Predictive Pandas presents

BeerNext

EST. 2020

Michelle Hocking
Ruohong Yuan
Eamonn McCallum
Yann Chye
Pooja Mallard
Kat Shamai

Now that Victorians can “get on the beers” again, what to drink?

You can go with the same old, same old but you’re probably tired of that since that’s probably all you’ve been drinking for the past 6 months

We thought it would be great to celebrate the world of craft beers and get beer lovers to try something different, and find a new favorite

Using review data and data on breweries, we set out on a virtual brewery data crawl to find good datasets, and ways to teach a model how to appreciate craft beer, and recommend a good one

Data points

From our search, we identified 3 data points which provided us with the information needed

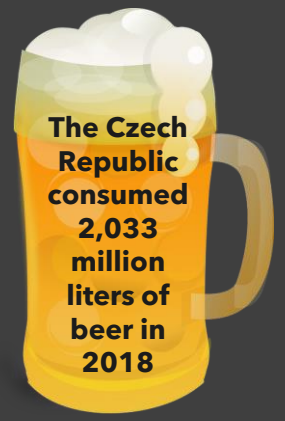
Data point 1 – Dataworld

Data point 2 – BeerAdvocate Website

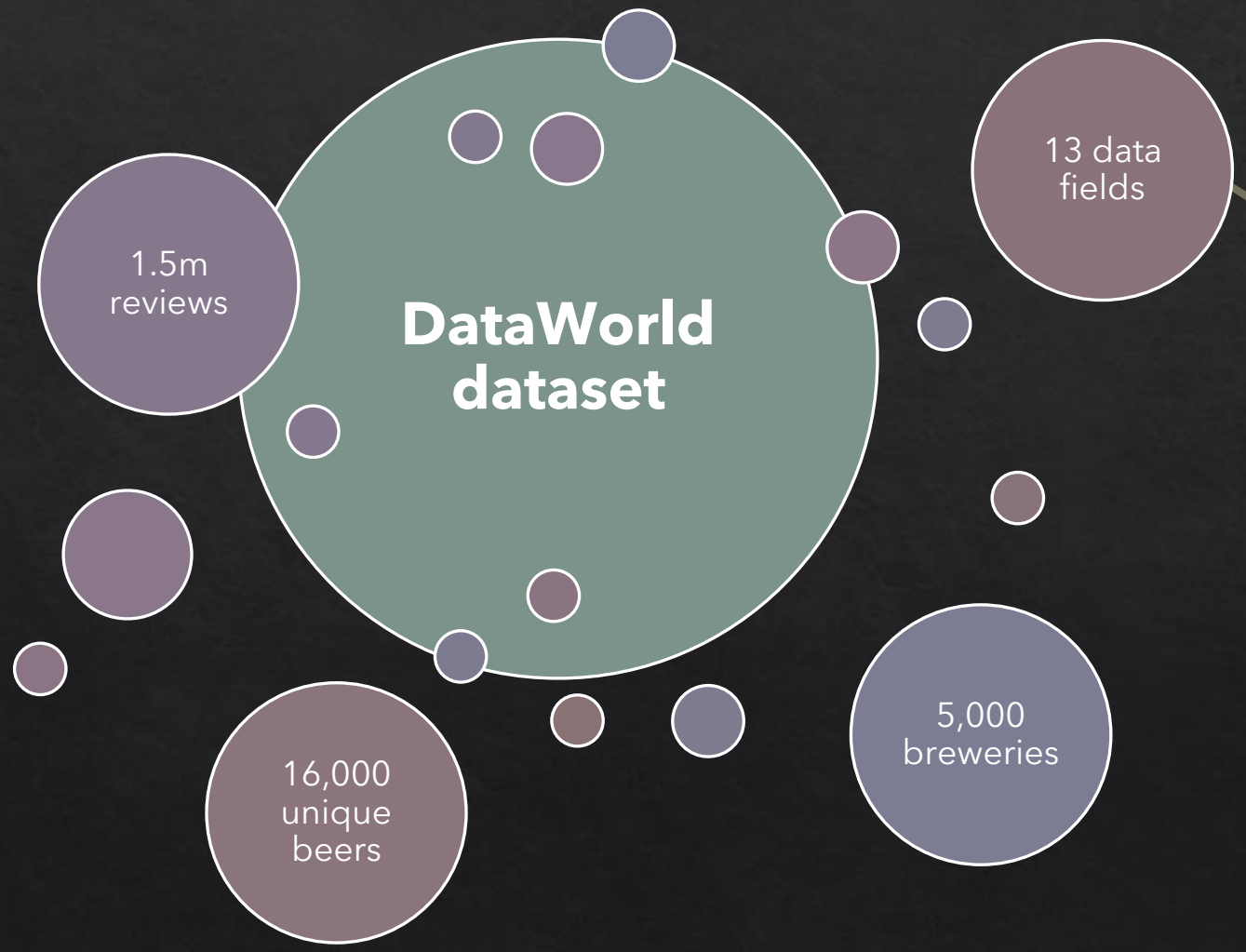
Data point 3 – Google API

Data Munging

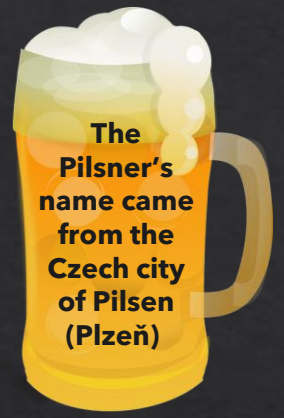
- Data extraction
- Data transformation
  - Exploratory analysis
  - Data aggregation
- Adding features
- Machine learning
- Training the model
- Cluster optimization
- Filtering and selection
- Web app
- Leaflet visualisation



Data Extraction



- 2 identifiers each with a descriptor
- 5 review scores for different features
- 1 timestamp
- 1 user profile name



Data Transformation

Clean dataset?

Data Profiling

Initial inspection and basic statistical analysis

Validate outliers (high alcohol content beers)

Reduce the number of beer styles (add a feature)

Investigate the high correlation of review score categories

Inspect reviewer score distribution and frequency

Aggregation: Group by unique beer
Average review scores
