Machine Learning Project Report - Regression Task

Garance Haefliger, Salomé Thueler

December 2020

1 Introduction

In this project we aimed at predicting the perceived pleasantness of the scent of a variety of molecules based on their molecular properties for example their molecular weight. This is a typical regression task and we will approach this problem using both linear and non-linear approaches. Throughout the whole project we mainly used the Root Mean Squared Error (RMSE) as a criterion for the comparison between methods.

2 Process and Results

2.1 Exploration of the data

A preliminary exploration of the data allow us to get a first ensemble view of the main characteristics of the data set. To predict the pleasantness of the odor of a 708 molecules, we were provided with 4870 predictors. These predictors are both numerical and categorical and the latter will be transformed into numerical data. Note that since we did not use one hot coding for the categorical data to avoid adding a "new predictor" to the model, instead we just transformed low/high into 1/2. We also observed that pleasantness was ranked as integers between 0 and 100 (even though the maximum values is 98) and that it is approximately normally distributed. We also observed that among the 4870 predictors, 1842 have a null variance, meaning they can not bring useful information to our predictions and we remove them before constructing our models. Even though some predictors are correlated we see that removing them from our data set does not increase performance in our models, thus we decided to leave them for easier reproducibility with test data and as a safety. Preliminary pairwise visualisation reveals that pleasantness seems linked to sweetness/sourness, however this categorical predictor is not provided in test data and we will not use it as a predictor. We also tried to perform PCA with the data in order to visualize intrinsic data structure that could be helpful but, unfortunately, no clear structure was visible when projecting the first two principal component. This is however consistent with the observations of data normality on the histogram. To summarize we observed the need to transform categorical data and remove predictors with null variance.

2.2 Linear Methods

As a baseline model, we performed a simple linear regression on the data set with all the predictors whose variance is not null and we obtained an extremely high test error. In order to decrease the test error, we chose another method : Lasso regularization. We cross-validated on different lambda parameters (the strength of the regularization) to choose the one with the best performance on our model. We obtained a test RMSE of 22.57. We repeated this procedure similarly for Ridge regularization, which leaded to a test RMSE of 22.50. Thus, by comparing these two methods, we found that Ridge method gave slightly better results than Lasso method but overall both are quite similar. Since we observed in the exploration part that some predictors seemed to be highly correlated, we used bootstrap method to see if we could improve the test RMSE of Ridge regularization. We got slightly better results depending on the run and on the training set, but generally not significantly better on the test set. This is consistent with the fact that bootstrap is especially efficient with heteroscedastic data. We also tried to implement forward selection combined with cross-validation to select the best linear model, but linear dependencies were found and we thought that this method was not suitable.

2.3 Non-Linear Methods

2.3.1 Neural Networks

Our first approach using neural network, was to design a simple model with 64 units in the first layer, followed by 2 hidden layers with relu activation and a final layer with linear activation to recover the response. We experimented with the model, assessing its quality with the validation set approach. By comparing the validation and training RMSE and loss, it seemed that the model was over fitting the data since the training RMSE was too small compared to the validation RMSE. To try and overcome this issue we introduced dropout layers which randomly drops out neuron links to prevent over fitting (note that this part of the code was not left in the final script).
Next to have a better estimate of the actual RMSE we introduced the 10 fold cross-validation approach where the neural model is fitted to each 9 folds alternatively, evaluated on the remaining 10th and the resulting RMSE averaged across the 10 fits. This enables us to test the performance of different additional features with a more reliable error criterion. For instance we compared the cross-validated RMSE of the simple model with l1 regularization added in every layer to the simple model with early stopping. We also investigated the results for the model with l2 regularization and the best results for all of these trial is to use early callback. The early callback is a regularization on the epochs of the model where, once the validation loss is increasing (the model begins to overfit the training data), the fit is stopped. From this procedure we see that the best results are achieved when using L1 regularization on all the layer as well as early callback and the RMSE obtained this way is 17.95.
Note that we did not choose this model for kaggle submission but that we would have used this model fitted on the whole data set if had wanted to. Overall the design of a neural network is a complex task and finding the optimal network for the data set is a project on its own thus we focused on more consistent approach such as trees and boosting.

2.3.2 Trees

First, we performed a simple tree regression taking into account all predictors whose variance is not null in order to have a baseline tree model. We found a test RMSE of 27.50, which is quite high. Thus, we knew that there was the potential to improve. We tried cross-validation on pruned trees but the test error was not better. Then, we tried bagging which gave us an RMSE of 22.75 but we decided to improve it with random forest methods to decrease the test error. In order to increase performance of the random forest, we performed cross-validation to tune the m parameter (that is the level at which the tree is "cut") and we found that the best result was with m= 108. With this method we improved the test RMSE and we decided to do a kaggle submission.
Then, we chose to do gradient boosting method because it usually get more efficient results than random forest or bagging. The first approach was to select the lambda parameter (shrinkage parameter) that gave the lowest test error : λ=0.00316. Note that we used a validation set approach instead of a cross-validated one because of the running time since the range was very large. The found test RMSE was 22.426, which is a relatively good result. Thus, we decided to find a way to still improve this method. We did a cross-validation with the best lambda to select the optimal number of rounds : 1012. Then, we computed a boosting model which combined the best lambda and the best number of rounds. We achieved to improve a little the test RMSE : 22.422. Eventually, we tuned the maximum depth of the tree (max.depth) in the boosting model. In the previous models, we used max.depth of 3 but we wanted to verify this choice by cross-validation. We found that the optimal max.depth between 1 and 10 for the model with the optimal lambda and number of round was indeed 3 which could be expected since we tuned the two other parameters lambda and nround with a max.depth of 3. Finally, we tried to improve the prediction on the test data by computing this model on the whole training data and we submitted it to kaggle.

3 Conclusion

To summarize, we tried to predict the smell pleasantness of molecules based on their properties using linear and non-linear machine learning methods. The results with the non linear one did not improve as much as we hoped compared to the non-linear. Based on our attempts, we recommend to use Ridge regularized regression for an efficient and quick method, and boosting for a more elaborated and precise one but longer computation. Ultimately, we can not expect to perfectly predict the score because of the biased individual perception of smell.