Cyber Brista - Final Report

Jiayu Zheng

Brown University, DSI

https://github.com/BubbleJoe-BrownU/hands-on-machine-learning-project

DATA1030 Hands-on Data Science

Introduction

Many people love coffee. They enjoy to be energized by the effect of caffeine and the joyful scent of mixture of milk and espresso. But loving drinking coffee doesn't mean one have the expertise on choosing qualified coffee beans. We aim to develop our Cyber Barista, a customer-oriented, regression model featuring predicting quality scores of coffee beans based on limited information. By mentioning limited information we want to emphasize that even a customer who can only recognize information on the package of beans will be able to use our model. By meliorating user experience and lower entry barriers, Cyber Barista bears the potential to enlarge the coffee market to an unprecedented level.

First, we found a dataset on kaggle[1], which scraped data from the review pages of Coffee Quality Institute in January 2018[2]. There are not much previous work on it. Thupsaeng[3] performed coffee classification on the dataset with random forest classifier and RDF. Kaymonov[4] performed data visualization and quality prediction on it. Reza[5] performed random forest classifier on it. Because not much people are interested in this topic, this dataset is hardly described. Here is a brief manual description of the dataset made. Features in this dataset can be roughly divided into four parts according to four parts of the grading report: sample information, cupping scores, green analysis and certification information[6]. Sample information are metadata about the origin of coffee beans, such as country, region. Cupping scores contain 10 individual scores as well as a total score on a scale of 100. Green analysis[7] contains metadata of coffee beans itself, such as moisture and defects. And certification information is about certification information about the grader, such as expiration date and certification address.

Table 1 Feature types and the number of features of each type

Feature Type	**# of features**	**Feature Name**
Continuous	21	altitude, altitude low meters, altitude high meters, altitude mean meters, number of bags, bag weight, aroma, flavor, aftertaste, acidity, body, balance, uniformity, cup cleanliness, sweetness, cupper points, total cup points, moisture, category one defects, category two defects, quakers
Categorical	22	species, owner, owner_1, country of origin, farm name, lot number, mill, ICO number, company, unit of measurement, region, producer, in country partner, harvest year, grading date, variety, processing method, color, expiration, certification body, certification address, certification contact

EDA

The target variable for our project is the total cup points of each record, which shows the overall quality of a combination of individual properties such as aroma, aftertaste, etc. The dataset contains 1311 rows of records and 43 columns of features, among which 19 features have null values, with the highest null value rate reaching nearly 80% and mostly around 20%. The target variable is a numeric variable, which distributes around 83 and has a wide distribution range.

EDA plots

Here are some visualized exploratory data analysis results of various types.

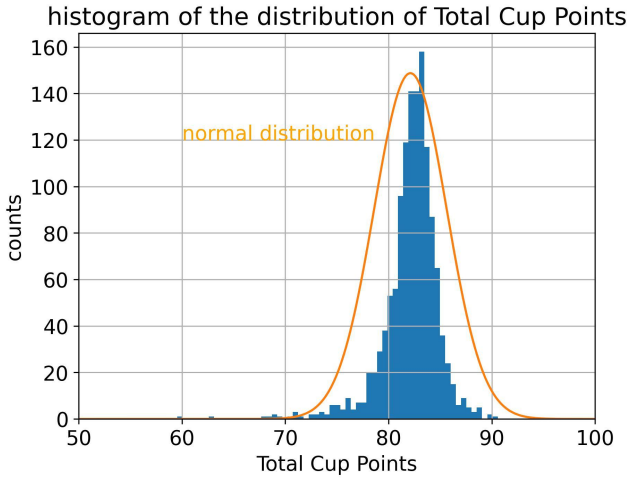

Figure 1 the distribution of our target variable - total cup points. As shown in the plot, the distribution of the target variable roughly aligns with the normal distribution and it doesn't have a heavy tail. So the distribution of our target variable is balanced, requiring only the basic split in data splitting part.

