ALGORITHMICS FOR DATA MINING

Usage of Knime for the prediction of the wine quality

Henry Qiu, Goktug Cengiz

Contents

1 Introduction

In this project we will present a usage of Knime, a data analytics, reporting and integration open source platform, that provides tools to organize and make easier the management of machine learning and data mining processes. We will use Knime to perform several data mining processes over a chosen data set, our aim is to give a detail explanation of the usage of Knime to perform these tasks, and to evaluate and interpret the results obtained.

To carry out this project, we have followed the **CRISP-DM** *(Cross Industry Standard Process for Data Mining)* method, that is a standard process model that describes common approaches used by data mining experts in industry companies. CRISP-DM breaks the process into six phases, these phases can be represented in the scheme in the Figure [1].

These phases are: business understanding, data understanding, data preparation, modeling, evaluation and deployment.

In the business understanding phase, we have to determine the objective of the whole project, in our case, the topic is related to the quality of wines, so our project's aim would be determining which are the factors that affect more wines' quality, so that we can focus on improving that factors to produce wines with better qualities. The business objective will be improving the quality of wines so that wines will have better sales and the enterprise will get more benefits.

Data understanding consists in knowing the available data, the meaning of the attributes and the features of these attributes. In our case these information is provided by the source of the data set.

Data preparation consists in processing the data so that they are prepared for the next processes. generally, when we have big amount of data, it is common to have some missing values or weird data that doesn't fit the expectations, these data seems to behave different to other data we have and it is necessary to threat them.

One of the most important step is to make the model for our data, thanks to the model, we will be able to make different tasks, as predictions, classifications, and so on. Therefore, it does not exist a correct model, there are only models more adapted to our aim than others, so in our project, we will need to create the model so that the quality of wines can be predicted the best possible.

Once we get the results, in the evaluation phase we have to analyze and interpret them, we will need to know what does the results mean so that we can take conclusions on what will make the wines have better quality. Without understanding the results they are just piece of numbers.

At the end, once we take the conclusions, the enterprise should take a business plan to apply these knowledge to see whether this project have contribute a clear improvement for the development of the enterprise.

However, this process can be cyclic, for future improvements, more analysis over wines can be done, for example knowing the type of grapes used with different features of the wines, etc, and this process would be carried out again.

2 Data Understanding

The data set chosen is the **Wine Quality Data Set** provided in the *Machine Learning Repository of the UCI* (`https://archive.ics.uci.edu/ml/datasets/Wine+Quality`). This repository was created with the goal of modeling wine quality based on physicochemical tests. The data set is multivariate and the attribute characteristics are real, the data set can be viewed as classification or regression tasks. Concretely, two data sets are included in the repository: red wine quality data set and white wine quality dataset. Taking into account that in different type wines' the quality can be determined by different factors, we have decided to focus on only one of them, that will be the red wine quality data set.

We have to do the data understanding phase, apart from the brief description of the data set in the repository, we can get the detailed information of the data with Knime. The data set is in a csv format, to extract the information of the data set we can use the tool CSV Reader of Knime. This tool allow us to import the data set to the system with the given path of the file, and extract general information about the instances and the attributes. We can see the general of the data set in the Figure [2].

We can see that the number of instances of the data set are 1599, there are 12 attributes: Fixed acidity, Volatile acidity, Citric acid. Residual sugar, Chlorides, Free sulfur dioxide, Total sulfur dioxide, Density, PH, Sulphates, Alcohol and Quality. All of them are numerical variables, concretely, quality are integer numbers, while the rest of the variables are double types.

With the information provided by the repository, we know that the last attribute *Quality* is the output attribute, that will be the objective of the evaluation of the processes. This attribute will be a number between 0 and 10, being 10 the highest quality. The rest of the variables represent different components or properties of a wine, the meaning of these concepts belongs to a domain that in our case we are not experts, maybe this will have effect in the interpretation of the results obtained, applied to the reality, there should be an expert who have enough knowledge of the domain able to analyze the results properly.

After the reading of the data, Knime provide more tools to make easier the comprehension and extraction of information from the available data. The statistics tool generates some tables and plots that contain statistical information about the data set. One of the most interesting is the statistics table, that provide us important information as the mean, the standard deviation and the variance of the data for each variable. With this table we can see the range of values that the variables are getting. For example we can see that the amount of sulfur dioxide of the wines has a really high variance, we can see also that the wines' quality are not so high as we expect. In the last column of the table a histogram is also shown, we can see that the wines quality is mainly near to 5 or 6, so we can deduce that it is being difficult to have big amount of good quality wines at the moment. The content of the table is shown in the Figure [3].

