Beer Recommendation App Whitepaper

Dong Yi Kim | Saloni Gupta | Sooyeon Kim | Saif Gorges

Introduction **3**

App Architecture **4**
- Data Source 4
- Data ETL 5
- Deployment 5
- Machine Learning 5
  - Similarity Based 5
  - Ratings Based 5
- Front End 6
  - Homepage 6
  - Beer Recommendation - Similarity 6
  - Beer Recommendation - Ratings 7

Visual Data **8**
- Beer Reviews Score History 8
- Key Elements Evaluation 9
- Top 10 Highest Scored Beers 10
- Key Elements Boxplot 11

Conclusion **13**

Introduction

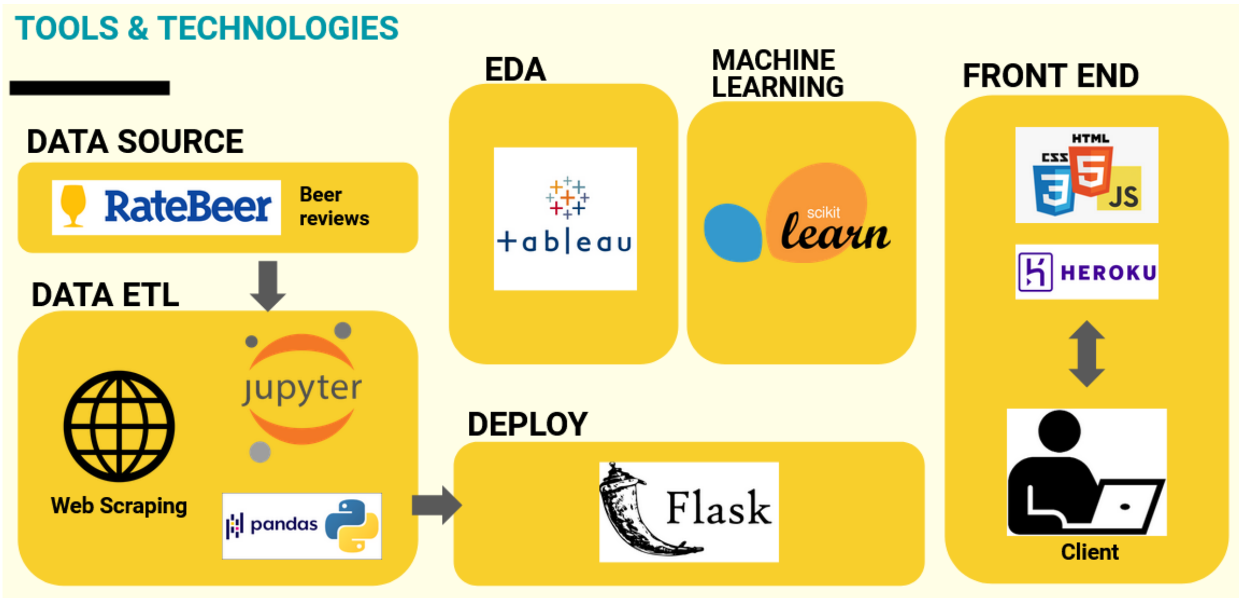
In today’s fast-paced society there are only a few things that will anchor us down to the moment and allow us to be present with those around us.

Beer.

Whether you enjoy a Lager, Ale, Stout or Porter there is one thing we can all agree on - we enjoy beer. But what makes a beer enjoyable? What are the factors that each and every one of us holds valuable in a beer? Do some of us care more about taste over texture? Or maybe a combination of both? Or perhaps some of us can’t pinpoint what it is exactly that we enjoy, however we can pinpoint specific beers that satisfy our cravings. Is it possible to consider a beer drinker’s input and recommend beers to them based on different factors such as texture, flavor and aroma? Or perhaps it is easier to recommend a beer based on other brands?

This project is set out to answer those questions. Our web-app will recommend beer to users based on their individual preferences using Machine Learning algorithms. The main problem with using ML to recommend beer is that these systems are not human, so how would they know what to recommend? We tackle this problem by sourcing all of our data from peer to peer beer reviews. In this way, all of our data is based on actual human beer sentiments and we use Machine Learning to find the best beer for a user based on their input and their peers' beer reviews.

App Architecture



In essence, the data used to rate beer was web-scraped from ratebeer.com. The data was cleaned using Python and deployed as a web-app online via Heroku.

There are two available beer recommendation modules that can be used to recommend beer to the user. The first is through beer similarities and the other is through beer ratings. Once the user selects a recommendation module, they will be directed to a webpage where they will need to enter specific parameters of beer choices and submit. Once submitted, the input will be fed into a machine learning module which uses collaborative filtering to find the three best beers for the user based on their input. The recommendations will be fed back into the front-end where the user will see their three recommended beers.

Data Source

Ratebeer.com is a widely used and trusted source for gathering beer information based on user ratings. These ratings are split into multiple factors such as aroma, texture and taste. All of the data gathered for this project was sourced from ratebeer.com. The project samples 100 beers from 17000 users, 91776 reviews and spans a timeframe of 20 years. We chose ratebeer.com because of its wide variety of beers ratings and crowd-sourced reviews. The data was extracted via web-scraping and stored in CSV format.