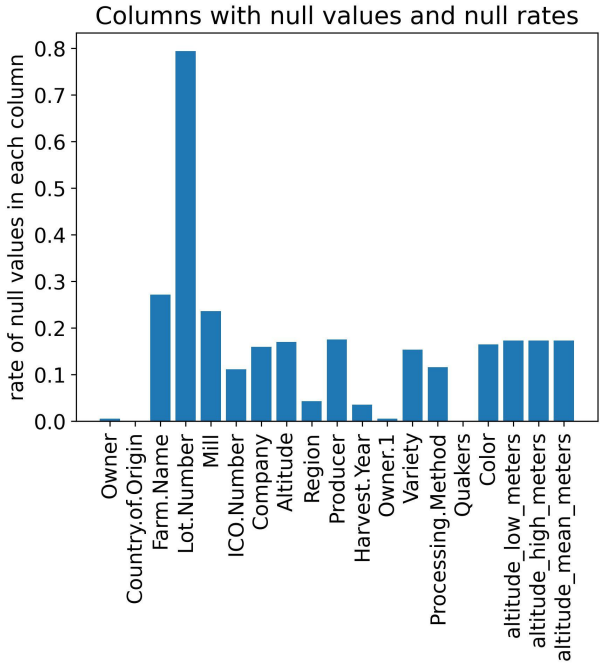

Figure 2 Columns with missing values and the corresponding null value rate. There are 19 features out of 43 in total having missing values, with the maximum missing value rate around 80% and most of them around 20%.

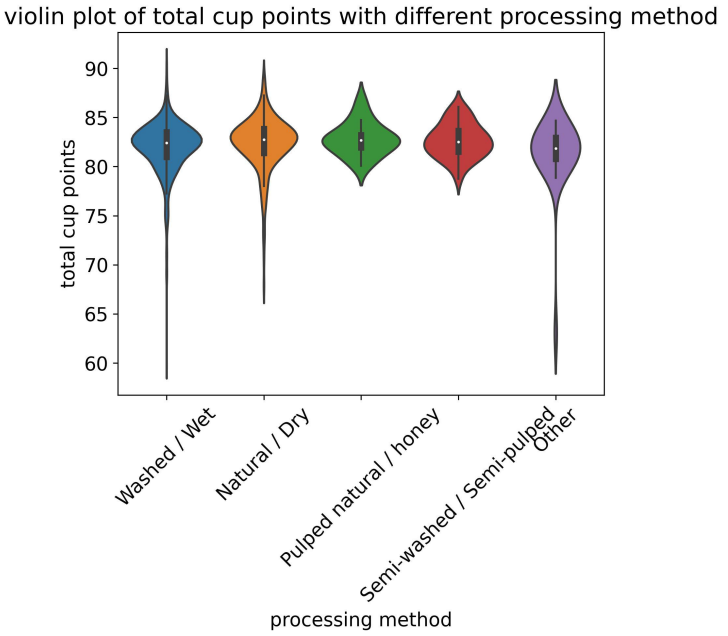

Figure 3 Distribution of total cup points (target variable) with different processing methods.

