36114 Advanced Data Science for Innovation

Assignment 2 Report

Richard Rui Hu Zhang 13627753

Github repository: RichardRuiHuZhang/36114-assignment2-2022

API URL: https://ruihuzhang-36114-assignment2.herokuapp.com/

Introduction

A dataset of beer reviews is used to generate a machine learning (ML) model to predict the beer type with the beer brewery name as well as user scores. This allows the beer brewery to study the different beer brews for their own brewing experimentations and marketing.

Business Understanding

The provided dataset is a beer review dataset. A ML model based on this dataset allows the brewer to attempt to match a new brew of beer with test review scores against existing brews. This then allow the brewer to estimate the commercial potential of this new brew by researching the predicted brew made by the ML model. As this ML model is deployed on an Internet webpage for public use, this allows the test results to be available via any device that is connected to the Internet to generate the prediction.

Data Understanding

The dataset contains 195713 rows of data, with 13 variables (see Appendix A). The variable beer_type is the target variable. This is a factor variable, hence the goal of the ML model is a multi-class classification model.

Out of the remaining variables, the following are used for input into the ML model. These are therefore the independent variables considered for the ML model training purpose. Other variables are therefore discarded prior to model training, and are not expected for subsequent input of test dataset.

- brewery_name,
- review_aroma,
- review_appearance,
- review_palate,
- review_taste,
- beer_abv

Five of these independent are numerical variables of float type. These are listed below. From the dataset it can be inferred that the first four of these variables listed below all have the range of 0.0 to 5.0. The variable beer_abv ranges from 0.0 to 57.7 from the dataset. This is a reasonable range for the alcohol content in percentage for consumable alcohols in general.

- review_aroma,
- review_appearance,
- review_palate,
- review_taste,
- beer_abv

For the seven variables considered in the ML model, the following number of null values are observed. These rows are dropped. 188930 rows remain.

- beer_abv: 6783 counts

For the string values of the independent variable brewery_name, 743 unique strings are identified. For the string values of the dependent variable beer_type, 103 unique strings are identified.

Model Pipeline

The following pipeline is developed for this webpage app.

1. Preprocessor column transformations:
    a. For the factor column brewery_name, a Ordinal Encoder is used to convert each name string to an integer. Given the large number of unique name strings (743), one hot encoder will generate too large a dataframe for training purpose (i.e. a 743 rows by 188930 columns for brewery_name alone). An integer encoding is sufficient for the purpose of differentiating the brewery names.
    b. For the five numerical columns, a min-max scaler of a range of 0.0 to 1.0 is used. This brings all data range to the same range to avoid any numerical issues caused by the differences in the value ranges.
2. ML model fitting: This will be the option that is switched to the optimal model based on the study outlined in the later section of this report. This outputs an integer for the target variable.
3. The target variable is transformed using inverse transform of a second Ordinal Encoder to recover the string corresponding to the predicted class integer.

ML Model Training and Selection

Performance Metric of the ML Model

As this is a multiclass classification model (i.e. prediction of one class out of 103 unique brew names), the model accuracy can be used to measure the ML model performance. This is a good general measure for obtaining the positive prediction.

Logistic Regression

A basic model for multiclass classification is generated. The default parameter values are used for this model. This is used as a base model to estimate the likely accuracy of a shallow ML model.

Random Forest

A second shallow model, random forest, is generated. This tests whether a set of simple decision logics can make accurate prediction. Grid search is performed.

Neural Network

Neural network models of various depth have been generated. Both PyTorch and Scikit Learn packages have methods for the definition of a neural network ML model. As the PyTorch is a deep learning package, it is the better package in terms of generating ML models with relatively good accuracy, even without optimising the hyperparameters of the model.

ML Model Evaluation and Selection

Shown below is the model accuracy metric comparison table. From this it can be shown that the Scikit Learn random forest model after three grid searches is most accurate based on all ML models trained. Therefore in terms of model performance, this is the best model for deployment.