Data ETL

Once the data was extracted from ratebeer.com, it was stored in a CSV file and cleaned using Pandas in Python. Once the data was cleaned, it was saved as a CSV and stored back into the repository.

Deployment

Flask was chosen as the middleware tool to serve the HTML and back-end of the app. User input from the HTML webpage is passed into the machine learning modules via Flask and returned back to the HTML front-end via Flask.

Machine Learning

The machine learning method chosen to determine beer recommendations is ‘Collaborative Filtering’.There are two models available for determining beer recommendations and both are based on the chosen recommendation of collaborative filtering.

Similarity Based

The similarity based method focuses on recommending beer based on one beer of choice and preferred factors such as aroma, taste or mouthfeel (texture).

Ratings Based

The similarity based method focuses on recommending beer based on beer ratings. The user will select their top three favorite beers and rank them on a scale of 1 to 5.

Nearest-neighbor collaboration filtering is classified into **user-based** and **item-based** , among which, we have implemented item-based collaborative filtering as it offers better recommendation accuracy.

Item based collaborative filtering is preferred because of the following reasons:
1) Very popular beers are often viewed and appreciated by most people, regardless of taste. Also, individual tastes of users are too diverse which can skew the dataset.
2) However, in most cases, it is difficult to compare similarities with others because the number of users leaving a review or rating the beer are limited.

Therefore, we applied the nearest neighbor collaborative filtering as an item-based algorithm.

Item-based Nearest Neighbor-based Collaborative Filtering

Regardless of the attributes of the item, the user's preferences are evaluated. An algorithm that is the basis for recommending items with similar scales.

Calculation of Similarity between beers

The similarity based method focuses on recommending beer based on one beer of choice and preferred factors such as aroma, taste or mouthfeel (texture).
The beer data collected includes beers with at least 1 review from the users. We build a dataset of only beer and users with at least 10 reviews.

```
# Finding cosine similarity from item-user matrix
item_sim = cosine_similarity(ratings_matrix_T, ratings_matrix_T)

# Convert the beer name to a DataFrame by mapping the beer name to the NumPy matrix returned by cosine_similarity()
item_sim_df = pd.DataFrame(data=item_sim, index=ratings_matrix.columns,
                           columns=ratings_matrix.columns)

print(item_sim_df.shape)
item_sim_df.head(3)
```

```
(82, 82)
```

beer_name	Hoegaarden Rosée	Alexander Keith's India Pale Ale	Baltika 7 Eksportnoe (Export)	Bavaria 8.6 (Original)	Beck's	Bellwoods Bring Out Your Dead	Bellwoods Jelly King (Raspberry and Blackberry)	Berthold Keller Premium Lager	Boxer Ice	BrewDog / Weihenstephan India Pale Weizen	Bud Light	Bud Light Apple
beer_name												
Hoegaarden Rosée	1.000000	0.141732	0.385892	0.303081	0.346399	0.157306	0.033456	0.170984	0.031929	0.284598	0.208556	0.093603
Alexander Keith's India Pale Ale	0.141732	1.000000	0.247895	0.169166	0.318291	0.182993	0.130136	0.249939	0.134634	0.061400	0.313484	0.129463
Baltika 7 Eksportnoe (Export)	0.385892	0.247895	1.000000	0.310713	0.452780	0.168058	0.049181	0.229820	0.084728	0.233912	0.322482	0.062611

3 rows × 82 columns

```
# Extracting only 5 beers with a similarity to Baltika 7 Eksportnoe (Export)
item_sim_df['Baltika 7 Eksportnoe (Export)'].sort_values(ascending=False)[:5]
```

```
beer_name
Baltika 7 Eksportnoe (Export)     1.000000
Erdinger Weissbier                0.484676
Tiger Beer                        0.474786
Kirin Ichiban                     0.466757
Weihenstephaner Hefeweissbier     0.460691
Name: Baltika 7 Eksportnoe (Export), dtype: float64
```

For the beer ‘Baltika 7 Eksportnoe (Export)’, the illustrated beer ‘Erdinger Weissbier’ has the highest similarity, followed by ‘Tiger Beer’ and ‘Kirin Ichiban’.

Personalized Beer Recommendations

It is a method that recommends beers with high predicted ratings after calculating the predicted ratings of all other beers based on item similarity and rating data of previously rated beers for the beers that have not yet been tasted by the individuals.

$$\hat{R}_{u,i} = \sum N\left(S_{i,N} * R_{u,N}\right) / \sum N\left(|S_{i,N}|\right)$$

Ru, i: Personalized predictive rating value of user u, item i
Si, N: similarity vector of Top-N items that have the highest similarity with item i
Ru,N: Actual ratings vector about Top-N items that have the highest similarity with the user u’s item i

This recommendation approach involved the following steps:

1) We calculated similarity b/w all the beers in our dataset using cosine similarity function to extract 20 beers with high similarity.
2) Next , we have used predictive evaluation index i.e. MSE (Mean Squared Error) to measure error b/w predicted and actual rating inputted by the user.
3) Finally, we reduced the MSE value by applying a function which applies similarity vector only to beers having most similar similarity & calculated rating to identify top beers that are highly similar to user's preference.

