Beer Review Regression Problem

Mohamad Mostafa
Politecnico di Torino
Student ID: s291385
s291385@studenti.polito.it

***Abstract*—In this report we develop a possible solution for predicting beer overall quality by means of regression techniques considering two sets of predictors, one related to the reviewer, and the other one is related to the beer. The proposed approach is based on selecting the most convenient feature engineering steps without losing important information, also some text analysis techniques were adopted. Best model have been picked from several models depending on the evaluation of the R2 score.**

I. PROBLEM OVERVIEW

The proposed competition is a regression problem on the Beer Review Dataset, a collection of beer overall quality ranging from "one" to "five" and other beer related information are provided by different users with other information related to the users. The goal is to predict the overall quality of beer. The dataset is divided in two different parts:

- A *development* set, containing 70,000 reviews of which the overall quality is known.
- An *evaluation* set, containing 30,000 reviews without the corresponding quality.

The regression model built is based on a set of predictors having different types which are presented in Table 1,and we notice that we have three features that are present to specify a beer which are its alcohol by volume and its name and style, then there is five features representing the user three of which are regarding the age of the user and one for the gender and last for user profile name, and finally we have set of reviews from a user to a certain beer which mention the score for its appearance, aroma, palate, taste,and text. Looking at the dataset we notice that no duplicates exist. However, we notice that there are two data quality problems which are the existence of missing values with different quantities among the variables which can have a significant effect on the conclusions that can be drawn from the data and which should be treated carefully depending on their amount and the variable itself and its type, see Table 1 .Also, there is the problem of high cardinality of categorical variables which as we see from Table 1 can be very high causing a very large sparse feature matrix. It is important to notice that three of the features represent the age of the user which can be represented by one age feature.

Regarding the target variable it is our dependent variable which is numerical and ranges between one and five and half score are allowed representing the review quality for a specific beer given by a specific user. We can see in Fig. 1 the distribution of the target variable that behaves like a normal distribution with skewness of -1.023 which represent the asymmetry of the distribution having the tail of the distribution longer towards the left hand side of the curve as shown in Fig. 1 and since it is lower than -1 our distribution is highly skewed, also the distribution have kurtosis of 1.62 which as we can see have a high peek since its value is higher than 1.

Variables	Type	# NA
beer/ABV	Numerical	3,107
beer/name	Categorical	0
beer/style	Categorical	0
review/appearance	Numerical	0
review/aroma	Numerical	0
review/palate	Numerical	0
review/taste	Numerical	0
review/text	Text	18
user/ageInSeconds	Numerical	55,355
user/birthdayRaw	Date	55,355
user/birthdayUnix	Numerical	55,355
user/gender	Categorical	41,819
user/profileName	Categorical	14

TABLE I
TYPES AND MISSING VALUES OF VARIABLES

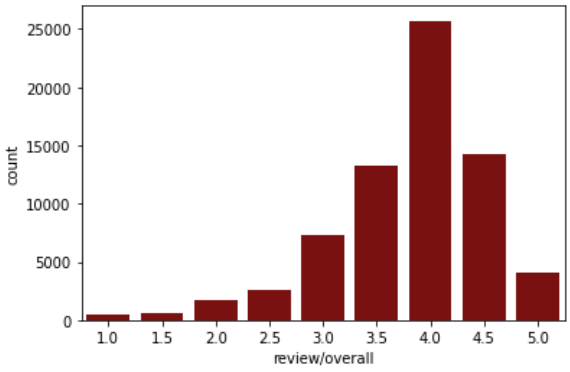

Fig. 1. Beer Overall Quality Distribution

II. PROPOSED APPROACH

A. Data preprocessing

We have seen that the features have various types. Considering the numerical features we notice from Table 1 that there are three features corresponding to the user's age with different formats which can be substituted by one column representing the age in year by converting the age of user in seconds to age in years and we can see from Fig. 2 the boxplot for the age and we notice the existence of outliers representing people