Knime also provide a linear correlation tool, this tool generates a correlation table and a correlation matrix. The correlation between two variables means how related the variables are, that means that when two variables have positive correlation, while one of them increases, the other also does, when the correlation is negative, while one of the increases, the other decreases. The interesting information we can extract here is which variables are more related to the attribute quality. As the amount of information is big, a correlation matrix where correlation is represented in colors is also created. In this case we can see more clearly than the variables that much affect the quality are alcohol, sulphate, fixed acidity, residual acidity and volatile acidity(negatively). The correlation table and correlation matrix is shown in the Figure [4] and the Figure [5].

The next tool generates a box plot that allows us to see the quartiles of the data. What is useful for us is that the values that are outside the limits of the box are outlier values, we can see in the Figure [6] that variables like free sulfur dioxide and total sulfur dioxide have a lot of outlier values.

Finally, we have used the tool scatter plot. A scatter plot indicates the dependency between two variables, when two variables are not dependent between them, the distance between the cloud of points of the variables will be very far. With the generated plot we can choose which relation of two variables we want to see.

3 Data Preparation

This phase consists in a filtering of the incorrect data. In our case, we do not have any missing values. However, as we saw before, there are some data that seems to be outliers. Our data preparation phase will consists in fixing the outliers with the tool given by Knime.

There are different ways to fix outliers, they are mainly removing outlier rows or replacing the outlier values. If we do the firsts approach, we will lose part of the important information, that will maybe make unrealistic because we are removing part of the reality, so we will need to replace the outlier values by others so that these values do not affect too much the analysis of the data in general. Knime provides two different strategies of replacing outliers: replacing the values for missing values, aand the closest permitted value option. The first case is not interesting for us, because we can not treat the missing values neither, so the strategy chosen will be the closest permitted value, that replaces the value of each outlier by the closest value within the permitted interval R.

We can see the settings of the Knime interface in the Figure [7]

4 Modeling

4.1 Linear regression

Linear regression is a linear approach to modelling the relationship between a dependent variable and one or more explanatory variables (or independent variables). Translated to the geometry, many times we can see that data that has certain meaning are distributed so that they are focused in a space making a shape of line that follows some direction, making a linear model will be drawing the straight line that better fits with the explanation of those values. It is very difficult to find a model which explains perfectly the data, because it is expected that the data do not need to follow exactly a straight line.

The linear regression is given by the following formula:

$$y = b_0 + b_1 x$$

Where x is the explanatory variable and y the dedependent variable, the slope of the line is b_1 and the intercept is b_0.

A linear regression model assumes that the relationship between the dependent variable y and the vector of regressors x is linear. This is what we want to guarantee before applying the model, otherwise the results obtained will not make sense.

Another assumption we should make is that there should not be significant outliers in the data. We can imagine in a plot that outliers are values that are far away from the main data, hence, making a line that also "explain" that outliers values will distort absolutely the model. In our case, we do not have to worry about this because outliers are already fixed in the previous step.

4.2 Polynomial regression

For the modeling process, we did four different modeling tasks: polynomial regression, linear regression, random forest, and tree ensemble. In our project, we will create different metanodes, that are kind of modules where inside each metanode we will have different tools, each tool will perform a different task to complete the modeling process. The meaning of the tools or interpretation of the results will be explained with more details in the evaluation process.

Polynomial regression is a form of regression analysis in which the relationship between an independent variable x and a dependent variable y is modelled as an nth degree polynomial in x.

Polynomial regression is a special case of linear regression, we have also an axis for the data value and an axis for the target value. The reason we use this kind of models is because in the real world, in the most of times, the data do not behave as a straight line, we can observe that plots of the data are frequently scattered, in such case only polynomial regression can explain better the data.

The polynomial linear regression is given by the following formula:

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + ... b_n x_1^n + \epsilon$$

Where ϵ is an unobserved random error with mean zero conditioned on a scalar variable x.

The aim of the modeling process is to find values that substituting the data into the equation, the error between the value given by the equation and the real value is the minimum possible. In fact, we could create a model where the error is minimized to 0, that will mean that the model fits perfectly with the actual data, for instance, that does not make sense, because that model would not be useful to predict a new possible value.

4.3 Random forest

Random forests or random decision forests are a learning method for classification, regression and other tasks. It consists in constructing a multitude of decision trees at training time and then merges them together to get a more accurate and stable prediction.

Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. Trees can be made more random, by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

Another advantage of random forest is that looking at the feature importance, we can decide which features we may want to drop, because they don't contribute enough or nothing to the prediction process.

4.4 Ensemble learning

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

Tree ensembles are flexible predictive models that can capture relevant variables and to some extent their interactions in a compact and interpretable manner. Most algorithms for obtaining tree ensembles are based on versions of boosting or Random Forest.