Based on these steps, the algorithm calculated predictive scores as shown below:

beer_name	Prediction_Score
Stella Artois	2.378123
Hoegaarden Grand Cru	2.230628
Krombacher Pils	2.098706

Front End

The web-app uses HTML, CSS, Javascript and Bootstraps to function. The pages are served through Flask and deployed via Heroku. The following are the pages accessible through the web-app.

Homepage

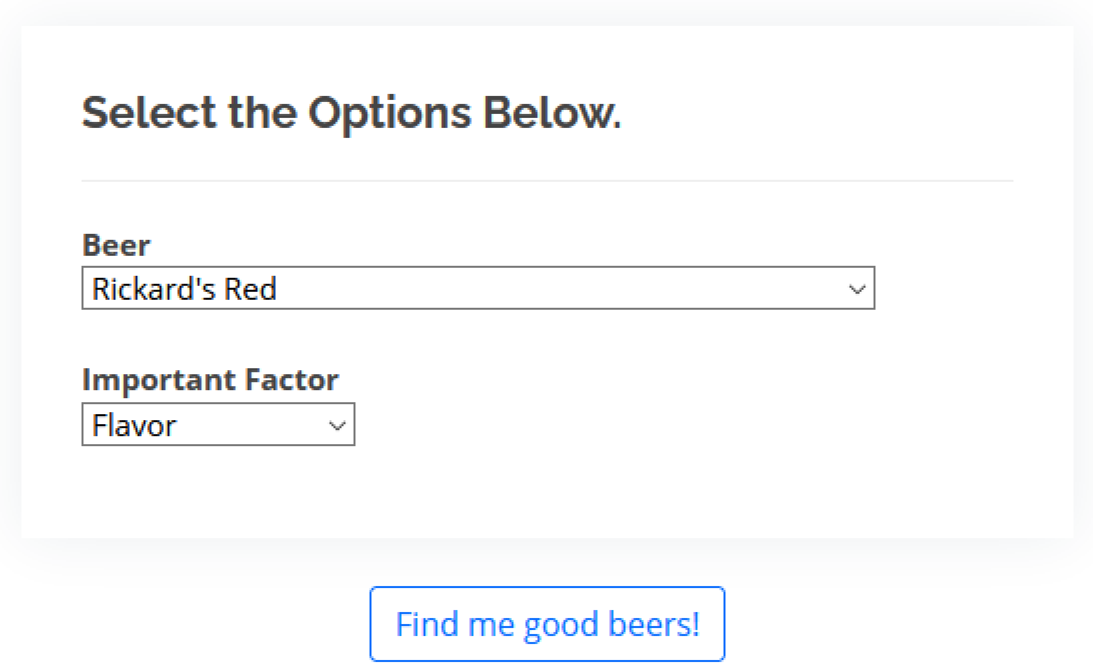
The homepage is the main page that the user will land on when using the link: https://beer-recommendation-app.herokuapp.com/

Within the homepage there is an 'About' section which describes the project's goals, as well as links to the Github repository and the University of Toronto website. The homepage also includes statistics from the beer data included within the app. These tables are embedded from Tableau and served onto the webpage.

Below the statistical data are two links that lead to the two modules for beer recommendation as well as links to each team members Linkedin and Github repositories.

