Course: Advanced Data Science for Innovation

Student Name: Hitoshi Fukuda

Date: 19 December 2022

**GitHub Link:**

|  |  |
| --- | --- |
| **GitHub for Model** | <https://github.com/hitoshi0531/adv-dsi-2022-at2> |
| **GitHub for API** | <https://github.com/hitoshi0531/adv-dsi-2022-at2-api> |

**API URL:**

|  |  |
| --- | --- |
| **API root** | [https://polar-sierra-70192.herokuapp.com](https://polar-sierra-70192.herokuapp.com/) |
| **API health check** | <https://polar-sierra-70192.herokuapp.com/health> |
| **API document** | <https://github.com/hitoshi0531/adv-dsi-2022-at2> |
| **API beer prediction sample** | <https://polar-sierra-70192.herokuapp.com/beer/type?brewery_name=Amstar&review_aroma=5&review_appearance=5&review_palate=5&review_taste=5&beer_abv=5> |

# ****Assessment 2: Neural Network Model with API - Final Report****

This report states my experimental steps on a machine learning project following the CRISP-DM methodology, and presents achievement that I found, issues encountered, and my effort made to the project.

## Business Understanding (Background & Objective)

In the beer business, breweries are producing a variety of beers by their own production methods with their original tastes, appearances, aromas, palates, and beautiful names of the beers. By creating a lot of beers, the breweries aim to distinguish their own brand and to attract and get as many consumers as possible, and then earn profit from them.

In this project, I am provided with those beer-related data that are evaluated and posted by consumers, and I am tasked to build and deploy a Machine Learning model into the production of a API service for predicting beer style based on the data inputted into the API.

Hence, the main business objective of this project is to build the API service and provide the accurate prediction of beer style. This API would be used by consumers, breweries, and some beer-related associations. The API can stimulate consumers motivation of buying beers and post their evaluations onto the internet, which will in turn grow beer market. The result of the predictions by API can also be utilised for breweries to classify their own beers into accurate categories and to label them on the beers. Therefore, if the accuracy of prediction by the API will reduce their benefit, so inaccurate result will deteriorate consumers’ interest and breweries’ production.

## Data Understanding

### Datasets

I am provided with a CSV file for this project, and so the dataset can be input into a form of dataframe by using a method of “read\_csv” provided by pandas.

* beer\_reviews.csv: a dataset for training machine learning models

However, since the dataset is only a single set, I should split them into three datasets for training, testing, validating.

#### Observations/Rows

Each row is a review of a beer.

#### Target Variable

The target variable for prediction is ‘beer\_style’.

* beer\_style: Type of beer, such as American IPA, English Pale Ale, Czech Pilsener, etc.

#### Features

Each column is characteristics of each beer and information of a review posted on the internet. The dataset given for the experiment originally includes following features.

(\*doubt that the requirements are different between the ‘assignment page’ online and ‘brief’ of word file, so my work is based on the requirements on the brief file)

| No. | Feature | Description | Type | API Expected Parameter |
| --- | --- | --- | --- | --- |
| 1 | brewery\_id | Identifier of brewery | int64 | No |
| 2 | brewery\_name | Name of brewery | object | **Yes** |
| 3 | review\_time | Timestamp of review | int64 | No |
| 4 | review\_overall | Overall score given by reviewer | float64 | No |
| 5 | review\_aroma | Score given by reviewer regarding beer aroma | float64 | **Yes** |
| 6 | review\_appearance | Score given by reviewer regarding beer appearance | float64 | **Yes** |
| 7 | review\_profilename | Profile name of reviewer | object | No |
| 8 | review\_palate | Score given by reviewer regarding beer palate | float64 | **Yes** |
| 9 | review\_taste | Score given by reviewer regarding beer taste | float64 | **Yes** |
| 10 | beer\_style (target) | Type of beer | object | No |
| 11 | beer\_name | Name of beer | object | No |
| 12 | beer\_abv | Alcohol by volume measure | float64 | **Yes** |
| 13 | beer\_beerid | Identifier of beer | int64 | No |