One of the ensemble methods are BAGGing, or Bootstrap AGGregating. Given a sample of data, multiple bootstrapped subsamples are pulled. A Decision Tree is formed on each of the bootstrapped subsamples. After each subsample Decision Tree has been formed, an algorithm is used to aggregate over the Decision Trees to form the most efficient predictor.

5 Evaluation

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it.

In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation.

The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once. Based on these advantages, we preferred k-fold Cross Validation instead of using partitioning node and percentage split method.

So as to use this technique, we used X-Partitioner node. As you can see in the node, there are 2 arrows. The upper arrow represents the training data and the bottom one represents the test data. Train data is going to stream via upper arrow to train algorithm and test data is going to stream via bottom arrow to prediction algorithm. Number of Validations was selected as 10 and Stratified

Sampling was used based on class column "Quality". In X-Partitioner node when Strafied Sampling was signed, the partitions are sampled randomly, but the class distribution from the column selected is maintained. This will enable us to achieve healthier results.

After this process will be fragmented, we will need to collect them after the prediction algorithm. In order to do that, X-Aggregator node was used. It collects the result from a predictor node, compares predicted class and real class and outputs the predictions for all rows and the iteration statistics. Moreover, X-Aggregator also have arrows as upper and bottom arrow which is used for data streaming. One of them is prediction table and the other is error rate. We can access them in this node if we would like to view. While configuring target column was selected as quality and prediction columns was selected as prediction (quality). In error rate demonstrates some kind of error types such as total squared error and mean squared error. Mean Squared Error (MSE) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors that is, the average squared difference between the estimated values and what is estimated. Additionally, the MSE is never negative, and values closer to zero are better.

$$TSE = \sum_{i-1}^{n} (Y_i - \hat{Y}_i)^2$$

$$MSE = \frac{1}{n} \sum_{i-1}^{n} (Y_i - \hat{Y}_i)^2$$

Where y_i is the actual value, $\hat{Y}_i$ the predicted value, and $(Y_i - \hat{Y}_i)^2$ the squared error.

We calculated errors with using a node which is Numeric Scorer computes certain statistics between the a numeric column's values (y_i) and predicted (p_i) values. It computes $R^2 = \frac{1-(Y_i-\hat{Y}_i)^2}{\sum(\frac{Y_i-1}{n\times\sum Y_i})^2}$. We can see some parameters below:
Mean absolute error: $\frac{1}{n\times\sum|Y_i-\hat{Y}_i|}$
Mean squared error: $\frac{1}{n\times\sum(Y_i-\hat{Y}_i)^2}$
Root mean squared error: $\sqrt{\frac{1}{n\times\sum(Y_i-\hat{Y}_i)^2}}$
Mean signed difference: $\frac{1}{n\times\sum(Y_i-\hat{Y}_i)}$
The computed values were inspected in the node's view and/or further processed using the output table.

R-squared (R^2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. High R-squared indicates the success of the model.

Mean Absolute Error (MAE) is a measure of difference between two continuous variables. Besides, it is the average vertical distance between each point and the identity line and also the average horizontal distance between each point and the identity line.

Mean Squared Error (MSE) has already been mentioned.

Root Mean Squared Error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. RMSE is never negative, and a value of 0 (almost never achieved in practice) would indicate a perfect fit to the data. In general, a lower RMSE is better than a higher one. However, comparisons across different types of data would be invalid because the measure is dependent on the scale of the numbers used.

RMSE is the square root of the average of squared errors. The effect of each error on RMSE is proportional to the size of the squared error; thus larger errors have a disproportionately large effect on RMSE. Consequently, RMSE is sensitive to outliers.

Mean Signed Difference is one of a number of statistics that can be used to assess an estimation procedure, and it would often be used in conjunction with a sample version of the mean square error.

In our case, we used 4 algorithms which are called as Polynomial Regression, Linear Regression, Random Forest (Regression), Tree Ensemble (Regression). The algorithm with the lowest error rate and the highest R-squared is Tree Ensemble (Regression). The order is as follows: Tree Ensemble (Regression) ¿ Random Forest (Regression) ¿ Polynomial Regression ¿ Linear Regression. There is no big difference among Tree Ensemble (Regression) - Random Forest (Regression) and among Polynomial Regression - Linear Regression. However, if we evaluate as pair and say Pair A includes Tree Ensemble (Regression) - Random Forest (Regression) and B includes Polynomial Regression - Linear Regression, we can say that there is such a big difference based on Error Rates and R-squared.

All the results can be seen from the Figure [8].

6 Development

As we explained before, the development part consists in taking a business plan to apply these knowledge. However, we will not be able to carry out this project into the reality, hence, regarding to the CRISP-DM schema, our project will finish here.