Beer Recommendation - Similarity

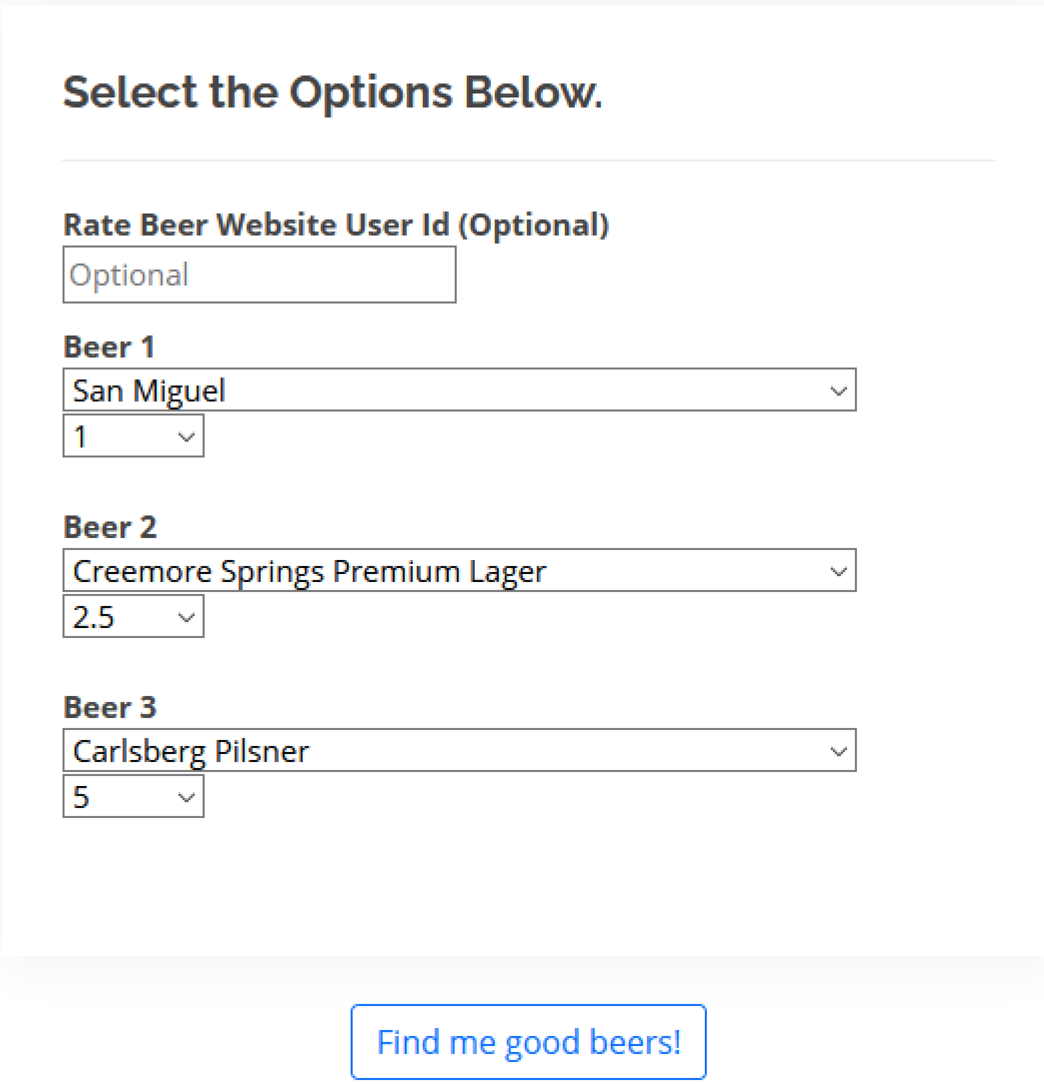
This is the main page where the user can select their beer of choice and the factor which matters to them most about their selected beer. The link can be found here: https://beer-recommendation-app.herokuapp.com/recommend_a
Both the beer and beer factor and chosen via dropdown menu.



Once the user selects ‘Find me good beers!’ the input will feed into the Machine Learning module and return three beer recommendations with images on the current webpage.

Beer Recommendation - Ratings

This is the main page where the user can select three beers of choice and rate them on a scale of 1 to 5. The link can be found here:
https://beer-recommendation-app.herokuapp.com/recommend_b
Both the beer and ratings are chosen via dropdown menu.