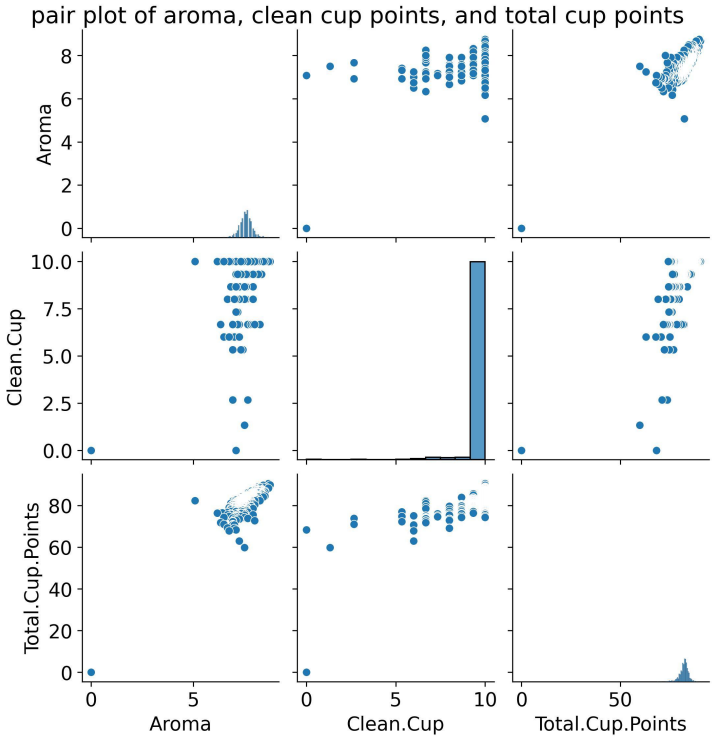

Figure 4 pairwise scatter and histogram plots between Aroma, Clean Cup points and Total Cup Points. Aroma points highly correlates with the total cup points; but clean cup points is not strongly correlated with either aroma or total cup points. The result shows that some individual points, like Aroma, are also candidate target.

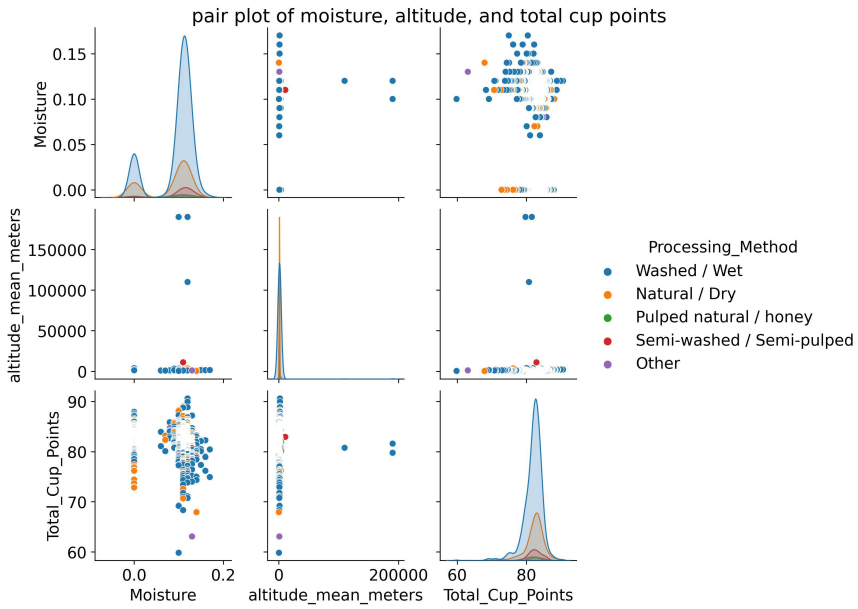

Figure 5 Pair-wise plots of total cup points, mean altitude, and moisture, with the color of points denoting the processing methods. The distribution of moisture is roughly at 0 or around 0.1; most of the coffee beans are from farms located at a altitude lower than 2000 meters.

Methods

Feature selection

In order to make our model accessible to ordinary customers, we have to be careful about the number and types of inputs we use. We remove features that are irrelevant to the prediction or that only contain company-level information that customers don't have access to. For example, we eliminate pure textual data containing only identity information because they are not strongly related to coffee quality. We also remove features that provide too much detail about the origin of coffee beans, as well as features that don't help us predict coffee quality, such as grading details and CQI. After removing unnecessary features, we are left with 28 features, 11 of which relate to scores. We filter out the species because all of the beans are Arabica, and we eliminate the Harvest Year because coffee quality is stable over time. We keep only the total cup points from the score features, and we use a unit of measurement to standardize the units of the altitude data. After these steps, we are left with 6 features: Country of Origin, Mean Altitude, Variety, Processing Method, Moisture, and Color.

Data splitting

The distribution of the target variable is quite balanced as shown in the plot below, so it’s enough just to do basic split using sklearn train_test_split method. Also in the cross validation phase, because this dataset is of a small size, we considered using a k-fold split to get a robust averaged model. We split 20% of the data as the test data, then performed the 4-fold split on the remaining data to get the training and validation data. The split ratio of training, validation and test data is 0.6:0.2:0.2.