7 Conclusion

Thanks to this project, we had the chance to practise the use of Knime to do some usual tasks related to data mining. We have realized that Knime is a really useful tool, it is very simple to use, in few time it allows user to make quite complex data mining systems.

Following the RISP-DM schema has been useful to understand how a data mining project works in the enterprises. Before doing the project we thought it would be a simple project, but now we contemplated why it is required to separate this processes and how complex the system is, if one of the processes are not done correctly ,the implications can be very negative and will affect the next steps. Fortunately, the model accepts going back and readjust the mistakes in the previous steps.

With the application of the wine data set we had the chance to check the behaviour of the algorithms in the practise. Fortunately, wine quality data set is a data set very appropriate to work with, the results of the analysis allows us to understand better the meaning of the results given by these tasks.

Finally, even that Knime has made easier our task of building the models, it is necessary to understand the meaning of the algorithms, otherwise it would be impossible to generate and interpret the results. The learning of Knime will be very useful for the development of our projects related to data science in the future.

8 Annex

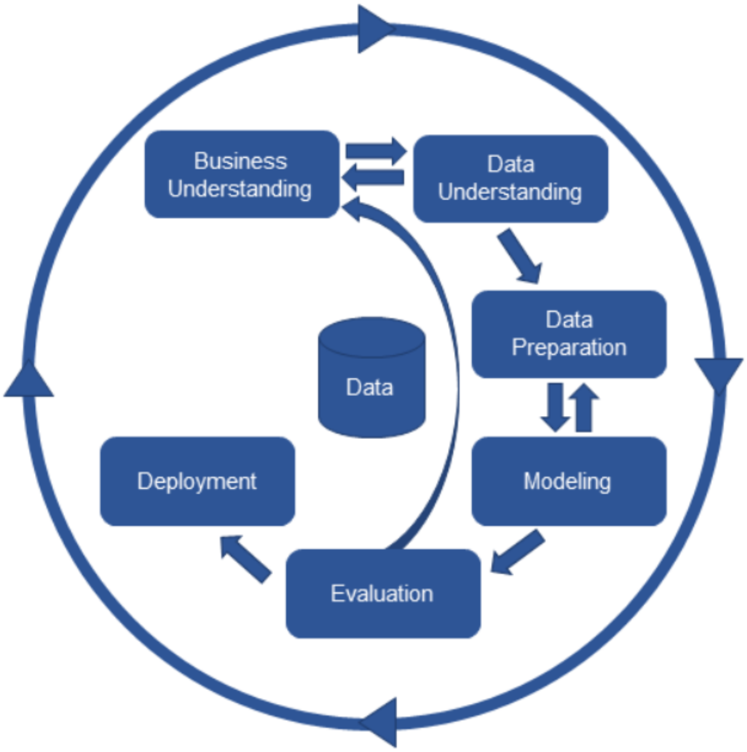
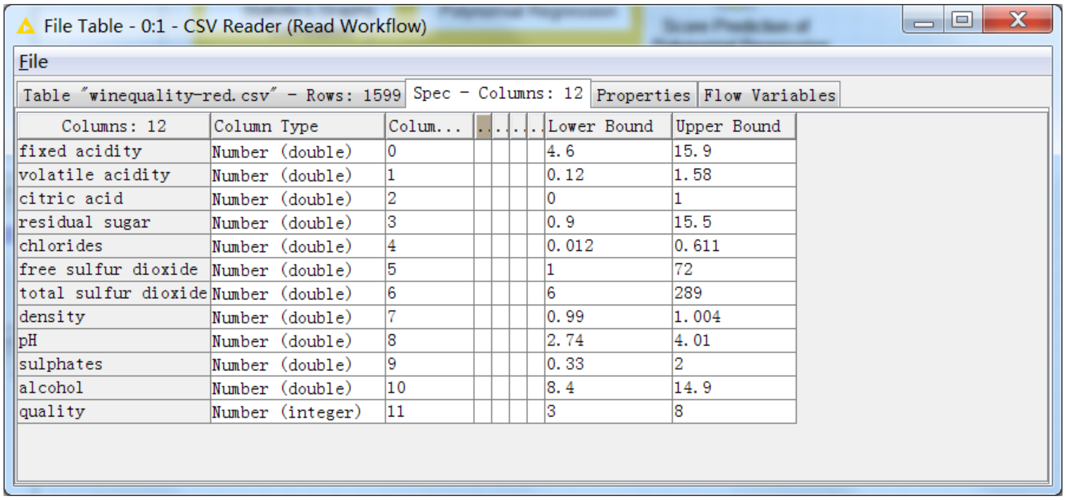