Drop poorly rated beers i.e. >3

Do we need more features?

Adding Features

Scrape for availability (1 feature)

Scrape for descriptive words (173 features)

Scrape for brewery locations (3 features)

Remove retired beers

Remove breweries with no location

Remove closed breweries

Dataset 13.2k records

Consideration of Heroku constraints

Reduce dataset?

Use of a database?

Load static csv file

Machine Learning

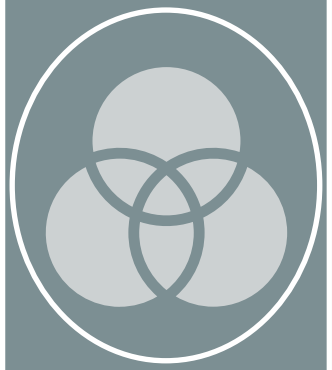
- We wanted our model to have the ability to recommend beers outside of rigid styles like “Ale” or “Stout” but aligned to your taste preferences
- To answer this, we used K-means Clustering to generate discrete groups to enrich our dataset independent of beer style
- Our model optimised at 22 clusters, confirmed with a Silhouette Score of 0.61
- With our new information fed back into our data we apply a cosine similarity analysis based on user input to predict five different beers

Training the model

2 data input approaches were tested

- All features converted into dense vector
- Limited features converted into sparse vector

Limited features was selected based on arbitrary no. of clusters and comparison of Silhouette score



Limited features broken up in types

- No. of reviews
- Scores
- Type
- Flavor
- Origin

Arbitrary no. of clusters were used, and combinations of features were compared using Silhouette score

Final combination of features compared by cosine similarity

- No. of reviews
- Score
- Type
- Flavor
- Origin

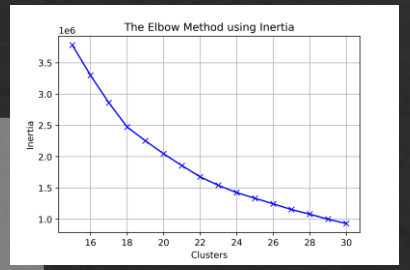
Excluded from kMeans model

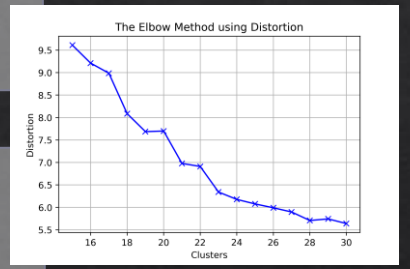
- Flavor
- Origin

Cluster Optimisation

Cluster optimisation was done using range of clusters between 15 and 30.