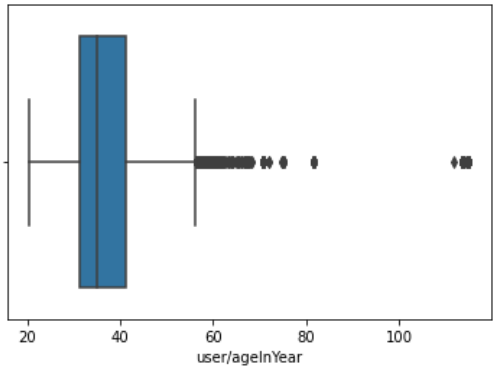

Fig. 2. Boxplot of user/ageInYear

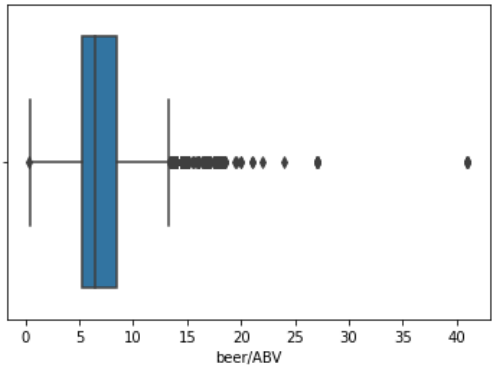

Fig. 3. Boxplot of beer/ABV

beer/name	beer/ABV
Witbier	NaN
Witbier	NaN
Witbier	4.6
Witbier	4.0
Dark Marc	NaN

TABLE II
CONSIDERED CASES FOR MISSING VALUES OF BEER/ABV

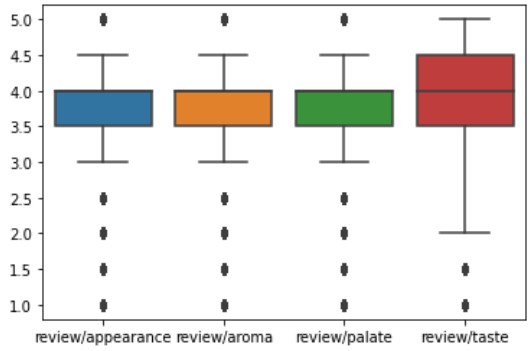

Fig. 4. Boxplot of Beer Reviews Characteristics

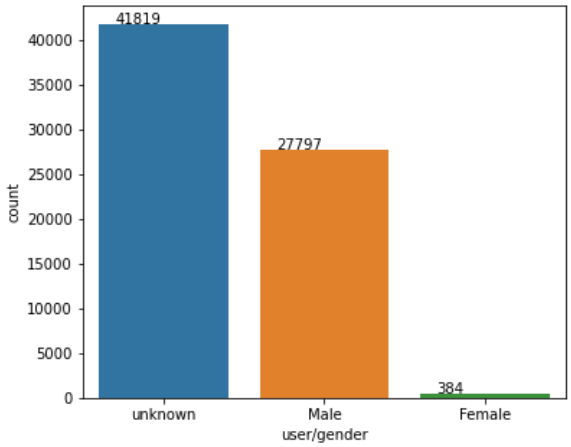

Fig. 5. user gender countplot

Variables	Cardinality
beer/name	14770
beer/style	104
user/gender	2
user/profileName	10573

TABLE III
CARDINALITY OF THE CATEGORICAL VARIABLES

with age older than 56 going till 115 which we decided to keep because we do not want to lose the data of users with older age. Moreover, although this feature contains 55,355 missing values as we saw from Table 1 which is almost 80% of the development set we still decide not to drop it because of the knowledge that all users have age over 20 years which is due to legal reasons so the mean was used to fill missing data. Also this feature has skewness of 3.466 so we applied log transformation which changed the skewness to 1.473 [1].

Another numerical variable that we consider is alcohol by volume of beer which is represented in Fig. 3 which shows outliers having high alcohol ratio from 11 to 41. Also we mentioned it having 3,107 missing values and as we can see from Table 2 we observe that for Witbier we have some with values and some missing so if this is the case we fill the missing ones with the mean of the Witbier ABV and if not like Dark Marc we use the mean of the whole dataset.

Then there are four review rating for each of appearance, aroma, palate, and taste with boxplots that can be seen in Fig. 4 ranging from one to five, which where kept without any modification.