The dataset is independently identically distributed because the data is randomly sampled, and teach observation of target variable are not correlated to the next one. The dataset don't have group structures and is not a time-series one. Thus, it's ideal to just use regular split and k-fold split for cross validation.

Preprocessing

For categorical features, we chose to use SimpleImputer with constant fill to deal with categorical missing values and use OneHot encoder to transform them. For null values of continuous features, since we don't want to fill them with default value for fear of contaminate its statistical attributes, we simply dropped those rows with missing value for normal machine learning models, and for XGBoost regressor we just kept the missing values since it will handle missing values properly. After that, We use StandardScaler to transform the continuous featuresg. After preprocessing, there are 74 features in total, with two of them continuous features, and the rest of them one-hot encoded categorical features.

Algorithms & pipeline

In this project, we thoroughly studied the coffee quality dataset with 8 different models, they are Linear regression, Ridge, Lasso, ElasticNet, Random Forest regressor, Support Vector regressor, K Nearest Neighbor regressor, XGBoost regressor. We use negative mean squared error as the training score and use mean squared error to select best models.

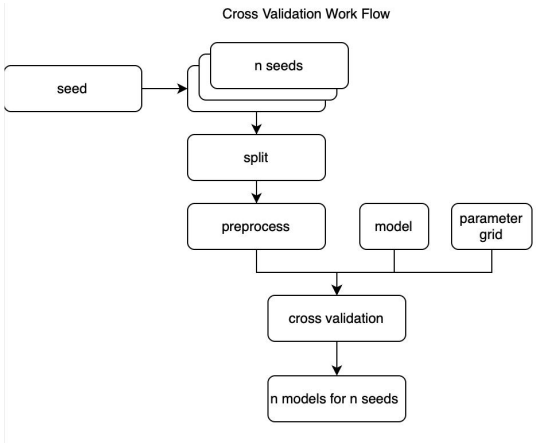

Figure 6 A brief flow chart of cross validation

In this project, I build a modular pipeline combining random state generating, data splitting, data preprocessing, hyperparameter tuning, test set inference and result saving altogether. For XGBoost regressor I implement a manual grid search pipeline, for the rest of models I just used the sklearn GridSearchCV method. I performed two rounds of grid search in the project in order to get a fine-tuned model. In the first round, for each type of model, n random states were generated under a fixed seed. For each random state, I performed grid search with parameter space size of 100 roughly, except for the linear regression which don't have parameters to be tuned. After the first round, I found XGBoost regressor was the best model type. So I performed the second round of grid search only for XGBoost regressor, with the parameter space size of 1250. The parameter space explored in this project are shown below.

Table 2 Parameter grid for each mdoel

Model	Parameter	Parameter range	Size of grid
Linear regression	None	None	1
Ridge	alpha	[0, 1000], 100	100
Lasso	alpha	[0, 1000], 100	100
ElasticNet	alpha l1_ratio selection	[0, 1000], 20 [0, 1], 5 random	100
KNN	n_neighbors weights p	[1, 50], 25 Uniform, distance 1, 2	100
Random Forest	max_depth max_features	1, 100, 20 0.5, 0.75, 1.0, sqrt, log2	100
SVR	kernel C gamma	Linear, poly, rbf, sigmoid 0, 100, 5 Scale, auto, 0.01, 0.1, 1	100
xgboost	learning_rate reg_alpha reg_lambda max_depth colsample_bytree subsample	[1e-2, 1e1], 5 [1e-1, 1e2], 5 [1e-1, 1e2], 5 [1, 100], 10 0.9 0.66	1250

In order to perform a quantitative analysis on the model performance, I chose MSE, MAE, RMSE, and R2 score as the metrics, among which MSE was used to train the model, and decide the hyperparameter and analyze feature importance. Specifically, negative mean squared error was used as grid search score because it straightforwardly adds penalty to the deviation of predicted value off the original values, and compared to MAE or RMSE, it applies larger penalty when predicted values differ too much from labels, which accelerates convergence.

Here we calculated the mean and standard deviation of n models for each model type and plot them in the same error graph.