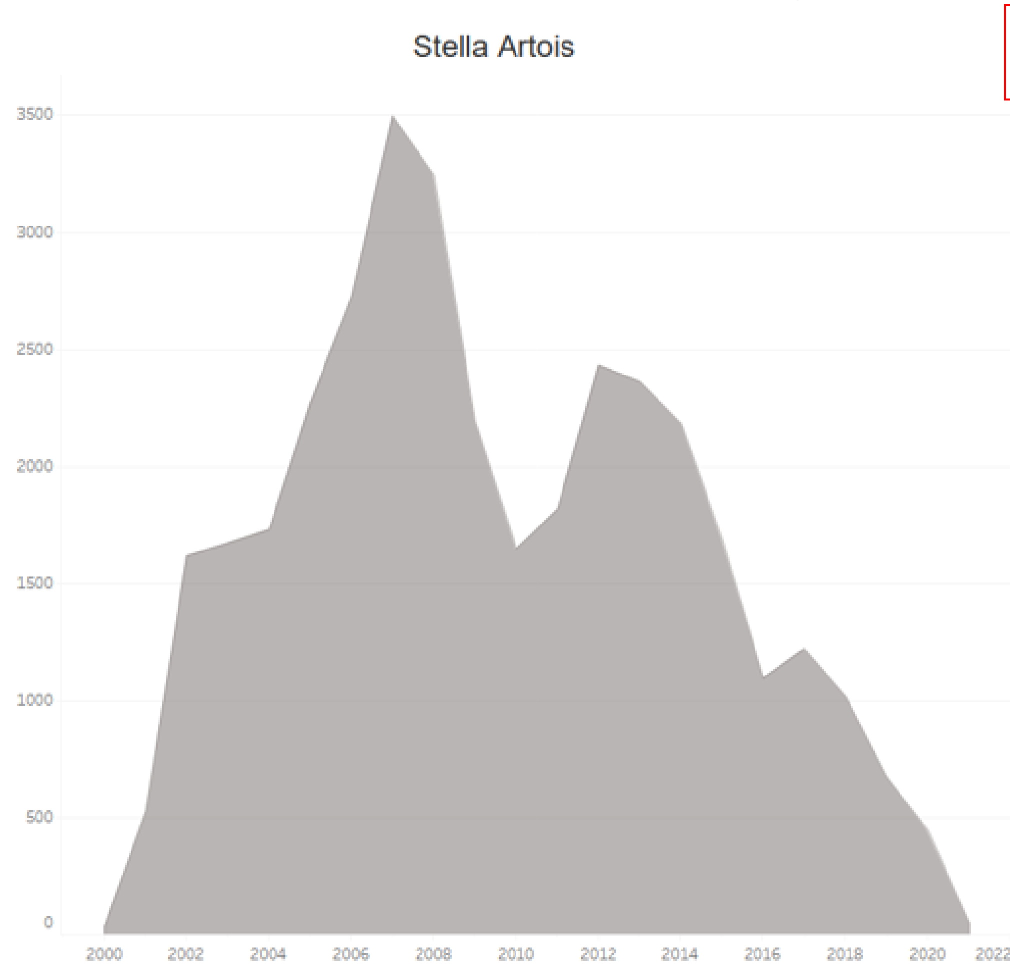


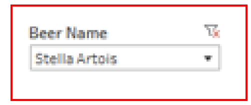
Once the user selects ‘Find me good beers!’ the input will feed into the Machine Learning module and return three beer recommendations with images on the current webpage.

Visual Data

Beer Reviews Score History

The ‘Beer Reviews Score History’ graph can be accessed from the main homepage of the web-app and is the first tab under the ‘Exploratory Analysis’ section. This analysis represents the score of a chosen beer over time. We have included 20 years of data and our dataset ranges from the years 2000 to 2020. The data is based on user ratings of the selected beer over 20 years.

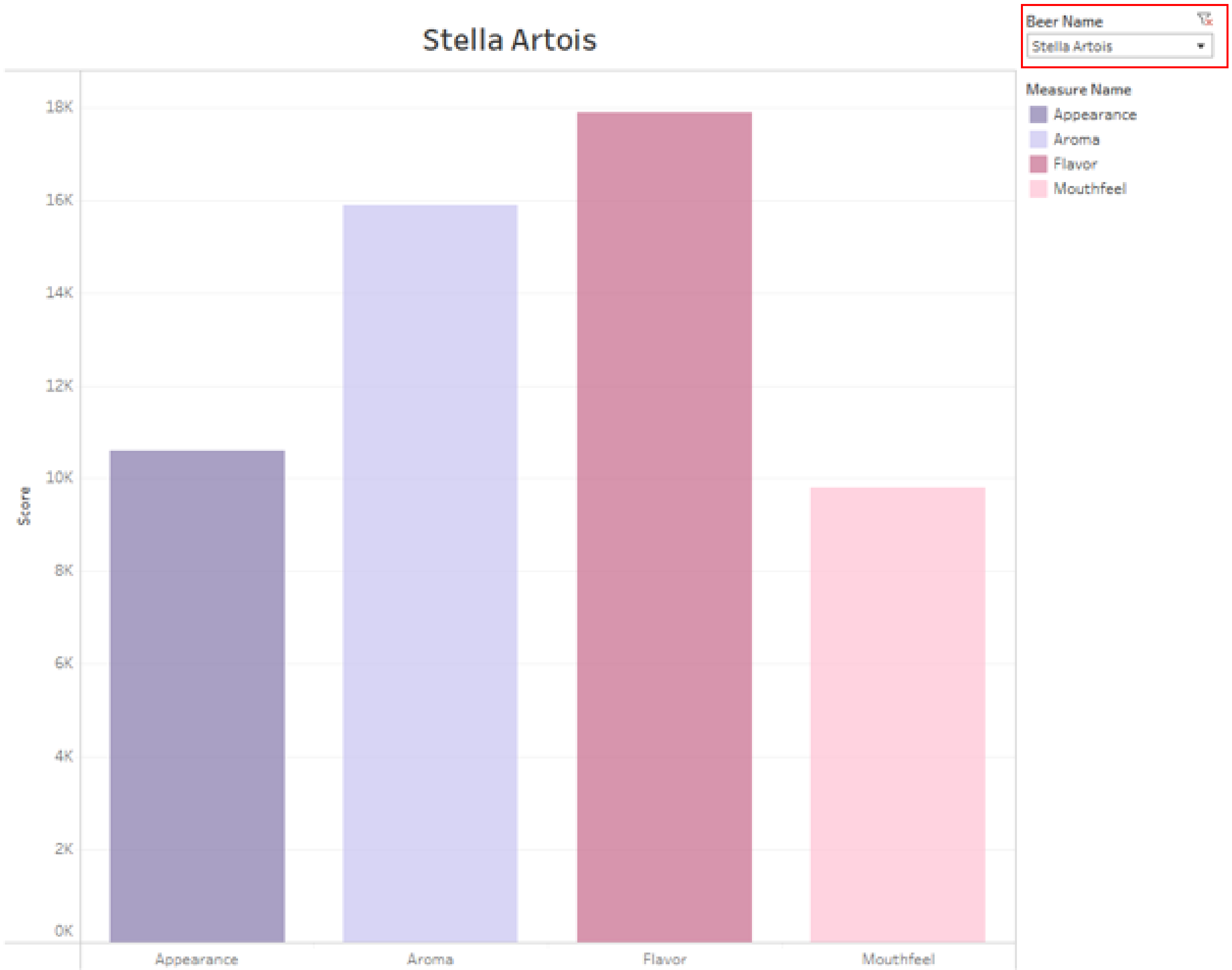




In the example above, we can deduce that the overall rating of ‘Stella Artois’ has been in a slow decline over the past 20 years, however there was peak interest in the beer during 2006 and 2007. This analysis can be useful in terms of comparing user sentiment vs marketing campaigns around the same times as the peaks and valleys of this analysis. A different beer can be chosen from the red highlighted box below and a similar report will be generated.