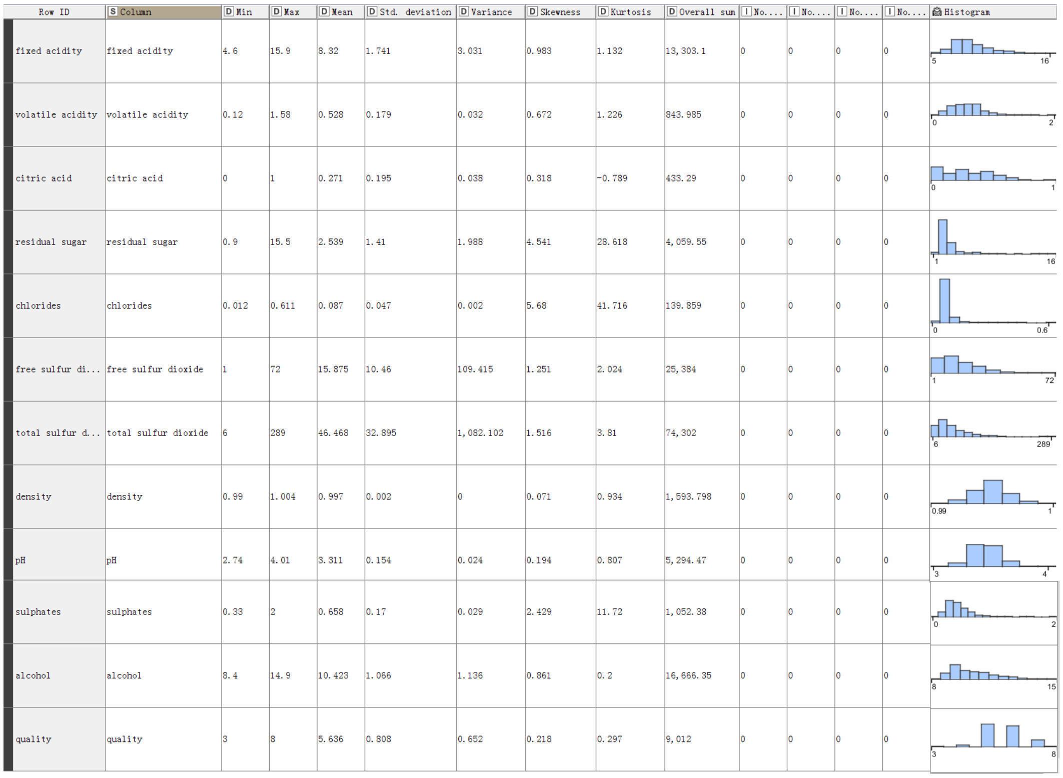
Figure 1: CRISP-DM schema



File Table - 0:1 - CSV Reader (Read Workflow)

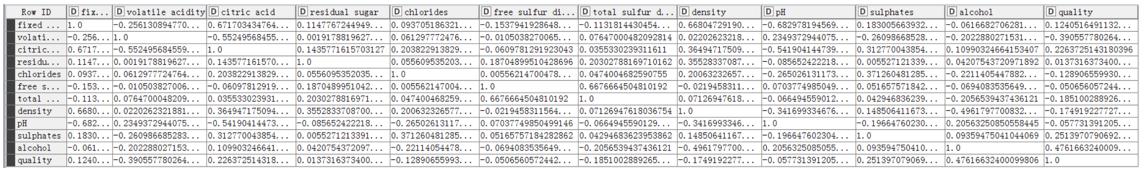
Columns: 12	Column Type	Colum...	...	...	...	Lower Bound	Upper Bound
fixed acidity	Number (double)	0				4.6	15.9
volatile acidity	Number (double)	1				0.12	1.58
citric acid	Number (double)	2				0	1
residual sugar	Number (double)	3				0.9	15.5
chlorides	Number (double)	4				0.012	0.611
free sulfur dioxide	Number (double)	5				1	72
total sulfur dioxide	Number (double)	6				6	289
density	Number (double)	7				0.99	1.004
pH	Number (double)	8				2.74	4.01
sulphates	Number (double)	9				0.33	2
alcohol	Number (double)	10				8.4	14.9
quality	Number (integer)	11				3	8