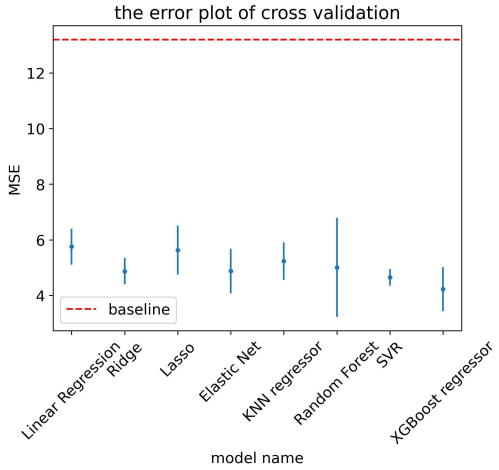

Figure 7 Baseline and model loss of seven model types. This graph shows the uncertainty brought by the random state and some randomness intrinsic to models themselves. According to the mean value of

each model type we can see that all models outperform the baseline significantly. Among these models, random forest regressor has the highest standard deviation because of its randomness of trees, and linear models also have high deviation probably because their oversimple architecture is vulnerable to local minimum.

Results

Predicting performance

The MSE of our models are no more than 5.756, which is achieved by linear regression. The lowest MSE, 4.227, is achieved by XGBoost regressor, which is our best model in this project. All of our models outperformed the baseline MSE, 13.20, meaning that they fitted the dataset quite well. We also calculated the standard deviation of MSE between our models and the baseline.

Table 3 Baseline errors and model error

Model	MAE	MSE	RMSE	R2 Score	MSE reduce(%)
baseline	1.846	13.20	3.633	0	0
Linear Regression	1.650	5.756	2.399	0.1808	56.38
Ridge	1.613	4.875	2.208	0.1764	63.06
Lasso	1.686	5.632	2.373	-9.25E-06	57.32
Random Forest Regressor	1.611	5.012	2.239	0.1533	63.01
SVR	1.517	4.651	2.157	0.2141	62.02
KNN Regressor	1.586	5.238	2.289	0.06999	64.75
xgboost	1.492	4.227	2.056	0.3044	60.31

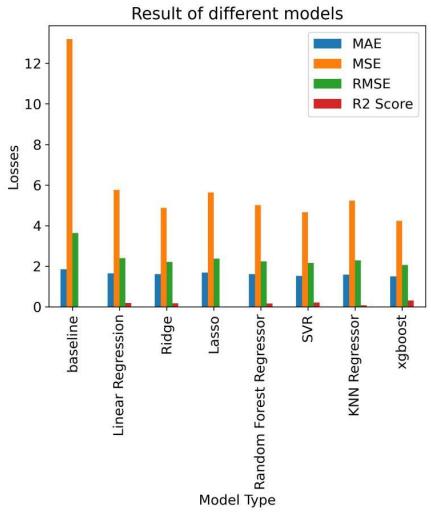

Figure 8 Baseline and model loss of seven model types.

Global feature importance

We calculated the feature importance in three ways in order to interpret our best model, XGBoost regressor. We performed per-feature perturbation, XGBoost built-in feature importance and SHAP to interpret the feature importance.

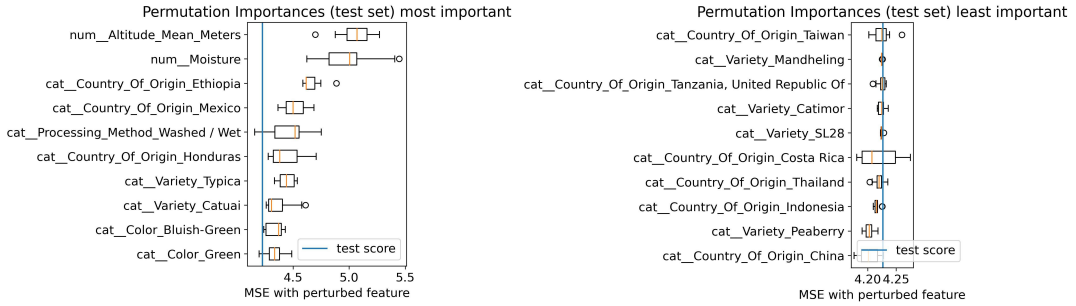

Figure 9 Top 10 most and least importance features via feature perturbation.