Key Elements Evaluation

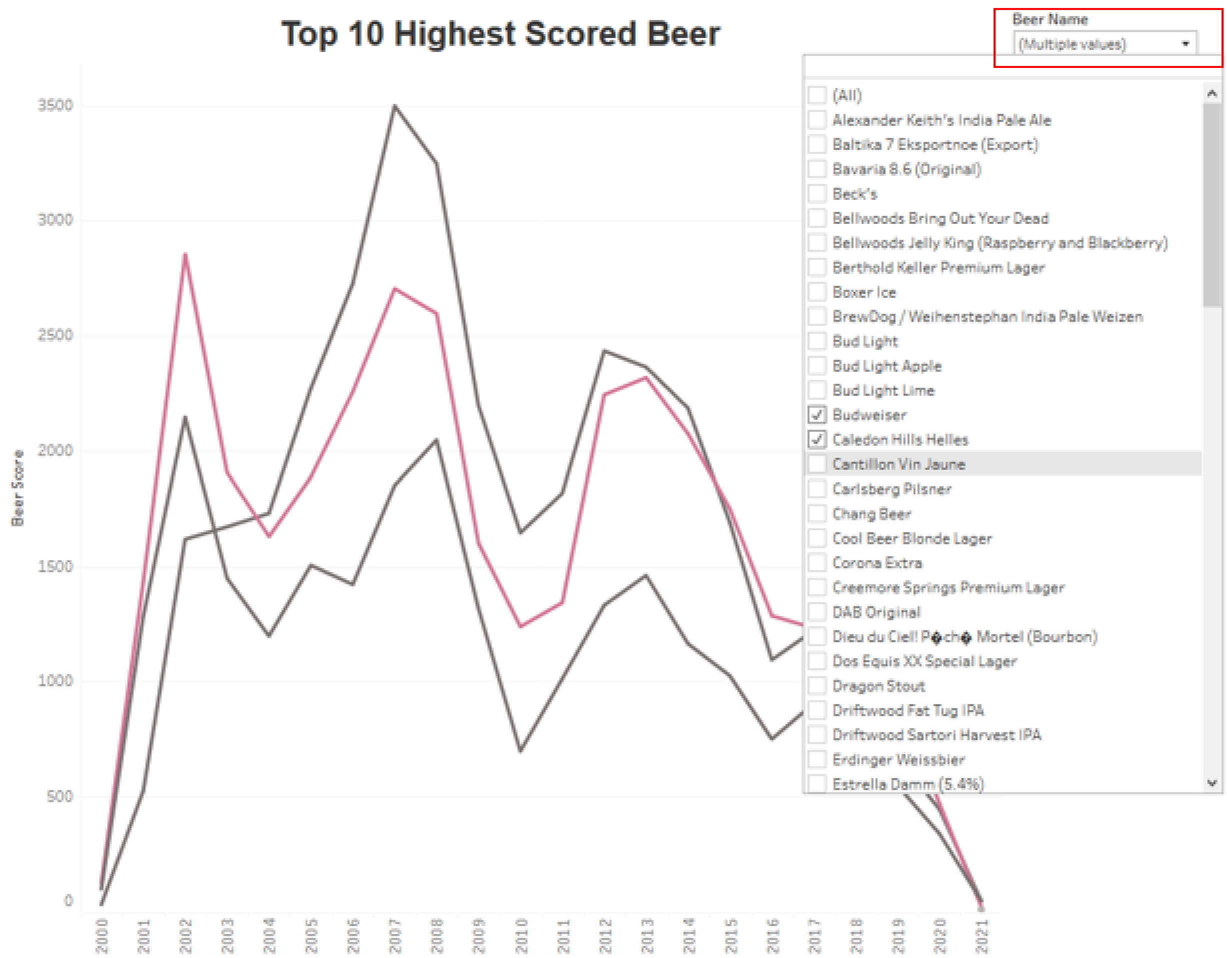
The ‘Key Elements Evaluation’ graph can be accessed from the main homepage of the web-app and is the second tab under the ‘Exploratory Analysis’ section. This analysis represents factors that beer drinkers find are the most attractive about the selected beer. We have included 20 years of data and our dataset ranges from the years 2000 to 2020. The data is based on user ratings of the selected beer over 20 years.



In the example above, we can conclude that the top two reasons why beer drinkers enjoy ‘Stella Artois’ is because of the flavor and aroma. This data can be useful in terms of improving the sales of ‘Stella’ by tackling some of the features in which beer drinkers do not find ‘Stella’ as attractive such as appearance and mouthfeel (texture). A different beer can be chosen from the red highlighted box below and a similar report will be generated.

