Using the Untappd API & Data Wrangling

Date: May 16, 2017

The Data

For my initial capstone project ("A Guide to Selecting Your Next Craft Beer"), the primary data source is the Untappd application. According to their website "Untappd is a new way to socially share and explore the world of beer with your friends and the world." Practically speaking, Untappd is an application that allows its users to keep a log and rate the beer that they drink. Ratings are based on a 0-5 numeric scale allowable at 0.25 increments, with 5 being the highest rating. In addition to user ratings, there appear to be 30+ additional factors for each beer that may be used for analysis and prediction.

My project will leverage both my personal dataset of beers tasted as well as a dataset of untasted beers.

Data Acquisition Methodology

1. Acquire access to the Untappd API. Authenticate and explore the API and JSON formats.
2. Access and acquire a dataset (~200+ distinct beer records) of my personal ratings and history.
    a. Explore, analyze, and clean dataset.
        i. Easy to acquire, all results returned in one API call.
3. Access and acquire a dataset (~20,000+ distinct beer records) of untasted "new" beer to forecast ratings.
    a. Explore, analyze, and clean dataset.
        i. Difficult to acquire given only 100 records are returned per API call. This means it will require ~200 hours to return ~20,000 records.

Challenges

- Acquiring API access proved to be difficult as I had to exchange a few emails with the Untappd technical team and explain the purpose of my capstone project.
- **The primary challenge with acquiring the dataset is that Untappd limits the number of API calls allowed to 100 per hour**.
    - I plan to circumvent this by building scripts to acquire and load the data into CSV files (and eventually pandas data frames) incrementally.
- Running scripts hourly means that my computer constantly needs to be powered, as well as on power supply.

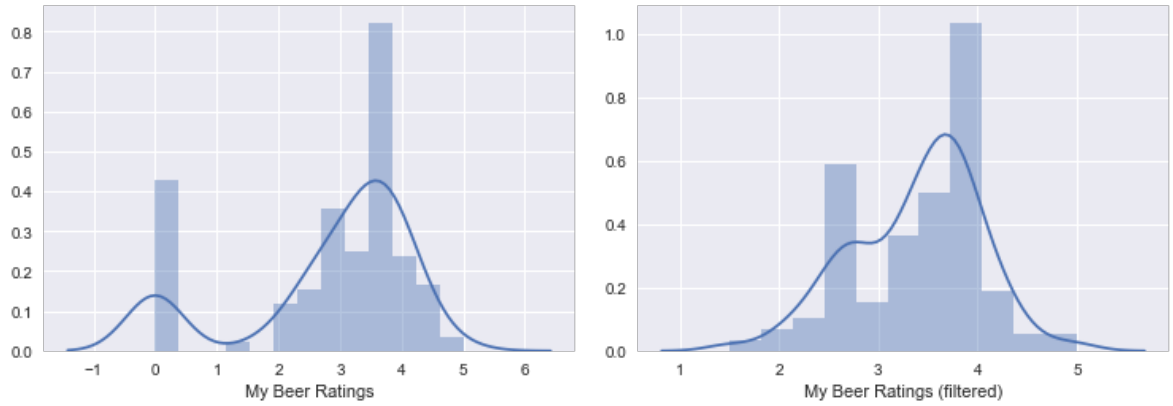
Analyzing the "My Personal Beers" Dataset (Dataset #1)

Initial Data shape: 218 rows (unique beer records), 35 columns (beer features)

Data Types: bool(1), float64(6), int64(9), object(19)

Null Values: Yes, some of the qualitative fields have null values (e.g. beer description, brewery city name). Fill the null values with empty strings (for now).

Outliers: Removed records if my rating was 0.0 or negative (meaning I likely forgot to rate the beer).



Left: Seaborn "distplot" of the initial data, some beers have ratings of 0.0.
Right: Seaborn "distplot" of the filtered data. Removed beer records with 0.0 (since it was user error).

Initial Count:	**35 columns, 218 records**
Null values removed:	*none*
Outliers removed	*36 records*
Final Data shape:	**35 columns, 182 records**

Analyzing the "New Beers" Dataset (Dataset #2)

Initial Data shape (as received via API calls in JSON): 29175 rows (beer records), 53 columns (beer factors)
Initial Data Types: bool(1), float64(15), int64(11), object(26)

Qualitative Fields and Null Values: such as "beer description", "brewery Facebook page", "twitter name", etc. were identified and removed. This accounted for 30 factors (that were removed).

Duplicate information: the #1 challenge with this dataset is the amount of duplicate records returned in the API calls. I believe that this is due to the fact that any user may create a record

for a beer, regardless if it already exists. This is clearly not ideal, especially in a production environment. This will be discussed in further detail in the capstone final report. 9308 records (32% of the acquired records) were dropped since they were duplicates.

Null Values: Yes, some of the descriptive records (brewery city, state) were null. Fill with empty strings for now.

Outliers: Yes, 29 records had no beer weighted rating score so we will remove these records.

count	19867.000000
mean	3.524786
std	0.356703
min	0.000000
25%	3.491960
50%	3.589430
75%	3.649190
max	4.749380

count	19838.000000
mean	3.529939
std	0.330505
min	1.081710
25%	3.492635
50%	3.589645
75%	3.649280
max	4.74938

Left: Without outliers removed we can see 0.00 values for the weighted beer ratings.
Right: Only a slight change, but we have removed the 0.00 values for the weighted beer ratings.

Initial Count:	**53 columns, 29175 records**
Features removed:	*30 columns*
Duplicates removed:	*9308 records*
Null values removed:	*none*
Outliers removed	*29 records*
Final Data shape:	**23 columns, 19838 records**

Final Data Types: bool(1), float64(5), int64(9), object(8)