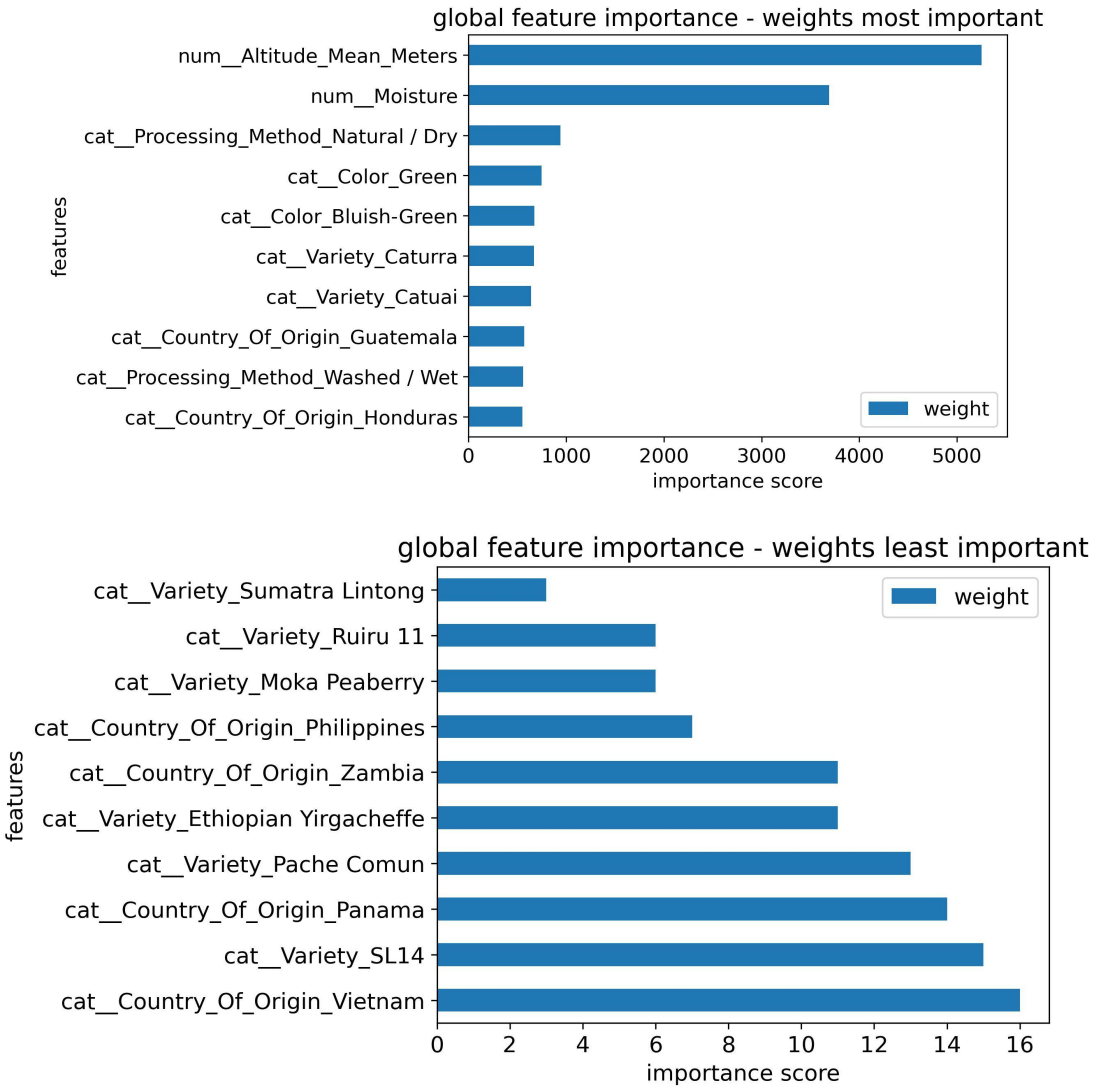

Figure 10 Top 10 most importance features via XGBoost feature importance with importance type weight. The most pronounced two features are altitude mean meters and moisture.