Top 10 Highest Scored Beers

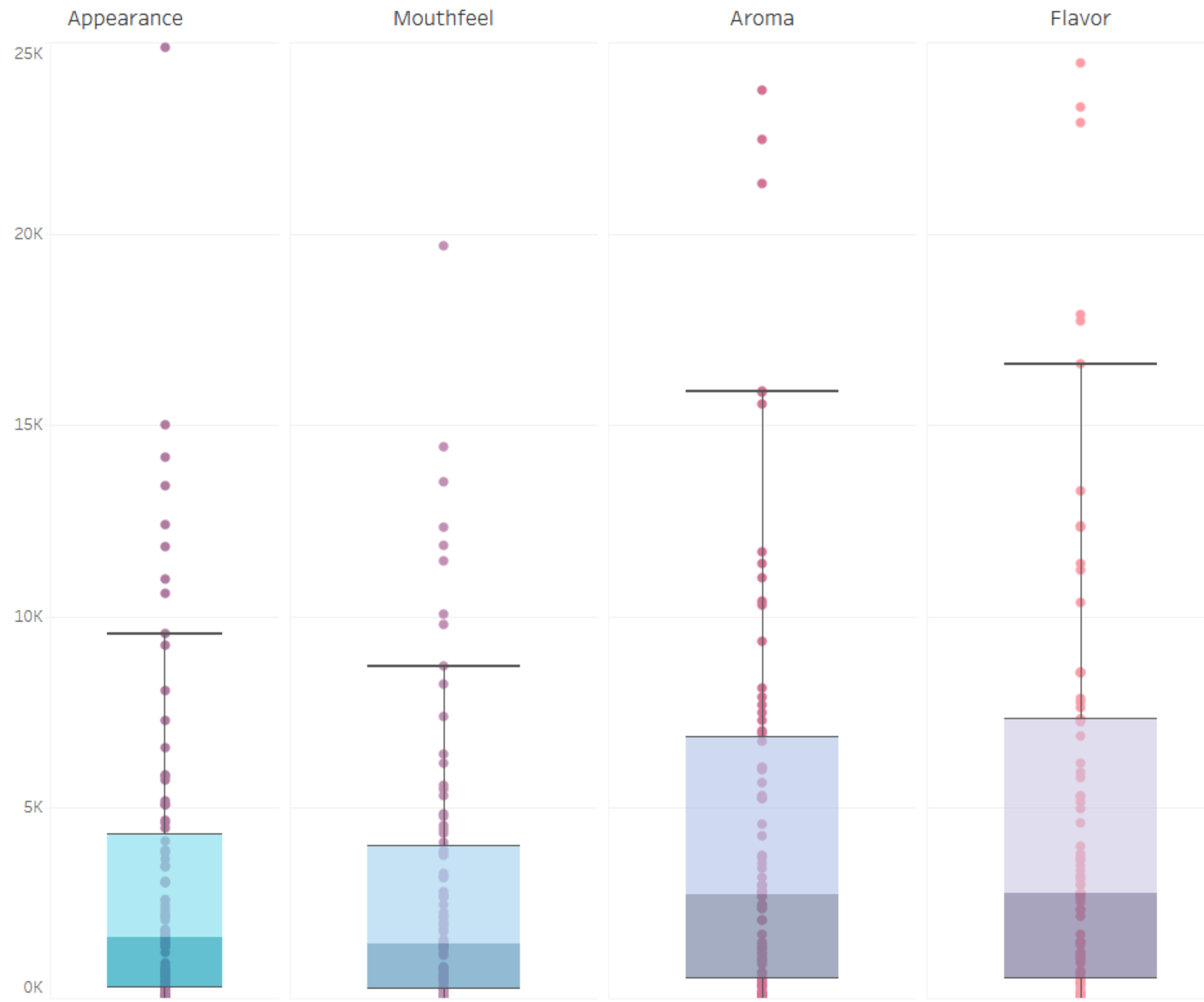
The ‘Top 10 Highest Scored Beers’ graph can be accessed from the main homepage of the web-app and is the third tab under the ‘Exploratory Analysis’ section.This analysis represents the score of a chosen beer over time. It is similar to the ‘Beer Reviews Score History’ however we have included the option to choose other beers from the dropdown menu in order to compare sentiment of different beers over time. We have included 20 years of data and our dataset ranges from the years 2000 to 2020. The data is based on user ratings of the selected beer over 20 years.