## Data Exploration

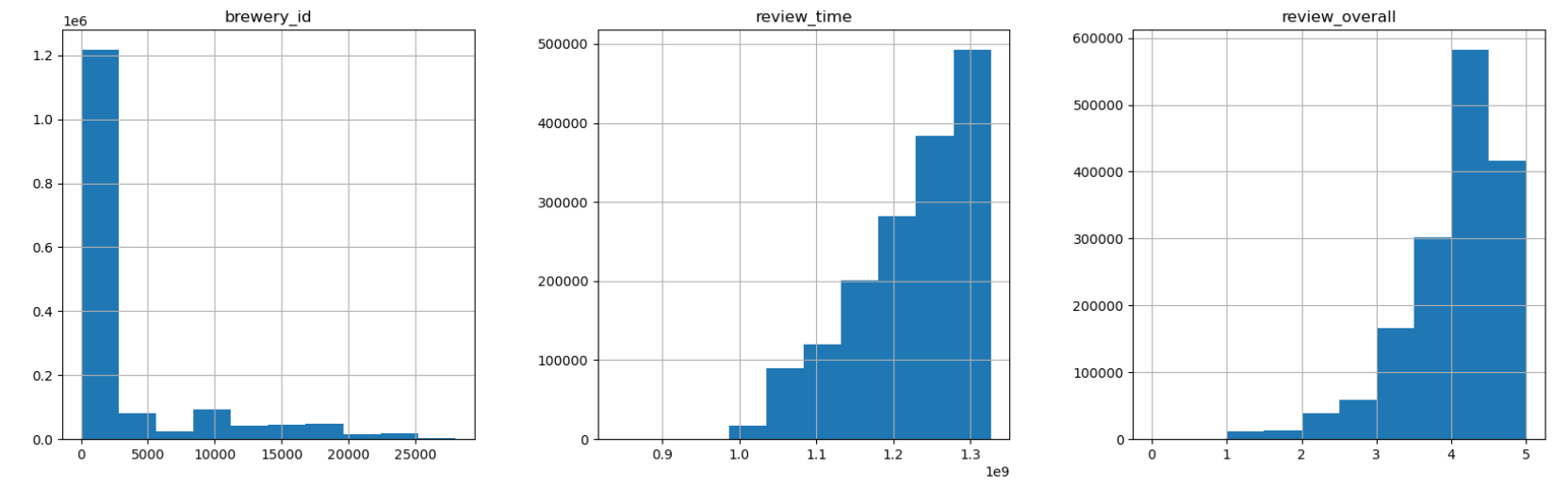
#### Structure

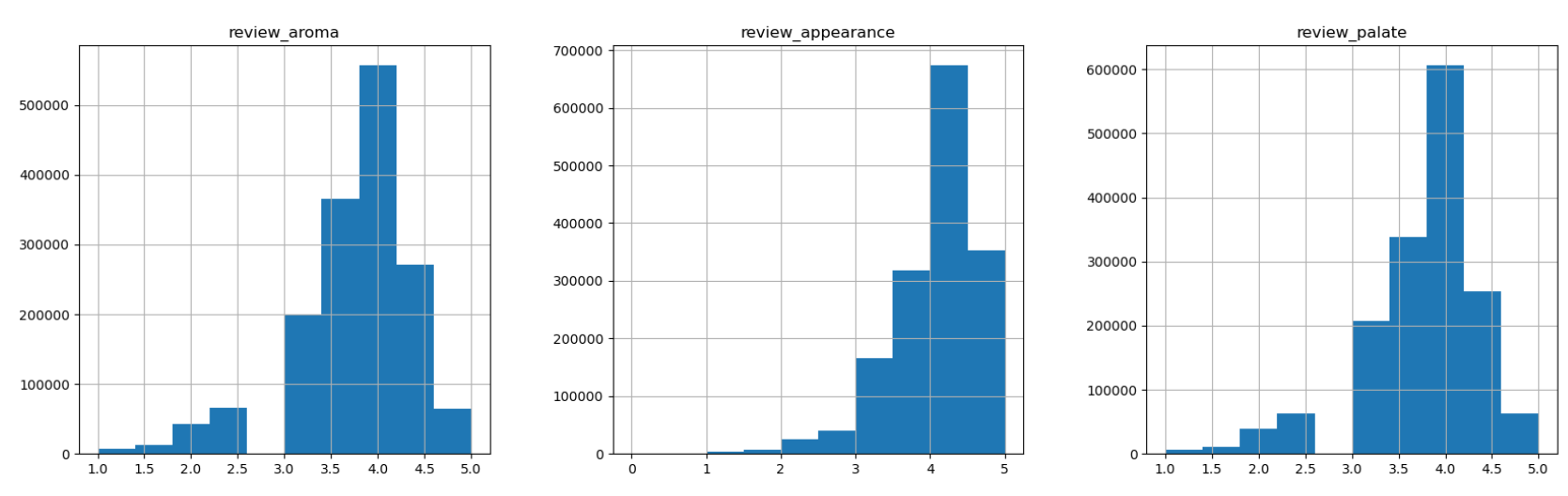
The dimension of the dataset is below:

