Data Engineering — Intermediate Report 3

Daan Brugmans (s1080742)
Radboud University
Nijmegen, Netherlands
daan.brugmans@ru.nl

1 DATA & PROJECT PROPOSAL

For the Data Engineering project, I would like to build a pipeline that serves data to data scientists working on a beer recommendation system. The pipeline should collect from multiple sources that serve data(sets) of beer and their attributes. Examples of attributes that should be served to the data scientists are name, brewery, alcohol by volume (ABV), international bitterness unit (IBU), category/style, textual description, flavor profile, and personal ratings. The pipeline should serve two sets of data to the data scientists: a big general dataset containing beer data from varying publicly available sources that can (and should) be updated, and a small dataset of an end user of the beer recommendation system that includes personal ratings of beers the end user has had before. These datasets will be served tabular. The data scientists can then use the ratings of the end user and the data of both datasets to build a system that can recommend beers from the big dataset to the end user's non-specified preferences, which can be extracted from the end user's data using machine learning. A realistic use case for such a recommendation system could exist for social media platforms revolving around beer and online retailers of beer that can use the system to increase revenue and beer sales.

For the small dataset of an end user, I will provide my own data of beer ratings that I have collected on Untappd. Untappd [4] is a social medium where users rate beers they try and share their ratings with friends by registering their rating on the medium ("check-in"). I participate in Untappd and have collected a dataset of these check-ins with ratings. I can access an export of my check-in data per GDPR request. This export contains information about a beer's name, category/style, brewery, check-in location, purchasing location, flavor profile experienced by the end user, rating, and timestamp. The data will be provided in CSV format and contain almost 400 check-ins, slightly over 300 of which being unique, so while I do not expect for missing data to be a major issue, duplicates will be more prominent.

For the big dataset containing beers that can be recommended to an end user, I have found multiple data sources of beer attributes. I want to aggregate the data from these sources into a single database of beers. Because my data sources serve data in varying formats, I expect that the aggregating of my data into a single collection will be a major challenge of the project. For example, I will have to make sure that all data uses the same formats and standards, that there are no duplicates due to beers being present in multiple sources, and that there will likely be missing data due to some data sources not storing certain attributes.

I want to make use of the following data sources:

- OpenBeerDB [2] is an archive of beer data. Although it currently serves an older, static dataset, OpenBeerDB is planning on updating this dataset in the future, possibly including regular updates to the data. The OpenBeerDB data is served as a set of `.sql` files that will create tables for beer, breweries, categories, and styles (finer granularity categories), and insert them with data. The resulting SQL database contains most attributes I would want to serve to the data scientists, with the exception of flavor profile.
- Rémi [3] serves a static collection of beer data scraped from BrewLogix [1]. The dataset contains over 30,000 beers stored as JSON files, where every JSON file is a collection of beers. The records contain almost all the data that I need, plus a lot more that I do not need, and seem quite complete. The dataset size is 80MBs.

2 DATA QUALITY

Before wrangling the data, I will talk about the data quality of the various data sources.

2.1 Untappd

The Untappd data contains all of the features I need. Most of these features do not have missing data: beer name, brewery name, beer type, ABV, IBU, and personal rating have 0% data missing. The flavor profiles experienced by the user have 1% missing data, but since that's so little, I expect to be able to handle that issue easily. More problematic is the feature for textual descriptions: 77% of the check-ins do not have a textual comment, and of the remaining 23%, not all comments are semantically relevant to the beer itself. I expect that this feature is not one I can make good use of.

Based on box-and-whiskers-plot and barplot visualizations, seem to be no outliers present in the data. Two features are units of measure, ABV and IBU, and they are formatted correctly: ABV as a percentage, and IBU as a float. The check-in timestamp is formatted to the standard ISO format, and floats use periods as the decimal mark. The dataset is in an unnormalized form.

Although the Untappd data provides a unique ID for each record, for our purposes, that is not the primary key of our data: that would be beer name + brewery name. We will assume that a brewery does not make multiple different beers that share the same name. Given this primary key, the Untappd dataset consists of 18% duplicates. This can be fixed during data wrangling: duplicates are different check-ins of the same beer. Features like beer name and brewery will be consistent across duplicates. For some features, we must apply some transformation. For example, personal ratings can be averaged, and flavor profiles of all check-ins can be merged into one list.

DE'23-'24, Radboud University, Nijmegen, Netherlands
2024.

2.2 OpenBeerDB

2.3 BreweryDB

3 DATA WRANGLING

Preferably, our pipeline should serve two different sets of data: the small personal dataset with a user's ratings, and the big general dataset to query beers from. We will store these datasets as parquet files. These datasets should share the same features, with the exception of the personal rating for the personal dataset. This list of features should be as follows:

- Beer Name
- Brewery Name
- Beer Category (General)
- Beer Type (Specific)
- Alcohol By Volume (ABV)
- International Bitterness Unit (IBU)
- Textual Description
- Flavor Profile

We want to represent these features in a tabular manner. Most of these features fit this structure. However, flavor profiles pose a problem, as they are currently stored as a list of varying size for every record in the Untappd dataset. My proposed solution is to wrangle the flavors into binary features: every flavor found in a flavor profile becomes a feature. If a beer's flavor profile contains that flavor, the feature's value is 1, otherwise it is 0. The main problem with this approach is that these flavor features are only stored in the Untappd data. If we want to use the features for the general dataset, we will have to extract that knowledge somehow. For example, it could be collected from the Untappd website, or the data scientists could use NLP techniques on a beer's name and description to fill in the features. Regardless, making full use of the flavor features will require additional work.

All data sources are currently represented in a tabular form (Untappd, OpenBeerDB) or a semi-structured form (Rémi). The semi-structured data is represented using multiple JSON files of the same shape. I will wrangle it into a tabular format using Python and Pandas, applying further operations on dataframes. Since OpenBeerDB's data is served as a set of SQL files that create tables with records to query, I will use SQL to get the features that I need, and wrangle those features using Python. I can perform this process in bulk.

I want to join the datasets by OpenBeerDB [2] and Rémi [3], assuming a primary key of Beer Name + Brewery Name. However, this is not possible, because Rémi's dataset does not contain information about a beer's brewery. Although it would be possible to create a program that would try to scrape brewery information from Untappd, for now, I will refrain from building such a program due to project scope constraints. Such a scraper could be worth the effort, though, since I could use it to fill in missing data, and standardize existing data to Untappd's standards. For now, my approach in joining this data shall be that I remove records from Rémi with duplicate names, then join on Beer Name.

Once the data for the general dataset has been joined, I will have to apply additional operations, because the different data sources have different possible sets of values for the same feature. The Untappd dataset only has one column about beer category, but it actually contains both beer category and beer type: beer type is the full string, while beer category is the substring prior to the "-". The other datasets already have two different features for beer category and type, but sometimes use different names for the same category/type. Changing these values to be the same as the one in the personal dataset would improve data quality.

REFERENCES

[1] BrewLogix. 2024. BreweryDB. https://www.brewerydb.com/
[2] OpenBeerDB. 2024. OpenBeerDB. https://openbeerdb.com/
[3] Philippe Rémi. 2019. beer-dataset. https://github.com/philipperemy/beer-dataset
[4] Untappd. 2024. Untappd. https://untappd.com/