Advanced Data Modelling

Assignment 3 – COSC 2670/2732

Riya Minesh Amin

S3807007

Overview

In this report, we analyse the dataset of Alcohol Beverage reviews, and propose the models to make prediction of a user review score in a recommendation system. Three algorithms are analysed in the report to predict the ratings from the beer reviews based on all the other information. Initially KNNRegressor and DecisionTreeRegressor algorithms are compared. Then stacking generalization algorithm is built using the KNN and Decision Tree model as level0 models and Linear Regression as a Level1 model. Using the same levels of model, a Super Learner Ensemble is also built.

Exploratory Data Analysis

The training dataset of Beer review consist of 84508 reviews from 10696 users. The number of beers reviewed is 14228. The dataset consists of features like BeerID, ReviewerID, BeerName, BeerType and rating. Additional features like BrewerID, DayofWeek, Gender, and ABV were merged from the features.tsv file to train and validation sets. Before merging, categorical features like Gender and Dayofweek were encoded using Labelencoder.

To gain a sense of the data, basic visualisation was done.

Figure 1 shows the total number of people who rated each beer kind. Due to the large number of beers available, we limited it down to the top 15 most popular beers as rated by users. The top three beers most usually rated by users are American IPA, American Double/Imperial IPA, and American Pale Ale (APA), as seen in Figure 2.

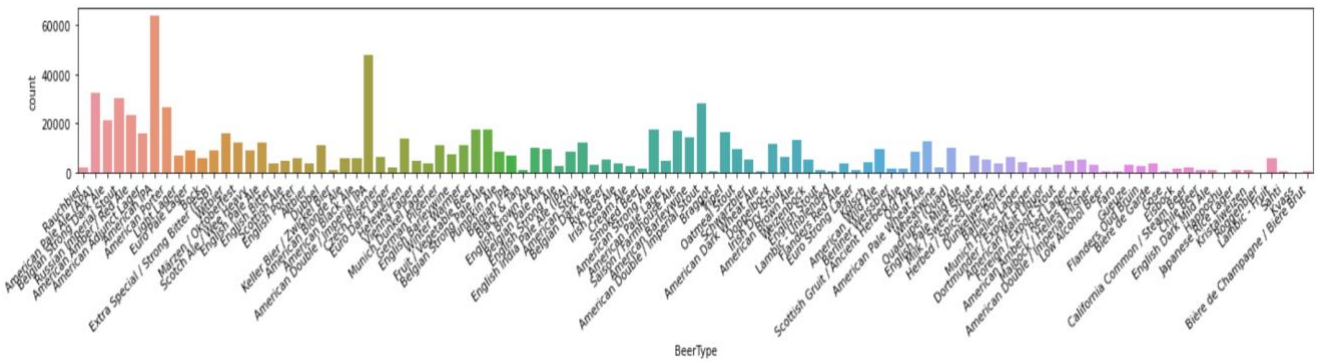

Figure 1: Total count of each beer rated by users

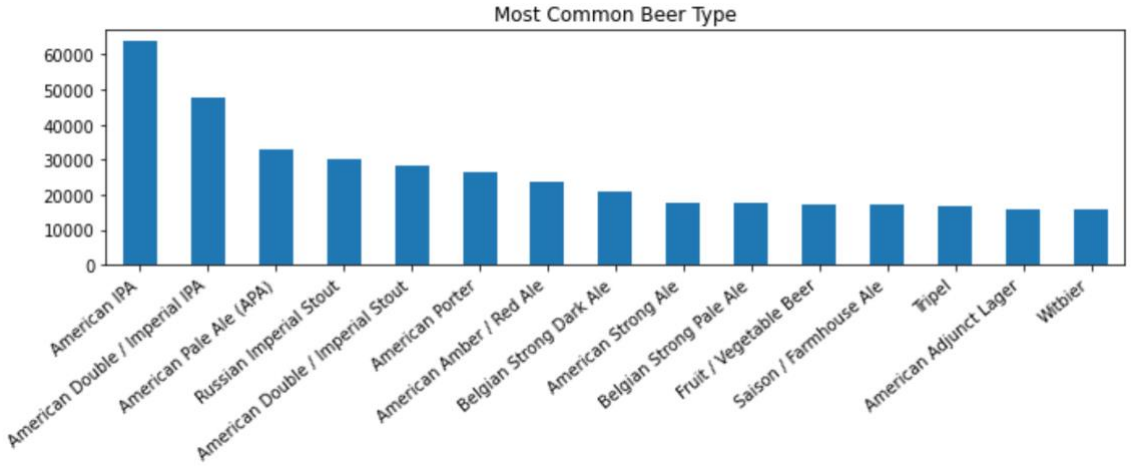

Figure 2: Most common beer rated by users

Figure 3 shows a distribution plot of Alcohol by Volume.

