Docker commands:

docker build -t xgb\_fastapi:latest .

docker run -dit --rm --name init\_xgb\_fastapi -p 8080:80 xgb\_fastapi:latest

http://localhost:8080

http://localhost:8080/health

http://localhost:8080/docs

[2.16] In yout browser, copy-paste http://localhost:8080/mall/customers/segmentation?genre=Female&age=65&income=38&spending=35

docker stop init\_xgb\_fastapi

hello:  
"This project was designed to take a few key features from a beer"  
"styles data set and then build a model that would take those inputs"  
"and then predict the given beer style. It takes four inputs"  
"(review\_aroma, review\_appearance, review\_palate, review\_taste)"  
"and each one must be given an integer score from 1-5 and then it will predict"  
"the beer style."  
"Here is the github repo: <https://github.com/freescania/advdsi_at2>"

Student: Angus Nicholson

Project Name: Assignment 2

Date: 14 March 2021

Deliverables:

Github repo: <https://github.com/freescania/advdsi_at2>

## Business Objective

We are a data science consultancy working for a local beer specialist store in Sydney. They have hired us to make a web app that will contain a machine learning model that will help recommend beers to people depending on their personal tastes. They have a large dataset with 104 different beer recommendation possibilities and one element they hope the project will accomplish is that it may entice consumers to look at some of the many exotic beers that they stock at the specialist beer store.

They would ideally like a web app and model that provides a relatively high degree of accuracy for people’s tastes, but also is able to show them interesting beer recommendations that they may not have thought of. They only wanted the model to use brewery\_id, review\_aroma, review\_appearance, review\_palate, review\_taste to predict beer\_style.

## Hypothesis and Objectives

We will be looking at building a neural network in pytorch and will investigate the best neural network architecture to try and deliver high quality predicted results based on consumers beer preferences.

## Data Preparation

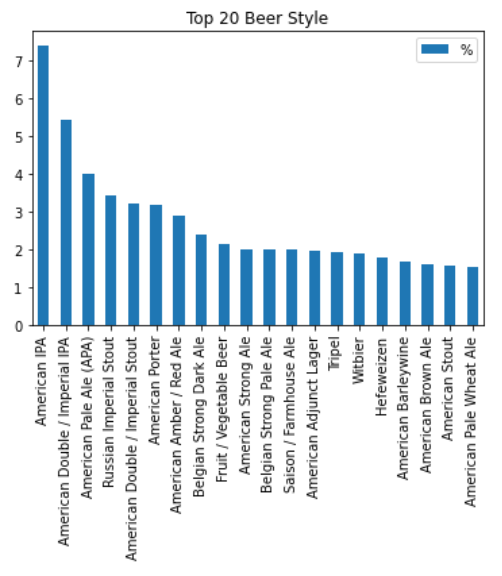
We received a CSV file named beer\_reviews that contained 1,586,614 rows and 13 columns of data. As noted above, the beer store only wanted the model to work with five features (brewery\_id, review\_aroma, review\_appearance, review\_palate, review\_taste) to predict the target variable beer\_style. And within beer\_style there were 104 different classes of beer. This immediately made it clear that we should regard this as a multiclass classification problem.

We first removed the unwanted columns, then brewery\_id we classified as a categorical variable, while for the rest we classified them as numeric variables. We tried to use one hot encoding on brewery\_id but there were 5840 unique brewery\_id which dramatically would have increased the size of the training data set in an already very large dataset causing CPU processing issues.

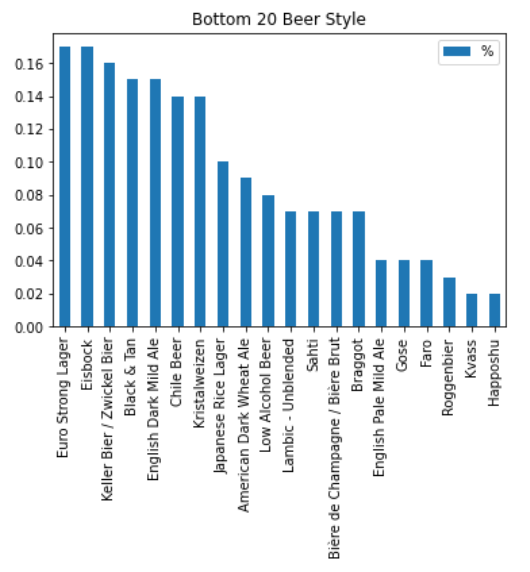
Given we wanted the target variable to stay as one column we decided to use an ordinal encoder to change the beer\_styles into numeric values so the model could process it even if the ordinal difference between them was largely irrelevant.

## Data Exploration

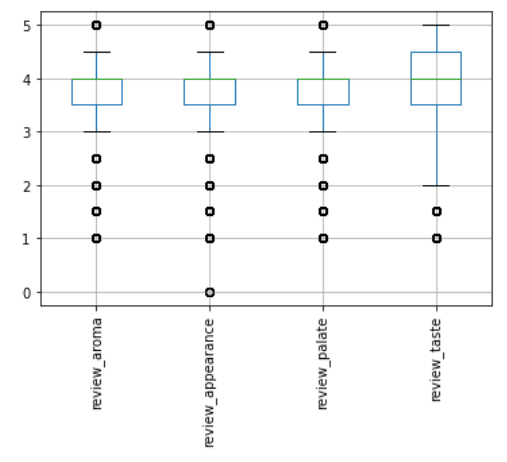
We began to look into the distribution of the 104 classes in the target variable. It was a comparatively large amount of classes with from a small number of features. The highest frequency classes are displayed below and the most frequent “American IPA” only appears just over 7% of the time, which does raise some concerns about how easy it will be to train the accuracy of the model over such a large possible set of classes in the target variable.



While for the least frequent beer styles, some such as Happoshu were appearing as infrequently as 0.02% of the time in the dataset, which doesn’t give one great confidence of building a model that would accurately predict this occurrence. The hope is that the data set is large enough that some of these low data frequency issues may be balanced out given the amount of data to train upon.



Of the four numerical features, it was clear that the were scored from 0 to 5 presumably as people reviewed the beers. One issue is that 0-5 scoring across four features may not provide enough variance in those features to help sort through the 104 classes in the target variable.



XGB results