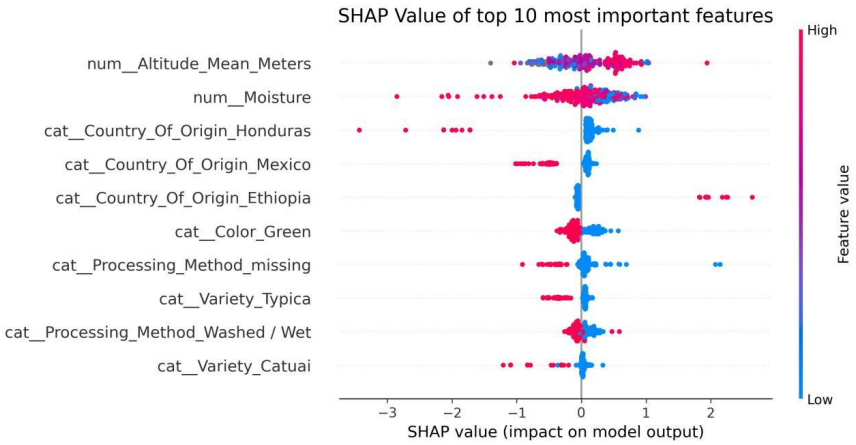

Figure 11 Top 10 most importance features SHAP. The most pronounced two features are altitude mean meters and moisture.

Overall, all three methods of global feature importance give highest credits to the only two continuous features, altitude mean meters and moisture, which might be due to that altitude strongly affect the climate in which coffee bean grows up in and moisture indicates how well beans are processed and kept. It's noteworthy that three countries, Ethiopia, Mexico, Honduras, which is well known for producing top coffee beans are successfully identified by our global feature importance method, XGBoost weight and SHAP. This shows our model successfully "understand" the key factors affecting the quality of coffee beans.

Once we have the most important features, we can plot our true and predicted target variables along with them in a plane in order to visualize the prediction performance. In these plots we can see that the predicted target aligned well with the true target, indicating successful fitting of our model in the dataset. Also we can find an interesting truth, which is resulted from our assumption when performing regression that simple model fits better, that predicted targets are in more condensed clusters than the true value. This is because the model didn't fit the noisy pattern in the dataset such as the variance between data points.

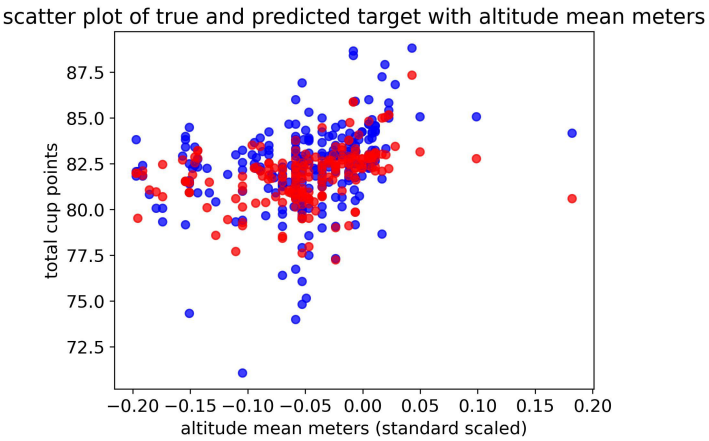

Figure 12 scatter plot of true and predicted target variable along with altitude mean meters. Altitude mean meters is among the top 2 most important features. And the plot shows a good alignment between the true target and the predicted target.

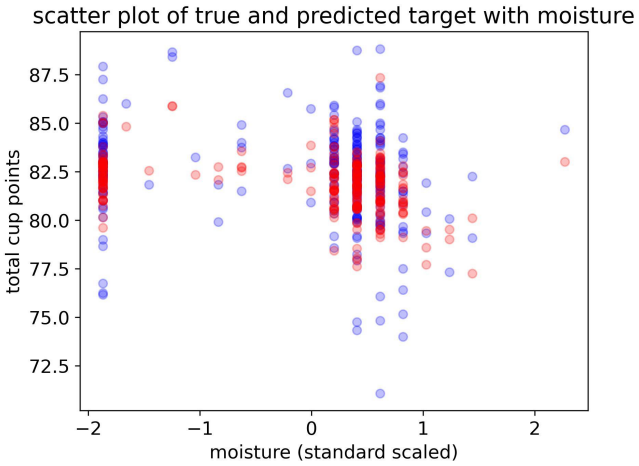

Figure 13 scatter plot of true and predicted target variable along with moisture. Moisture is among the top 2 most important features. And the plot shows a good alignment between the true target and the predicted target.