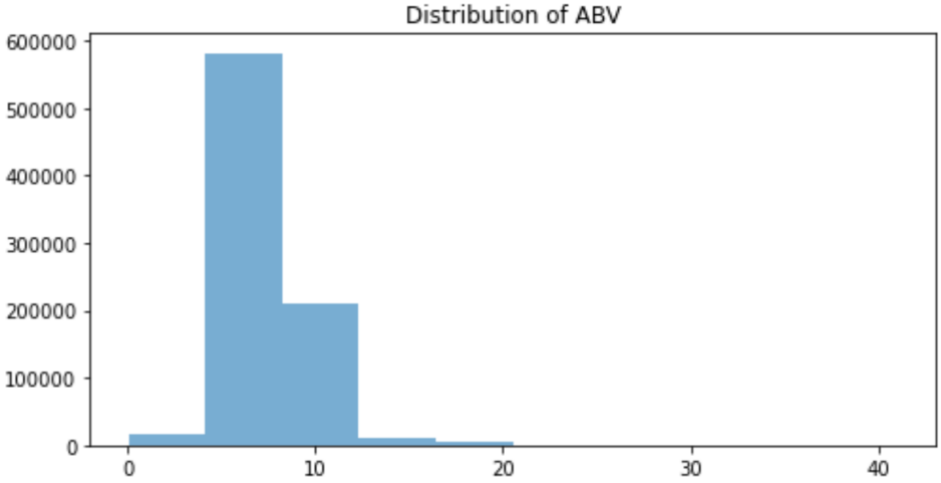

Figure 3: Distribution of ABV

In Figure 4 we see the top 10 strongest Beers by Alcohol by Volume. From figure 2 and 4 we see that one-off topmost commonly rated beer is the American Double/ Imperial IPA which also consist of highest alcohol volume.

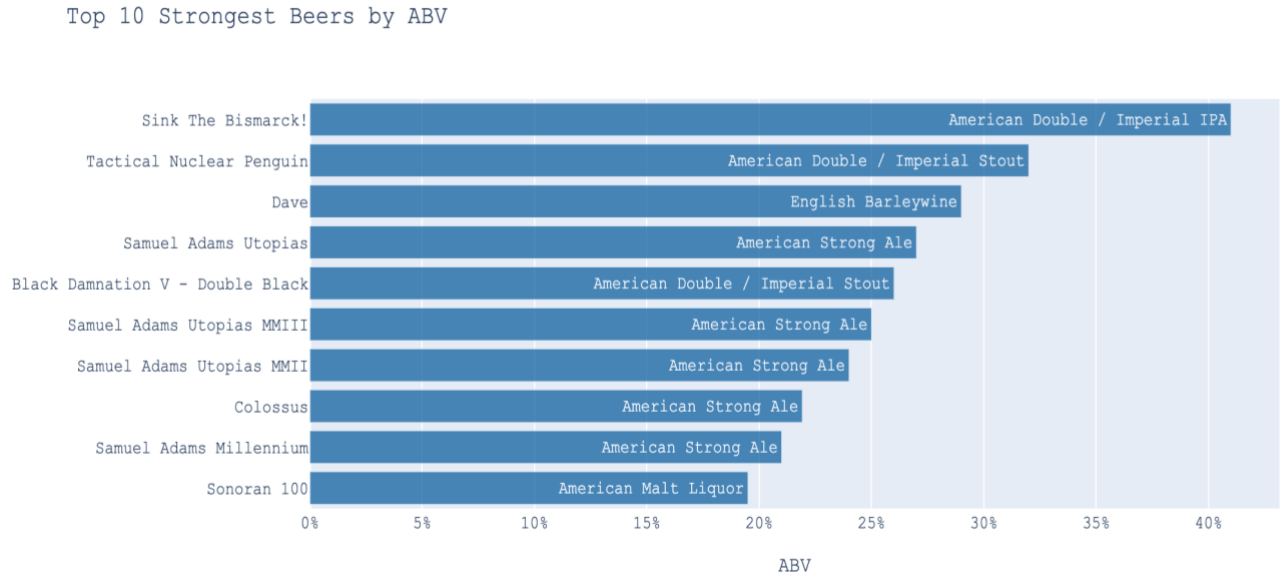

Figure 4: Top 10 Strongest Beer by ABV

In figure 5 we checked the correlation of the rating with gender and. ABV. We see that Gender and rating have no correlation, but ABV and rating seems to have a correlation of approximately 0.25.