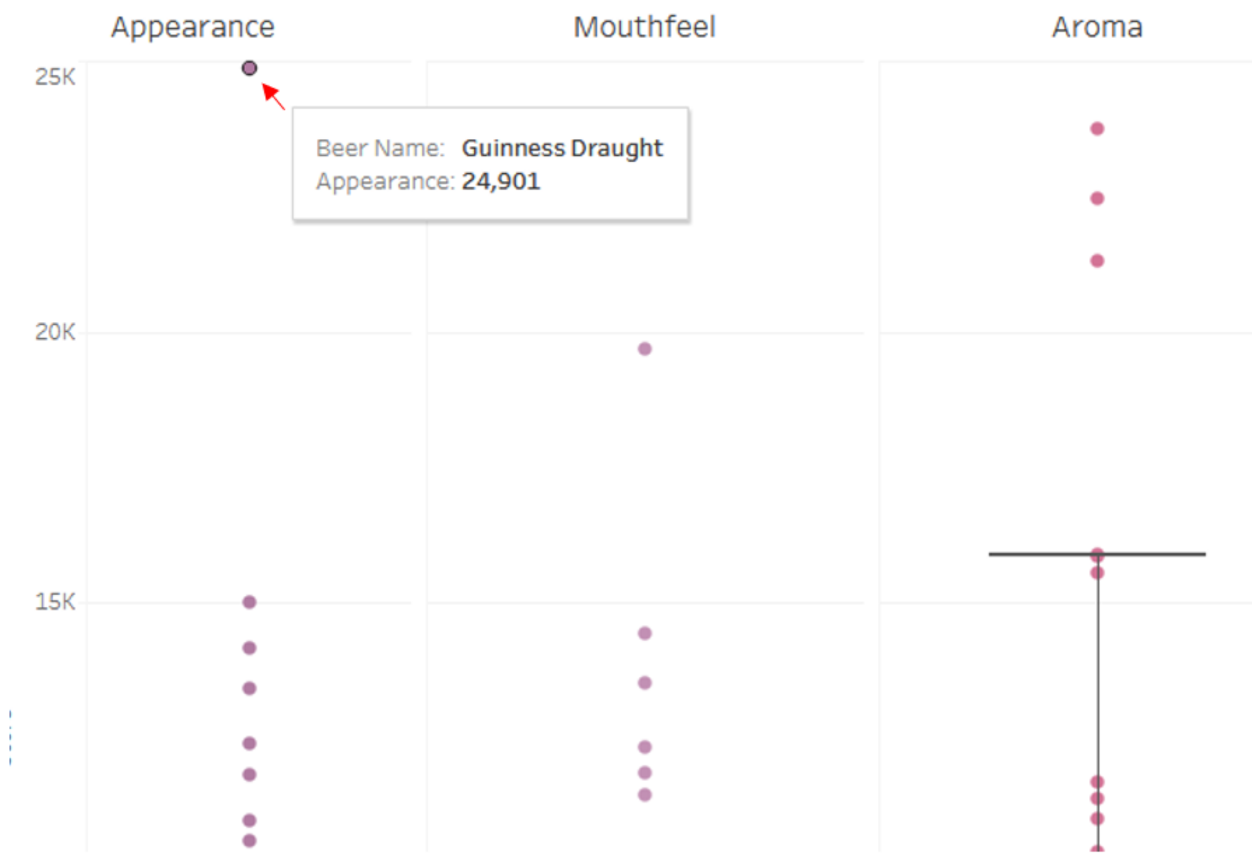
In the example above, we can compare how different beers of similar nature have scored over time. This data can be used to determine whether or not there is a trend of certain types of beers (ale, stout, lager etc..) over time. This can help with targeting ad campaigns on specific types of beers or improving on certain features of the beer using the ‘Key Elements Evaluation’ analysis from above.

Key Elements Boxplot

The ‘Key Elements Boxplot’ graph can be accessed from the main homepage of the web-app and is the fourth tab under the ‘Exploratory Analysis’ section.This analysis represents where each beer stands in terms of the appearance, mouthfeel (texture), aroma and flavor category. The data samples 100 beers from 17000 users, 91776 reviews and spans a timeframe of 20 years.



The above data can be used to determine where each beer stands out in a category. For example in the figure below, when we hover our mouse over the farthest outlier in the ‘Appearance’ category we see that this dot represents ‘Guinness Draught’.



This tells us that out of all 100 beers that were sampled, Guinness Draught scores the highest in the ‘Appearance’ category. This type of information can be used to understand what makes a beer successful in the eyes of a beer drinker. Knowing that Guinness Draught scores high on the list may compel a brewer to change the appearance of their beer to that more similar to Guiness and potentially attract more customers and increase sales.

Conclusion

In conclusion, we believe it is possible to source multiple factors such as flavor, aroma and mouthfeel (texture) in order to recommend beer choices to individual users based on their preferences. Using collaborative filtering techniques and sourcing multiple reviews from peers it is possible to create unbiased recommendation algorithms that are still based on human opinion.

The data that is collected from ratebeer.com is not only essential to helping the algorithm determine what the best beer to recommend is, but it is also useful to determine user sentiment on beers over time. Our exploratory analysis allows us to see how different beer factors (aroma, texture etc..) are rated by users which in turn allows us to understand what makes certain beers successful and which do not.

This project can be expanded on in the future to include feedback loops for the algorithm to help it determine whether the beers it recommended were actually enjoyed by the individual or not. This would allow the algorithm to fine-tune itself in order to provide a more refined recommendation. Other key future developments of this project include expanding the sample size of the dataset to include more beers and to evenly distribute the beer categories within the dataset. This would allow for a less biased and more accurate recommendation.