Figure 2: General information of the data set



Row ID	S Column	D Min	D Max	D Mean	D Std. deviation	D Variance	D Skewness	D Kurtosis	D Overall sum	I No....	I No....	I No....	I No....	Histogram
fixed acidity	fixed acidity	4.6	15.9	8.32	1.741	3.031	0.983	1.132	13,303.1	0	0	0	0	5 16
volatile acidity	volatile acidity	0.12	1.58	0.528	0.179	0.032	0.672	1.226	843.985	0	0	0	0	0 2
citric acid	citric acid	0	1	0.271	0.195	0.038	0.318	-0.789	433.29	0	0	0	0	0 1
residual sugar	residual sugar	0.9	15.5	2.539	1.41	1.988	4.541	28.618	4,059.55	0	0	0	0	1 16
chlorides	chlorides	0.012	0.611	0.087	0.047	0.002	5.68	41.716	139.859	0	0	0	0	0 0.6
free sulfur di...	free sulfur dioxide	1	72	15.875	10.46	109.415	1.251	2.024	25,384	0	0	0	0	1 72
total sulfur d...	total sulfur dioxide	6	289	46.468	32.895	1,082.102	1.516	3.81	74,302	0	0	0	0	6 289
density	density	0.99	1.004	0.997	0.002	0	0.071	0.934	1,593.798	0	0	0	0	0.99 1
pH	pH	2.74	4.01	3.311	0.154	0.024	0.194	0.807	5,294.47	0	0	0	0	3 4
sulphates	sulphates	0.33	2	0.658	0.17	0.029	2.429	11.72	1,052.38	0	0	0	0	0 2
alcohol	alcohol	8.4	14.9	10.423	1.066	1.136	0.861	0.2	16,666.35	0	0	0	0	8 15
quality	quality	3	8	5.636	0.808	0.652	0.218	0.297	9,012	0	0	0	0	3 8

Figure 3: Statistics of the variables



Row ID	D fix...	D volatile acidity	D citric acid	D residual sugar	D chlorides	D free sulfur di...	D total sulfur d...	D density	D pH	D sulphates	D alcohol	D quality
fixed ...	1.0	-0.256130894770...	0.671703434764...	0.1147767244949...	0.093705186321...	-0.1537941928648...	-0.1131814430454...	0.66804729190...	-0.682978194569...	0.183005663932...	-0.0616682706281...	0.1240516491132...
volati...	-0.256...	1.0	-0.55249568455...	0.0019178819627...	0.061297772476...	-0.0105038270065...	0.07647000482092814	0.02202623218...	0.2349372944075...	-0.26098668528...	-0.2022880271531...	-0.390557780264...
citric...	0.6717...	-0.552495684559...	1.0	0.1435771615703127	0.203822913829...	-0.0609781291923043	0.0355330239311611	0.36494717509...	-0.541904144739...	0.312770043854...	0.10990324664153407	0.2263725143180396
residu...	0.1147...	0.0019178819627...	0.143577161570...	1.0	0.055609535203...	0.18704899510428696	0.20302788169710162	0.35528337087...	-0.085652422218...	0.005527121339...	0.04207543720971892	0.0137316373400...
chlorides	0.0937...	0.0612977724764...	0.203822913829...	0.0556095352035...	1.0	0.00556214700478...	0.0474004682590755	0.20063232657...	-0.265026131173...	0.371260481285...	-0.2211405447882...	-0.128906559930...
free s...	-0.153...	-0.010503827006...	-0.06097812919...	0.1870489951042...	0.005562147004...	1.0	0.6676664504810192	-0.0219458311...	0.0703774985049...	0.051657571842...	-0.0694083535649...	-0.050656057244...
total ...	-0.113...	0.0764700048209...	0.035533023931...	0.2030278816971...	0.047400468259...	0.6676664504810192	1.0	0.07126947618...	-0.066494559012...	0.042946836239...	-0.2056539437436121	-0.185100288926...
density	0.6680...	0.0220262321881...	0.364947175094...	0.3552833708700...	0.200632326577...	-0.0219458311564...	0.07126947618036754	1.0	-0.341699334676...	0.148506411673...	-0.4961797700832...	-0.174919227727...
pH	-0.682...	0.2349372944075...	-0.54190414473...	-0.085652422218...	-0.26502613117...	0.07037749850499146	-0.0664945590129...	-0.3416993346...	1.0	-0.19664760230...	0.20563250850558445	-0.057731391205...
sulphates	0.1830...	-0.260986685283...	0.312770043854...	0.0055271213391...	0.371260481285...	0.05165757184282862	0.04294683623953862	0.14850641167...	-0.196647602304...	1.0	0.09359475041044069	0.2513970790692...
alcohol	-0.061...	-0.202288027153...	0.109903246641...	0.0420754372097...	-0.22114054478...	-0.0694083535649...	-0.2056539437436121	-0.4961797700...	0.2056325085055...	0.093594750410...	1.0	0.4761663240009...
quality	0.1240...	-0.390557780264...	0.226372514318...	0.0137316373400...	-0.12890655993...	-0.0506560572442...	-0.1851002889265...	-0.1749192277...	-0.057731391205...	0.251397079069...	0.47616632400099806	1.0

