Wine Reviews Dataset Regression Problem

Khudayar Farmanli
Politecnico Di Torino
s276310@studenti.polito.it

***Abstract*—In this report we introduce our approach to *Wine Reviews Dataset* regression problem. Particularly, some operations done to fill the missing values and remove duplicate records before encoding categorical features and finally regression model is applied. The proposed model performs good score in evaluation board and obtains overall satisfactory results .**

I. Problem Overview

The proposed competition is a regression problem on the *Wine Reviews Dataset*, a collection of some logistical information and review to particular wine expressed by both categorical and numerical attributes. The dataset is divided into two parts:

- a *development* set, containing 120,744 entries with numerical *quality* feature as a label.
- an *evaluation* set, containing 30,186 entries without *quality* feature.

We will use development set to build regression model and predict the quality labels for evaluation set.

One of the most important actions is to understand the data we will use. We are working with high cardinality categorical variables and for making dataset suitable to train the regression model we should pass through some very important steps. Understanding of data will provide the ideas to take relevant measures.

Quality column does not contain null value and as we can see in Figure 1, the values distributed normally between 20 and 80. We have 8 categorical attributes. Figure 2 shows the proportion of null and non-null values per attribute. 30 percent of *designation*, 17 percent of *region1* and 60 percent of *region2* is null. High proportion of missing values in *region2* make it useless for training dataset. On the other hand , we can consider some measures for *designation* and *region1*. Figure 3 shows the proportion of unique and non-unique values per attribute and especially in *description* column the presence of 30 percent non-unique values has specific importance. Because every single row of it contains comment written by reviewer in particular and in this case the percentage shows duplicate records. Of course, repetition means all values in that row belong to particular attribute are also duplicates but generally, repeated values are not considered unusual case other than *description*.

Use of categorical variables directly in regression model is not the case. We will use proper encoding technique before training and testing the datasets.

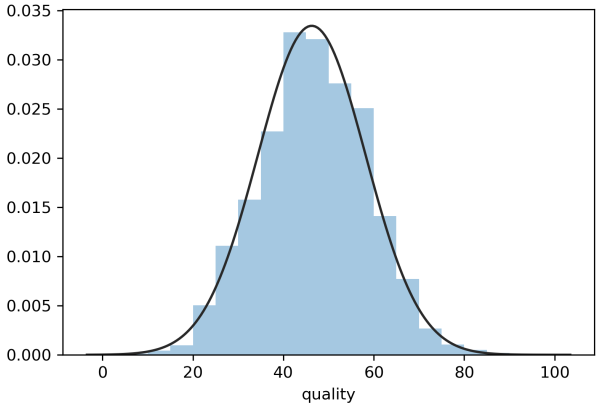

Fig. 1: Distribution of *quality* values

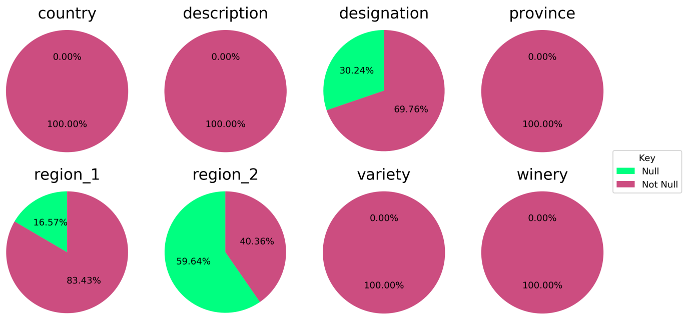

Fig. 2: Proportion of Null and Non-Null Values in each column

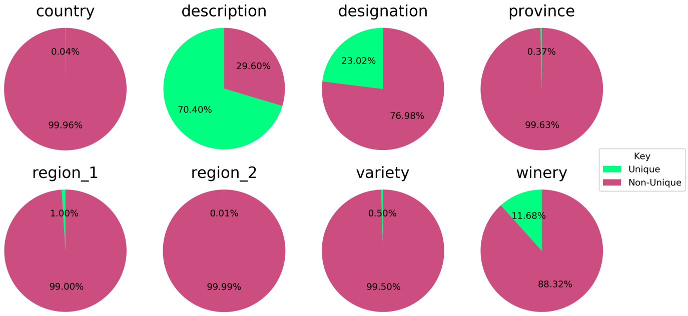

Fig. 3: Proportion of Unique and Non-Unique Values for Categorical variables

II. Proposed Approach

A. Data Preprocessing

We are not considering *region2* for final training set because it contains high number of missing values. On the other hand, we can use hierarchical order between *province* and *region1* to fill the latter. Also, missing values in *designation* can be solved in the same way, this time based on *winery*.