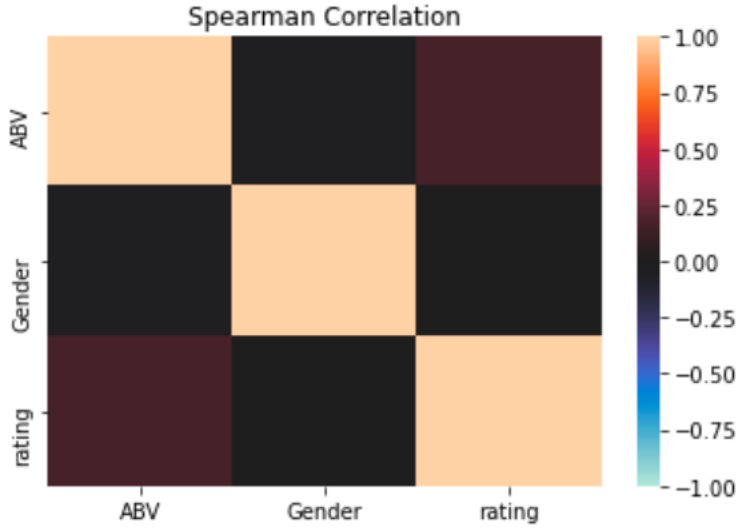

Figure 5: Spearman Correlation heatmap of ABV, Gender and Rating

Model Description & Experiments

Firstly, to test the performance of a Knn and Decision Trees, KNNRegressor and DecisionTreeRegressor were implemented on the dataset. To determine the hyperparameter of each model (model selection process), cross validation process is applied to compare and choose the best model when using each method. The Model performance and predictions are measured and scored by Mean Squared Error (MAE). KNN gives a mean MAE score of -0.572 with a standard deviation score of 0.003 on validation set. Whereas Decision Tree gives a mean MAE score of -0.636 with a standard deviation score of 0.004.

Since we got a reasonably good MAE value for both the algorithm, ***stacking generalization*** is implemented by combining KNN and DT as level 0 models and Linear Regression as a level 1 model.
The get_stacking() method is used to obtain the StackingRegressor model by first creating a list of tuples for each of the two base models, and then creating a linear regression meta-model to combine the predictions from the base models using 5-fold cross-validation. The function get_models() create the models we wish to evaluate. Then each model is evaluated using repeated K-fold validation. Then an evaluate_model() function takes a model instance and returns a list of scores from three repeats of 10-fold cross-validation.

The stacking model is evaluated, and we got a mean MAE of about -0.521 and standard deviation of 0.001. As we see form figure 6 stacking model has got the highest MAE score.

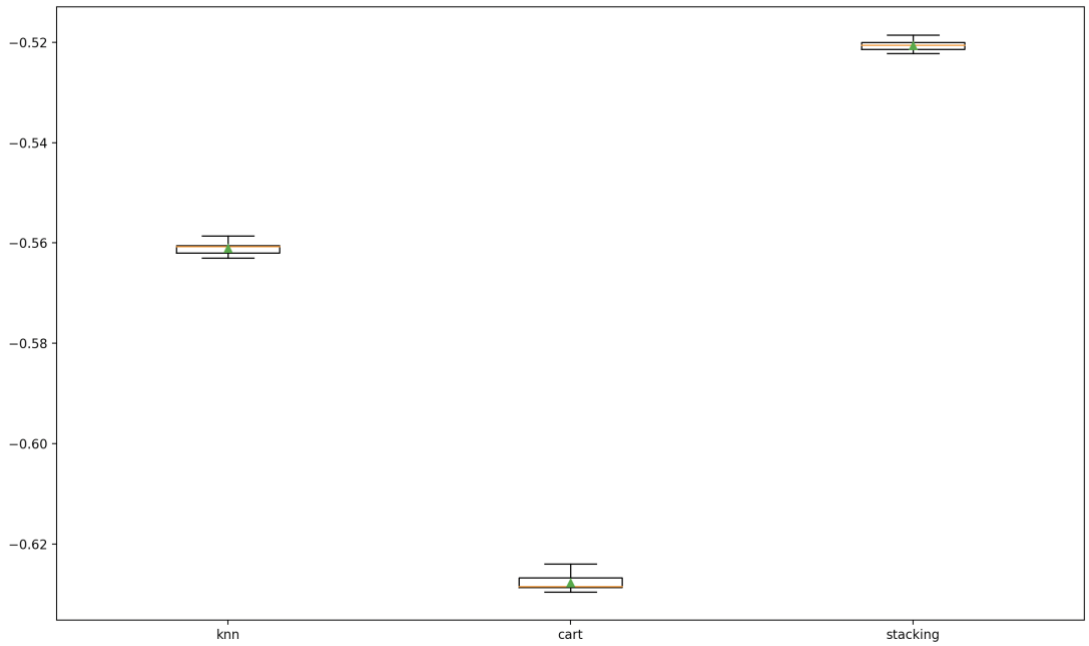

Figure 6: Mean MAE boxplot for KNN, Cart and Stacking models.

In our final model, we implemented a specialised type of stacking called ***super learner*** which uses a simple linear model as the meta-model. The super learner algorithm applies stacked generalisation, also known as stacking or blending, to k-fold cross-validation, in which all models utilise the identical k-fold splits of the data and a meta-model is fitted to each model's out-of-fold predictions. We again use get_models() function to define the models and return them as a list and use k-fold cross-validation to make out-of-fold predictions that will be used as the dataset to train the meta-model or "super learner." Here too we fit Linear Regression as blending model and evaluate the base-models on the validation sets. Predict_ensemble functions are created to make a prediction using the blending ensemble. We get a MAE score of 0.510 for the super leaner algorithm on the validation set. Where KNN gives a -0.579 mean MAE score and Decision Tree a -0.655 mean MAE score.

References:

- Brownlee, J. (2021). How to Develop Super Learner Ensembles in Python. https://machinelearningmastery.com/super-learner-ensemble-in-python/
- Brownlee, J. (2021). Stacking Ensemble Machine Learning With Python. https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/