Analyzing the user gender feature we notice from Fig. 5 that there is 41,819 missing values and there is two other values male or female which is highly imbalanced having a lot of males with respect to females and filling them with the majority will be misleading for some users which may affect the overall rating so we decided to drop this column.

We then consider the profile name of the user where it belongs to one of the features with very high cardinality of values as we see in Table 3 so using one-hot encoding would be very expensive memory-wise leading to 10573 columns,

so we decided to convert less frequent profile names to one category and the most frequent kept where one-hot encoding was applied to them and using 670 profiles was chosen because it led to best performance.

There are two features that identifies a beer which are the beer/name and beer/style but due to the very high cardinality of beer/name it was dropped and only beer/style was considered having only 104 unique value, to further reduce its dimensionality based on the face that there are much more frequent beer styles than others we renamed least frequent styles to other and considered most frequent 27 styles and applied one hot encoding on it.

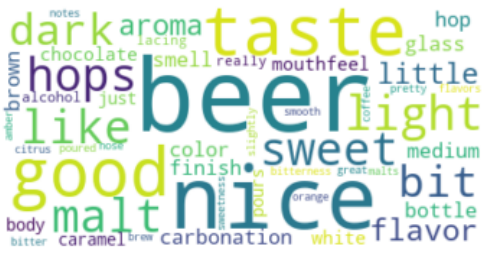

Fig. 6. User Gender Countplot

Finally, there is the text review for each beer that we need to process, so we used some feature extraction tool which splits each review into tokens and removes stop words, inverse document frequency was used to choose the words as shown in Fig. 6 and in order to avoid using too frequent or too rare words some parameters where used for this purpose (min_df=0.01, max_df=0.8) where all the words between the min_df and the max_df where considered which counts to 841 and only words with two or more characters.

B. Model selection

The following models were evaluated:

- **SVR**: This algorithm applies a transformation to the data which is in general non linear using a kernel function. It tries to find the hyperplane which maximizes the margin of errors between predicted and actual values that are tolerated. We chose to assess it because, in general, it is one of the best performing algorithm although it is computationally expensive.
- **Linear Regression**: Is a model that allows predicting the target variable value of new objects through linear relation with its features by minimizing the residual sum of squares between actual values and the estimated ones.
- **Random Forest Regressor**: Is an ensemble technique capable of performing regression tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees where each decision tree is trained on different portions of the training set and subsets of features.
- **RIDGE and LASSO Regression**: Regression analysis methods that perform both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. RIDGE simplifies the model by shrinking the size of some coefficients, while LASSO sets some coefficients to zero.

We avoided using polynomial models dues to the high dimensionality which is computationally very expensive to compute using function with high degrees like polynomial regression for example.

C. Hyperparameters tuning

We avoided using the grid search approach with cross validation due to the size of the dataset and the high dimensionality, so we decided to use the hold out method as observed in Fig. 7 where development set was divided to 3 sets training, validation, and test with sizes of 0.75, 0.15, and 0.10, respectively. The validation set is used to tune hyperparameters and the best parameters of each model are tested on test set to choose our best model after retraining on the union of the training and validation set.

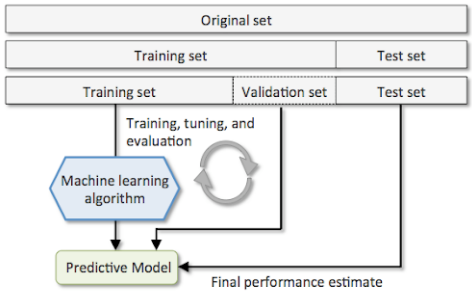

Fig. 7. Hold-out method used for model selection and hyperparameter tuning

We can observe in Tables 4, 5, and 6 the hyperparameters used for each model and their corresponding R^2 score.

