouette score: 0.834 (eps = .09, min_samples = 2)
    - Good!
    - BUT way too many clusters!
  - A lot of noise (-1)
- Tried KNN model

DBSCAN Clusters: 0 (“meh”) & -1 (Noise)





Recommender System

- Used Sci-kit Learn’s Pairwise_distances & Cosine Similarity
- Used KMeans clustered data
- Recommends top 5 closest beers based on other user ratings

Let’s try it out!

Continuing Steps

- Completion of the Flask Application for full functionality
- Scrape my own data for more current data
- Try to use the Surprise Library as the alternative for Pairwise distribution & Cosine Similarity, for the recommender
    - Read good things, so far!
- Try a few other models for better results

Pet Project: If time permits

- Create and use own dataset
  - Create my own survey
- Implement a music genre recommender to the beer recommender
  - Input: preferred music genre (“I am currently listening to jazz”)
  - Output: recommended beer style (i.e. Pale Ale, IPA, etc.)

Questions?