We validated results with elbow method of both distortions and inertia, as well as Silhouette.



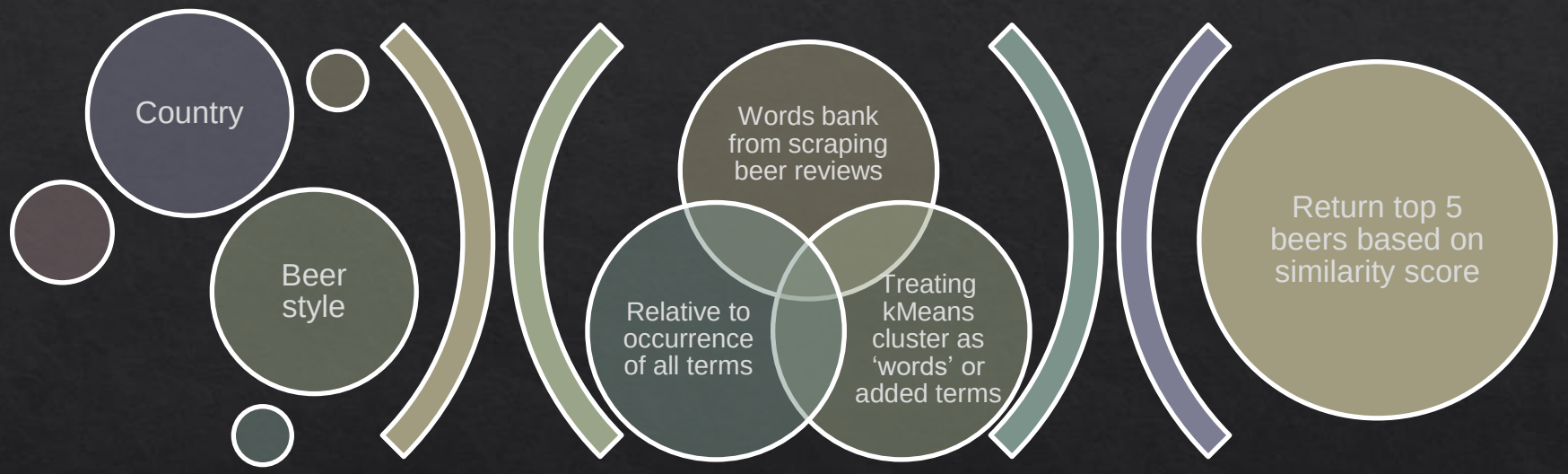


Final decision was 22 clusters that yielded Silhouette score of 0.61 with optimum distortion and inertia scores.

We note there was a negative correlation in elbow scores as numbers of clusters increased, so we didn't investigate any number of clusters above our tested range.

We also note the distribution of clusters was not consistent. It is our assumption that is due to the nature of our dataset.

Filtering and Selection



Include / Exclude beers based on user selection

Collaborative filtering using term frequency-inverse document frequency (TF-IDF)

Return results

Web App

Design

- Sketch of what the web app could look like
- Design of look and feel using d3, Javascript, and HTML
- Design of Flask back end
- Creation of BeerNext logo
- Deployment to Heroku

Customization for recommendations

- A key feature of the web app is the ability to customize recommendation features
- Use of sliders
- Use of autocomplete for free text fields to limit variability of user input

Recommendations

- Recommendations from machine learning model retrieved from Flask
- Leaflet map visualization is populated with brewery information
- Recommendations provided to users

Leaflet Visualisation

Idea

We wanted to add a map feature for users who want to visit the brewery or purchase the beers recommended

Visualisation

To create the map visualisation, we used Leaflet and the Google Map latitude and longitude data for each brewery