Hyperparameters	R^2 score
C = 0.1	0.687
C = 1	0.705
C = 10	0.709

TABLE IV
HYPERPARAMETERS TUNING FOR SVR

Hyperparameters	R^2 score
n_estimators = 100 max_depth = None	0.684
n_estimators = 250 max_depth = None	0.685
n_estimators = 500 max_depth = None	0.686
n_estimators = 100 max_depth = 50	0.683
n_estimators = 250 max_depth = 50	0.686
n_estimators = 500 max_depth = 50	0.686

TABLE V
HYPERPARAMETERS TUNING FOR RANDOM FOREST REGRESSOR

Models	Hyperparameters	R^2 score
Linear Regression	fit_intercept = True	0.699
Linear Regression	fit_intercept = False	0.699
RIDGE	alpha = 1	0.699
RIDGE	alpha = 5	0.7011
RIDGE	alpha = 15	0.7018
LASSO	alpha = 0.001	0.680
LASSO	alpha = 0.0001	0.698
LASSO	alpha = 0.00001	0.701

TABLE VI
HYPERPARAMETERS TUNING FOR LINEAR REGRESSION, RIDGE, AND LASSO

III. RESULTS

We now compare the results of each model on the test set after choosing the corresponding best configuration from the previous tuning step, and we can observe the R^2 score in Table 7 for each model.

Models	Best Configuration	R^2 score
SVR	C = 10	0.716
RIDGE	alpha = 15	0.708
LASSO	alpha = 0.00001	0.707
Linear Regression	fit_intercept = True	0.705
Random Forest Regressor	n_estimators = 500 max_depth = None	0.697

TABLE VII
RESULTS OF EACH MODEL USING BEST CONFIGURATION SORTED BY DECREASING ORDER OF R^2 SCORE

As we can see SVR with configuration C=10 had the best performance between all the evaluated models with an R^2 score of 0.716 while the Random Forest did the worse even different configuration was not effective like increasing n_estimators was not worth considering the extra computational time and it hard worse result even compared to Linear Regression which is the simplest model and all these results are from the test portion of the development set. Also for Ridge and Lasso the scores are quite good compared to SVR considering the big difference in computational time.

Regarding the public score performance, the predictions using SVR which is the best model have been submitted to the leaderboard achieving an R^2 score of 0.718 and as expected predicting using other models gave lower scores. The results on the public score had small difference close to 0.02 compared to the test portion of development set of the models considered.

IV. DISCUSSION

The proposed feature engineering approach led to the best performance using SVR while it was quite interesting the fact that Random Forest resulted in worse predictions than Linear Regression and we can see in Fig. 8 that the most important feature used was review/taste with importance more than the sum of all other features. It is important to mention that maybe different hyperparameters would have improved the model but the computation time due to the resulting huge amount of elements prevented it. Also some techniques like truncated SVD may have been used to reduce dimensionality but was avoided in order not lose the sparsity by moving points with binary dimensions with linear projections.

Some other approaches for the text columns could have been used to improve the performance using a good tokenizer or a stemmer or lemmatizer, and maybe instead of using only unigrams consider bigrams or even trigrams [2] with care of number of elements which can be tuned to avoid having a very large number of columns. Also something like number of characters for each instance could have been added, noting that number of words was tried but with no further improvement so it was not used. Moreover, the beer name and style could have been further preprocessed considering the existence of some frequent terms used like "Ale" and "IPA" which where common in both, also since the quality have specific values between one and five and halves being included we tried to round the predicted values to the closest possible values but it did not lead to better results. The used approach led to quite satisfactory results with less complex preprocessing techniques that we mentioned which would have probably improved our model.

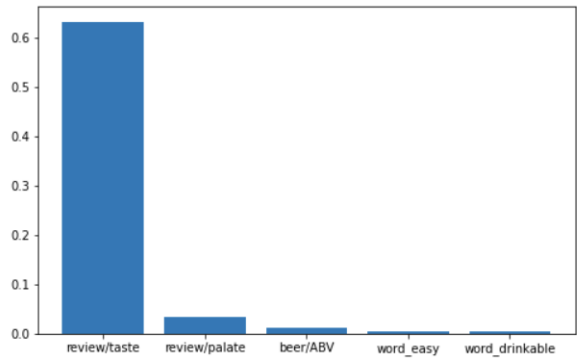

Fig. 8. Feature Importance by Random Forest

REFERENCES

[1] A. Zheng and A. Casari. (2018) Feature engineering for machine learning principles and techniques for data scientists. [Online]. Available: http://oreilly.com/safari.

[2] R. B. Benjamin Bengfort and T. Ojeda. (2018) Applied text analysis with python. [Online]. Available: http://oreilly.com/safari.