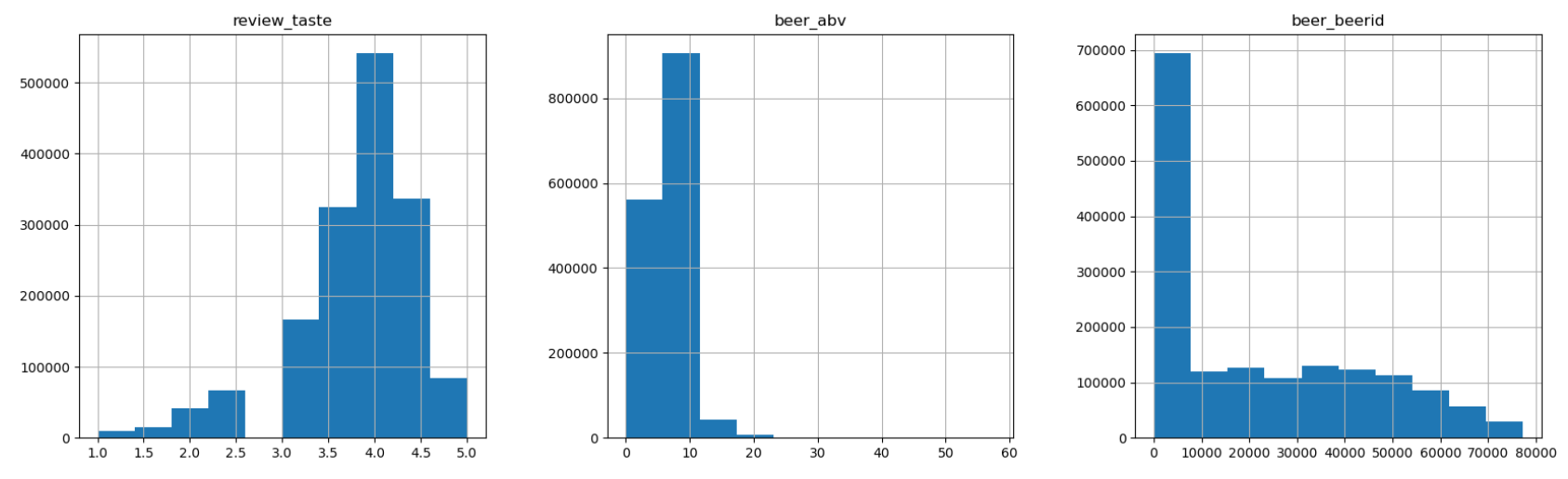
* 1,586,614 rows and 13 columns

#### Distributions

The distributions of numeric features are below:







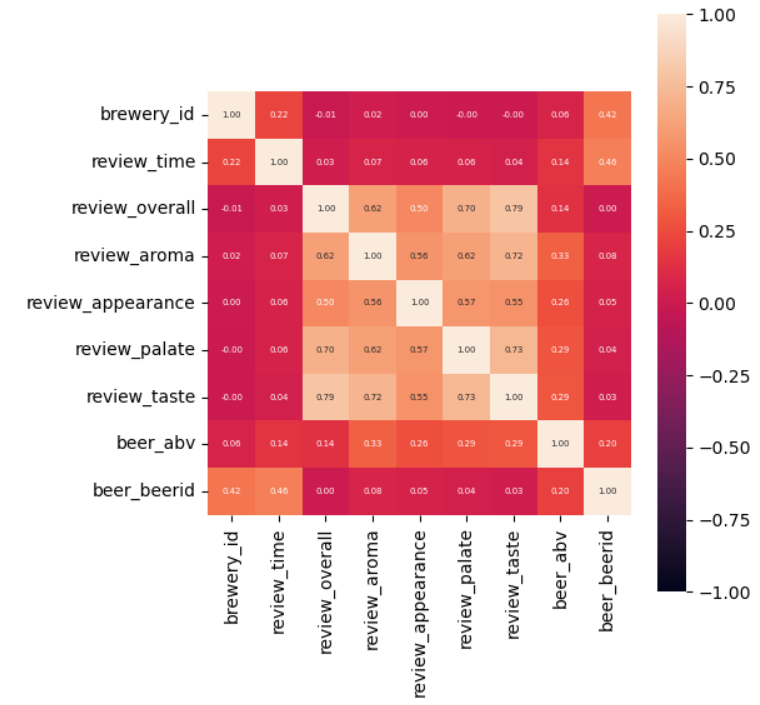
#### Missing Values

There are missing values in the dataset, and hence, I deleted the rows that include missing values. Missing values of the dataset are below:

| No. | Feature | Count of missing values |
| --- | --- | --- |
| 1 | brewery\_id | 0 |
| 2 | brewery\_name | 15 |
| 3 | review\_time | 0 |
| 4 | review\_overall | 0 |
| 5 | review\_aroma | 0 |
| 6 | review\_appearance | 0 |
| 7 | review\_profilename | 348 |
| 8 | review\_palate | 0 |
| 9 | review\_taste | 0 |
| 10 | beer\_style (target) | 0 |
| 11 | beer\_name | 0 |
| 12 | beer\_abv | 67,785 |
| 13 | beer\_beerid | 0 |

#### Correlation between the features (variables)

The correlations between the features are demonstrated in the matrix below:



#### Target Variable

The unique number of the target variable ‘beer\_style’ is 104, so I set 104 output units in my neural network architecture.

## Data Preparation

1. Drop the features that are not used for the API’s input/output parameter

Remaining features are:

'brewery\_name', 'review\_aroma', 'review\_appearance', 'beer\_style', 'review\_palate', 'review\_taste', 'beer\_abv'

1. Applying OrdinalEncoding to categorical features

Now, ‘beer\_style’ and ‘brewery\_name’ are converted into numerical features.

1. Applying StandardScaler to numerical features

Here, the scales of numerical features (including ‘brewery\_name’, except for ‘beer\_style’) are standardised.

1. Split the dataframe and save them

Split the dataframe into three datasets so as to use them as training, testing, and validating datasets, and save them in my data folder

## Modelling

I used Neural Network by Pytorch for predicting a variety of beer styles of target, as it is suitable for multi-class classification; other models that I have learnt are not suitable for multi-class classification such as Logistic Regression (suitable for binary classification)

The neural network has the following structure:

* (layer\_1): Linear(in\_features=6, out\_features=128, bias=True)
* (layer\_2): Linear(in\_features=128, out\_features=128, bias=True)
* (layer\_out): Linear(in\_features=128, out\_features=104, bias=True)

The hyper parameters are below:

* Learning rate: 0.1
* N\_EPOCHS: 30
* BATCH\_SIZE: 200

### Assessment of Performance

The performance of the model is below:

* Loss: 0.0208
* Accuracy: 0.0741

## Difficulty

The most difficult thing for me in this project is deploying the model as the API. Before deploying the model on Heroku, I had a confusion and trouble in understanding the relations and dependency between my python model, git, docker, and Heroku. I spent a great deal of time for solving problems in error of python method, connecting GitHub with Heroku, deploying the API onto Heroku, and access to the API online. Those processes are difficult for me to implement not ‘lab work’ but my actual materials because I must figure out the solution to adjust the materials by myself.

Another difficulty is in learning the whole course and working on my assignment by English because I came from Japan a few months ago and am not familiar with English, so it is very hard to learn knowledge, search for trouble shooting, and reach to the solution, but I made as much effort for learning English as possible and managed to make up my works.

## Deployment

This model can be used for predicting beer styles based on the expected input-parameters, and the accuracy of prediction is still not so good, so the model could potentially be improved by further feature-engineering, optimising hyper-parameters, and enhancing the architecture of Neural Network.

By utilising and applying this type of Neural Network and API combination, the model could be converted into other multi-class classification services such as those for searching rental houses, recommending food restaurants, and guiding students to ideal courses.