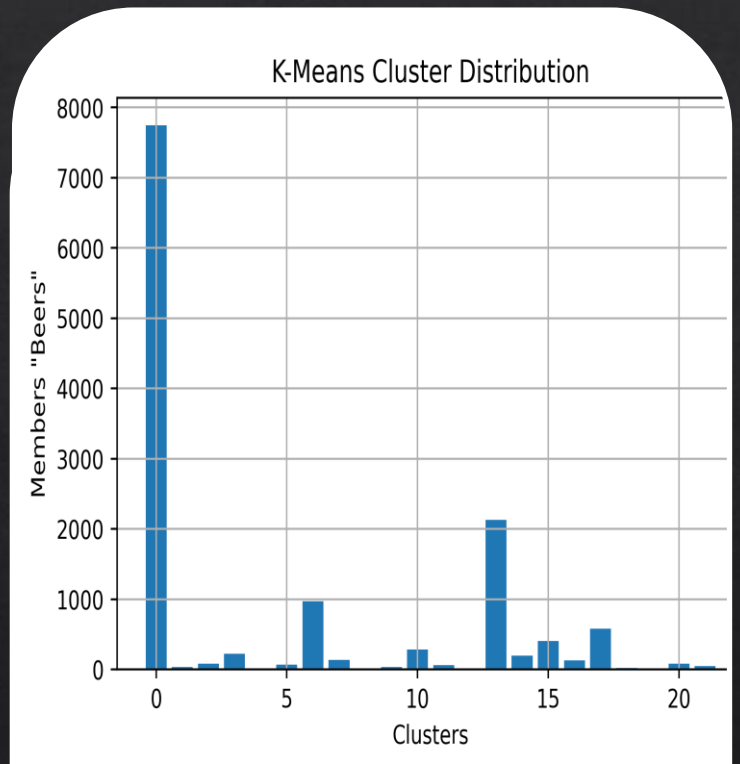
Plotting

A marker was created for each brewery (with a little beer stein of course!)

Information

Information popup boxes were created to show users the brewery name and address information

Interesting Observations



Uneven distribution of cluster is a characteristic of our data

Some styles are substantially better represented and closely related

For example, ales and IPA make up a 46% in flavor terms as they are related but you may not know IPA stands for India Pale Ale

- Ales consisted of 36%
- IPAs consisted of 10%

Of the total styles of beer (104), a vast number of were 'special' or 'minority' style beer

We decided to make limit of 10 major style and consider any other styles to be "exotic" so that the user input options would be simpler

While this label makes up 20% of our data, we preferred this approach to adding another 94 with minimal representation

We believe this to be the primary driver of uneven distributions of K-mean predicted classes

Challenges

- Finding the "right" features to improve predictability
- Getting them once you find them - BIG data means lots of hours scraping and crunching data together
- Translating the dataset into a concept and then sticking to it – danger of losing sight of what the model should do as the dataset evolves
- Landing on a model type
- Being able to 'validate' the accuracy (visualising, PCA, silhouette analysis)
- Filtering the model results from each cluster
- The good old GitHub version control / conflicts

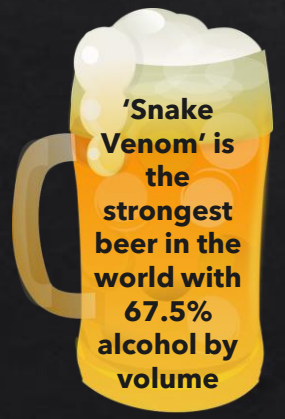
More Time

Compare different unsupervised model such as DBscan , or NLP and word2vec to handle text data in a different way

Spend time reviewing a much larger number of clusters, e.g. > 100, which may provide better results

With the enriched data, compare another supervised prediction method. We tried a Random Forest, however, due to the nature of our dataset, the amount of ram required to process this exceeded our capacity on both machines, and on Colab-Pro i,e, > 35 gig

A more effective way for users to filter / search features using keywords; auto-complete can be hit and miss and relies on users entering data in a particular way



Future Features

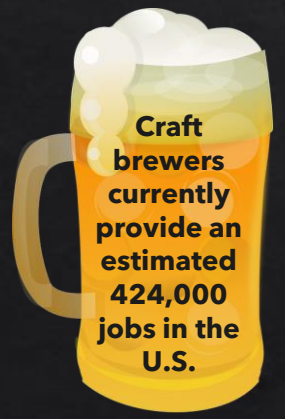
Include information scraped from brewery websites or snippets from reviews (e.g. top review and / or lowest review)

Provide reviewing capabilities to grow the dataset for machine learning

Include photo of recommended beer

Seek feedback from users on our recommendation to enable further refinement of the machine learning model

Demo


Craft brewers currently provide an estimated 424,000 jobs in the U.S.

Any questions?