Figure 4: Correlation Table

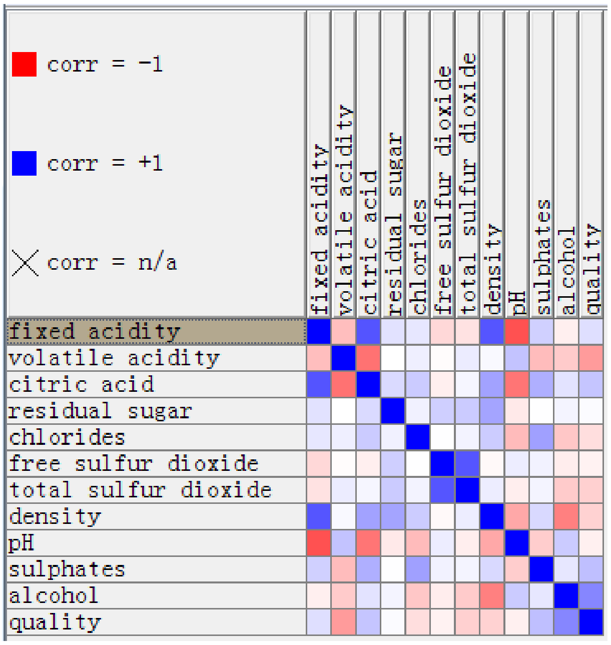


Figure 5: Correlation Matrix

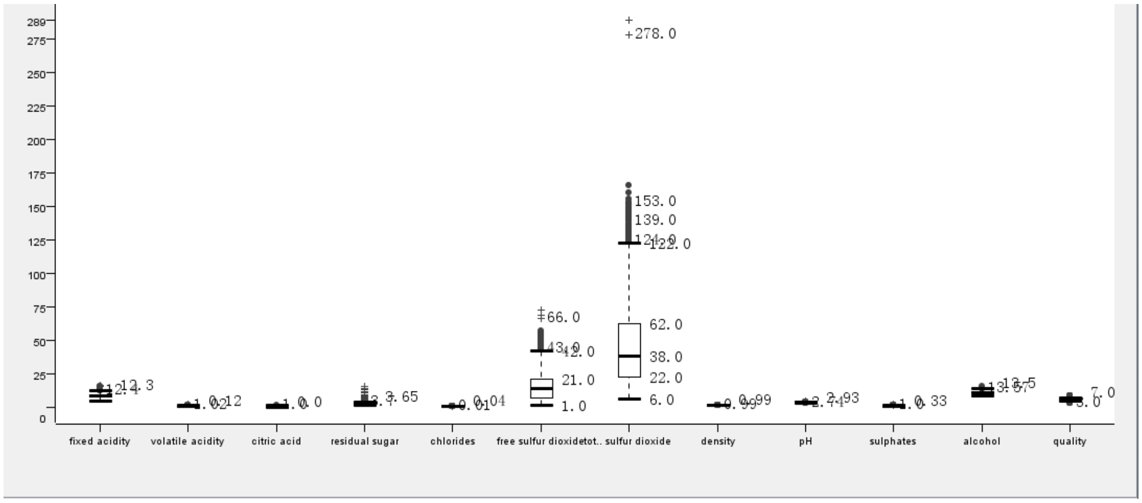
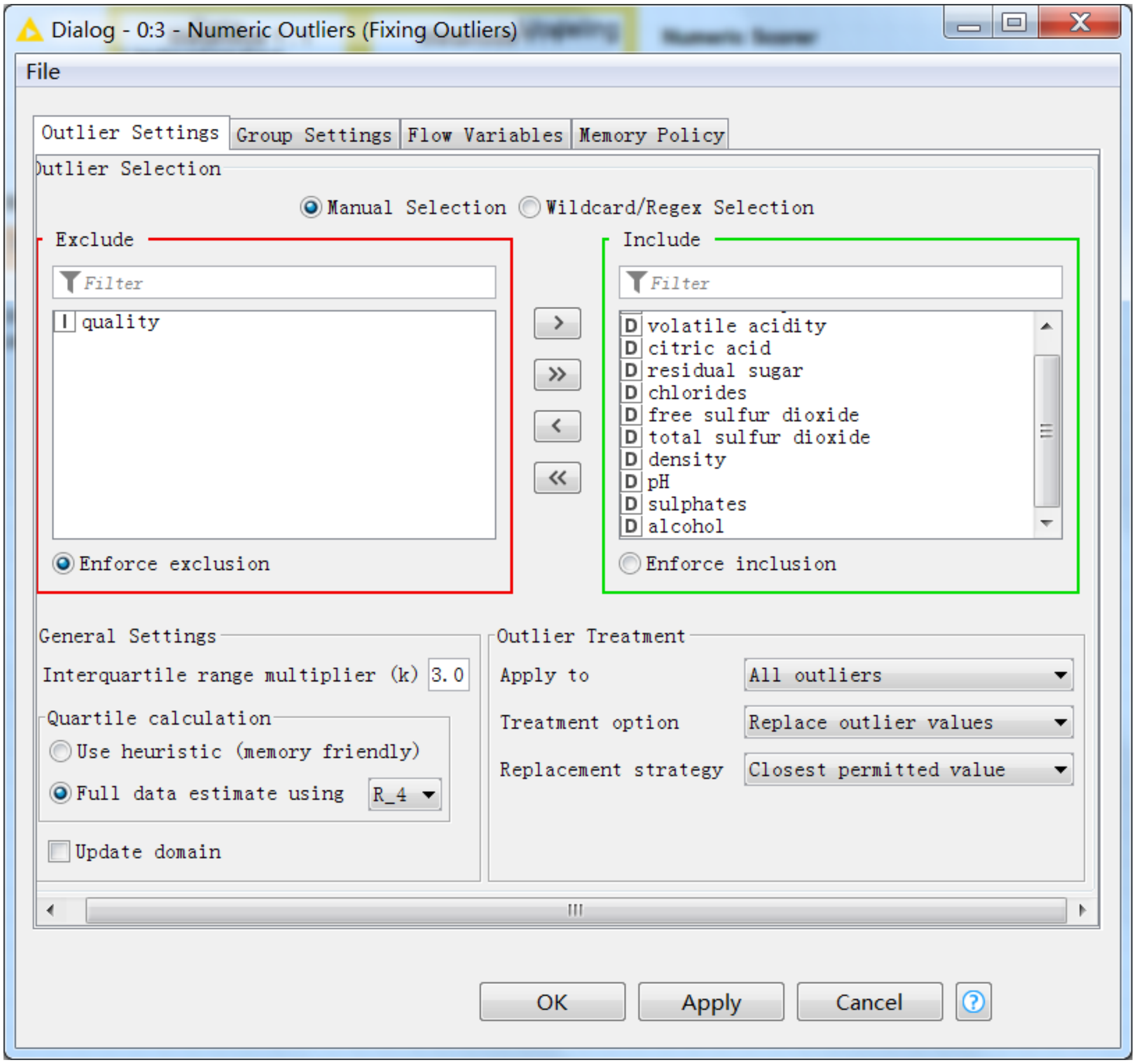