Local feature importance

Later on, I performed local feature importance on individual data points. We randomly sampled ten data points using the test set of best XGBoost regressor. Here we show two SHAP force plots below. Comparing the two plots we can see that altitude mean meters has a positive contribution to the prediction if its value is above the mean, negative if not.

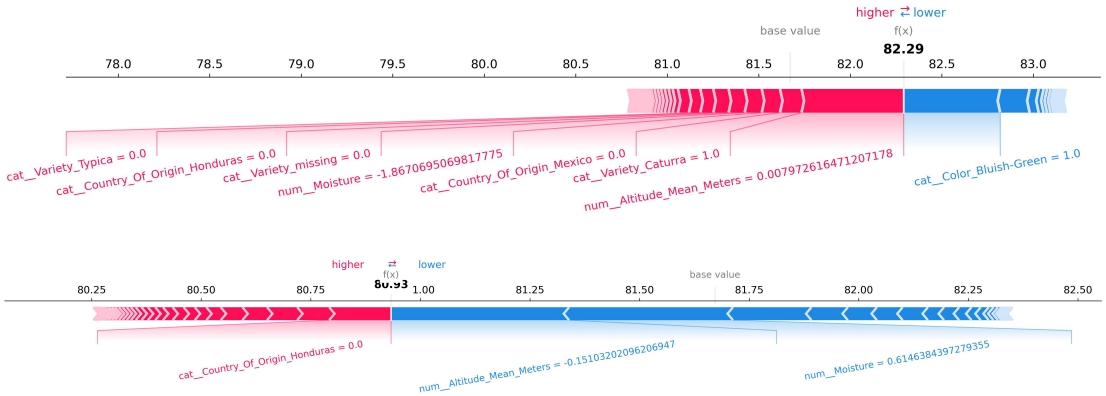

Figure 14 The SHAP force plot of two randomly sampled individual data points in test set. For the first data point, a slightly above average altitude mean meters and a pretty low moisture have positive contributions to the prediction. Also the bean variety Caturra also pushes the prediction to a higher value. For the second data point, a low altitude mean meters and high moisture both have negative contributions to the prediction

Outlook

The mean squared error of our models are no larger than 5.756, much lower than the base line error 13.20, indicating that our model has successfully fit the dataset. But the lowest mean squared error, achieved by the XGBoost regressor, is 4.227, indicating an average 2.06 points deviation of the total cup points compared to the labels. The imperfection of our model can be mainly attributed to the limited size of data set, limited number of continuous features, and perhaps not yet refined machine learning techniques. The relative small dataset size can make the prediction

suffer from the curse of dimensionality[8], making the data space saturated with sparse data points, strongly cripple model performances. We could further improve our model performance should we made progresses in solving them.

Although the machine learning algorithms we covered in studying this dataset seemed to be enough for the task, but we would expect better performance if more advanced models are applied. Just think of the extraordinary performance of XGBoost regressor!

References

[1] https://www.kaggle.com/datasets/volpatto/coffee-quality-database-from-cqi
[2] https://github.com/jldbc/coffee-quality-database
[3]https://www.kaggle.com/code/aphisitthupsaeng/coffee-classification-with-knn-rdf-visualization
[4] https://www.kaggle.com/code/nikitakaymonov/data-visualisation-and-predicting-the-quality
[5] https://www.kaggle.com/code/rezaif/arabica-random-forest-classifier
[6] https://database.coffeeinstitute.org/users/icps/arabica
[7] https://sca.coffee/research/coffee-standards/
[8] https://en.wikipedia.org/wiki/Curse_of_dimensionality