In terms of the relatively low accuracy of the ML models, this can be explained through the relatively small amount independent variables (6 variables) with relatively large number of target classes (103 unique names). There is likely insufficient data to differentiate between the large number of target classes, hence the ML models are unlikely to make accurate predictions.

Also, as the model requires to be deployed to an Internet website, which is hosted on Heroku. As Heroku is a third party commercial organisation, the memory size of the overall package needs also to be considered due to the limited allowable memory size for a free user account. When attempting to deploy a PyTorch model to the Heroku the package requires more than the free account limit of 500 MB (the typical memory requirement of a PyTorch model is around 900 MB), PyTorch models cannot be deployed for a free account. Therefore only a Scikit Learn model is required. From the Scikit Learn ML models, the best is a Random Forest model trained with cross-validation.

	Training	Validation	Test
Scikit Learn Models			
Logistic Regression	0.278066	0.281026	0.276769
Random Forest			
Base	0.278066	0.281026	0.276769
Grid search on max tree depth with 10-fold CV	0.417715	0.41786	0.413892
Grid search on minimum samples per node, with 10-fold CV and optimal tree depth	0.420081	0.420909	0.41608
Grid search on number of samples, with 10-fold CV and optimal tree depth and minimum samples per node	0.421604	0.422447	0.418356
Neural Network	0.153584	0.154527	0.152649
PyTorch			
One linear layer of 128 nodes, sigmoid	0.010575	0.011039	0.009933

activation			
One linear layer of 256 nodes, sigmoid activation	0.006578	0.007007	0.006651
Three linear layers of 512 nodes, relu activation and dropout of 0.2	0.003467	0.003453	0.003987

Table: Summary of the ML model accuracy score, with the highlighted row corresponding to the best performing ML model.

API Structure

The API based on the above-mentioned ML model pipeline is deployed on Heroku for public consumption. The Heroku app is linked directly to a Github repository (highlighted at the beginning of the report), and the app is automatically rebuilt after a modification to this Github repository. The following endpoints are available in this API.

- '/' (GET): Displays a brief description of the project objectives, list of endpoints, expected input parameters and output format of the model, link to the Github repo related to this project

- '/health/' (GET): Returns status code 200 with a welcome string.

- '/beer/type/' (GET): Returns prediction for a single input only. Each variable is requested by name as part of the input declaration.

- '/beers/type/' (POST): Returns predictions for the file name of a CSV file containing multiple inputs. This CSV file needs to have six columns, with the data in each column in the required formats.

Single Input Example

Type the following in the address bar

http://127.0.0.1:8080/beer/type?brewery=abc&aroma=1.0&appearance=2.0&palate=3.0&taste=4.5&alcohol=5.0

Multiple Inputs Example

Open http://127.0.0.1:8080/docs and use the POST endpoint.

Conclusions and Recommendations

A number of ML models have been generated on the beer review dataset. From these models, a PyTorch machine learning model is the best performing model in terms of accuracy of the model. On the other hand, due to the limitation of user account in terms of the package memory limitation, only a Scikit Learn ML model can be deployed. This is offset by a significant reduction in the model accuracy.

As a next step, a different method of hosting the API app can be examined. For example, a different commercial online platform could be considered. In this case, the primary considerations are memory allowance and cost of the hosting service. This allows apps with more Python packages, such as PyTorch, can be used as part of the ML model definition.

Another step that could be taken with the dataset is to use a different set of target values, so that the ML model performance could be better.

Appendix A: Data Dictionary

Column	Description	API Expected Parameter
brewery_id	Identifier of brewery	No
brewery_name	Name of brewery	**Yes**
review_time	Timestamp of review	No
review_overall	Overall score given by reviewer	No
review_aroma	Score given by reviewer regarding beer aroma	**Yes**
review_appearance	Score given by reviewer regarding beer appearance	**Yes**
review_profilename	Profile name of reviewer	No
review_palate	Score given by reviewer regarding beer palate	**Yes**
review_taste	Score given by reviewer regarding beer taste	**Yes**
beer_style (target)	Type of beer	No
beer_name	Name of beer	No
beer_abv	Alcohol by volume measure	**Yes**
beer_beerid	Identifier of beer	No