|  |
| --- |
| Assignment 2 – Beer Style Prediction  er Style Prediction  Advanced Data Science for Innovation  Advanced Data Science for Innovation |
| Mark Brackenrig  Due: 14th March  Student ID: 12964298 |



Table of Contents

[Business Understanding 3](#_Toc66619726)

[Data Understanding 3](#_Toc66619727)

[Data Preparation 4](#_Toc66619728)

[Modelling 5](#_Toc66619729)

[Architecture 5](#_Toc66619730)

[Evaluation 6](#_Toc66619731)

[Deployment 7](#_Toc66619732)

# Business Understanding

The purpose of this project is to accurately predict the beer style of a particular beer based on a BeerAdvocates users’ rating of the beer. Fields include criteria such as appearance, aroma, palate or taste, as well as the name of the brewery.

The business has shown interest in making the model available for production. While no specific use case was included in the brief, this could be used to automatically fill in the ‘beer style’ field of a review form, to assist the reviewer in generating their review.

# Data Understanding

The data was sourced from BeerAdvocates, a large beer review website. The data contained 1,586,614 records with 13 columns. The target variable was “beer\_style” – which contained 104 unique styles of beer.

Only 5 fields were kept for modelling purposes – review\_aroma, review\_appearance, review\_palate, review\_taste and brewery\_name. All fields except for brewery name were numeric, ranging from 1 to 5. There were 5742 unique breweries in the dataset.

# Data Preparation

Due to the large volume of data, only 60% of the data was used for modelling. The target variable was encoded using the OrdinalEncoder method from sklearn. While the data is not ordinal, this function still index encodes the target variable. All numeric data was normalized using the StandardScaler function.

For the “brewery\_name” field, only 2500 breweries were considered for encoding into the model. All other breweries were considered as ‘other’. This was because of a data processing error when generating a one-hot encoded matrix for the brewery names. Generating a non-sparse matrix that size was too large for the hardware available.

However, this also had the added benefit of flexibility when strings were provided to the API. In the case where the argument passed to the API is unknown (i.e. a new brewery), the model will still be able to produce a prediction.

If time permitted, I would have refactored my code to include the ‘fitting’ of the one-hot encoder in the data generator to avoid the size issues and include all data in my model training. This would have involved building a custom data generator, rather than the out of the box pytorch data loader.

# Modelling

Based on brief requirements, the model selected was a neural network. Some experiments were completed using different architectures, however most resulted in poor results. The best performing models were typically shallow networks.

Initially, I only extracted the top 1000 brewery names for modelling. Intuitively this made sense as this represented 93% of the brewery names in the dataset. In one of my experiments I decided to increase the number of extracted brewery names to 2500.

## Architecture

Due to the limited data available, a shallow architecture was selected. The architecture consists of only 1 hidden layer and the output layer. Dropout regularization is applied to the hidden layer. The relu activation function was applied to the hidden layer, and a softmax activation was applied to the output layer.

Text

Description automatically generated

Figure 1: Architecture of Neural Network

The relu activation function was chosen due to its popularity in hidden layers. The softmax activation function was chosen to allow for multiclass prediction.

# Evaluation

The evaluation metric used was

# Deployment

The model was deployed to a Heroku server on their free tier.