- We iterate through all the *province* values and fill the rows particularly with the highest seen *region1* value. If there is not even single region for particular province (and this is often the case) then 'NoListedRegion' is filled.
- Exactly same process applied to *designation* based on *winery* and 'NoListedDesignation' is filled in similar case.
- Also, there are only 4 missing values in *country* and *province* columns and some other small exceptions which are not worth to mention. We drop them because of very small importance.

We apply first two operations to both *evaluation* and *development* datasets for convenience (in other cases only to development dataset). Then we drop duplicate records for *description* column. But we are using intersection of description attribute with other categorical attributes for the sake of not losing their values, if at least one of them is unique particularly, even though there are only few of them. This step create 35,000 decrease in *development* dataset. In figure 4 we can see the result of dropping duplicates. The final step is encoding categorical features of both *evaluation* and *development* datasets. We use *OneHotEncoder* for encoding categorical attributes other than *description*. It is encoding categorical features as a one-hot numeric array. *TfidfVectorizer* is used for *description* column which converts a collection of raw documents to a matrix of TF-IDF features.

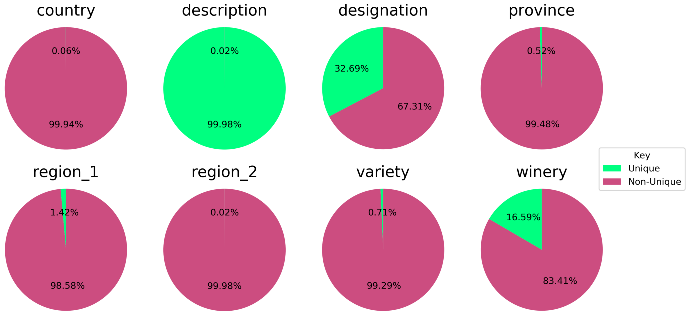

Fig. 4: Final look of attributes after dropping duplicate records.

B. Model selection

We have tested *Ridge* regression model for our case. Ridge is a variant of linear regression analysis method (just with penalty) that performs both variable selection and regularization with *L2* norm vector in order to enhance the prediction accuracy and interpretability of statistical model it produces. The coefficients of features which are not useful for the regression assigned closer to zero. They converge to zero but never can be zero. This process is very useful for decreasing model complexity and preventing overfitting.

Other than Ridge we also tried *LinearRegression*, *RandomForestRegressor* and *LogisticRegression* but non of them gave results closer to Ridge, that is why, we do not consider their results worth to mention.

III. Results

We let all the hyperparameters remain as default except *alpha* which is regularization parameter and affect the results even in small change in number. The value of alpha manually defined as 0.35 by doing run the algorithm every time with different number of alpha and check the *r2* score for train and test split of *development* dataset. We have obtained 0.93 r2 score locally which is really good result and the public score obtained from competition is 0.825 .

IV. Discussion

The result obtained by proposed approach outperform defined competition baseline r2 score which is 0.436, almost two times greater. But, indeed, some further steps can be considered to increase the score to some extent.

The following aspects might be useful:

- Some other regression models can be considered. For instance, *Lasso* regression (least absolute shrinkage and selection operator) which is similar to Ridge. The main differences are Lasso set some coefficients to 0 and it uses *L1* regularization parameter.
- Unfortunately, we have not done *GridSearchCV* which is used for tuning the hyperparameters. Because, it is really computationally hard task to do. If this step done then we can get better results. We can tune for example, *alpha*, *solver*, *max-iter* parameters.

Despite we have mentioned some ways to improve the result, obtained one is very promising score and there is not much room for further improvements.