Figure 6: Box plot



)utlier Selection

● Manual Selection ○ Wildcard/Regex Selection

Exclude

Filter

I quality

● Enforce exclusion

Include

Filter

D volatile acidity
D citric acid
D residual sugar
D chlorides
D free sulfur dioxide
D total sulfur dioxide
D density
D pH
D sulphates
D alcohol

○ Enforce inclusion

General Settings

Interquartile range multiplier (k) 3.0

Quartile calculation

○ Use heuristic (memory friendly)

● Full data estimate using R_4

□ Update domain

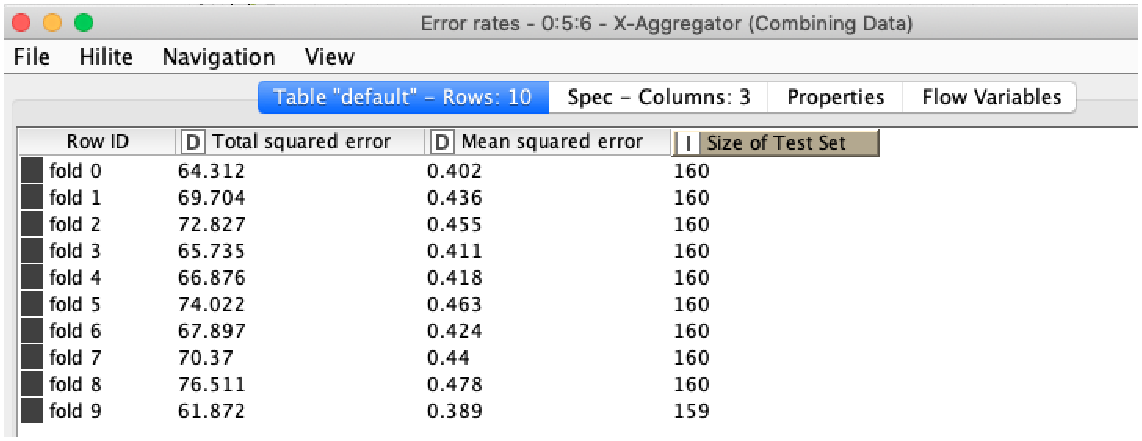
Outlier Treatment

Apply to: All outliers

Treatment option: Replace outlier values

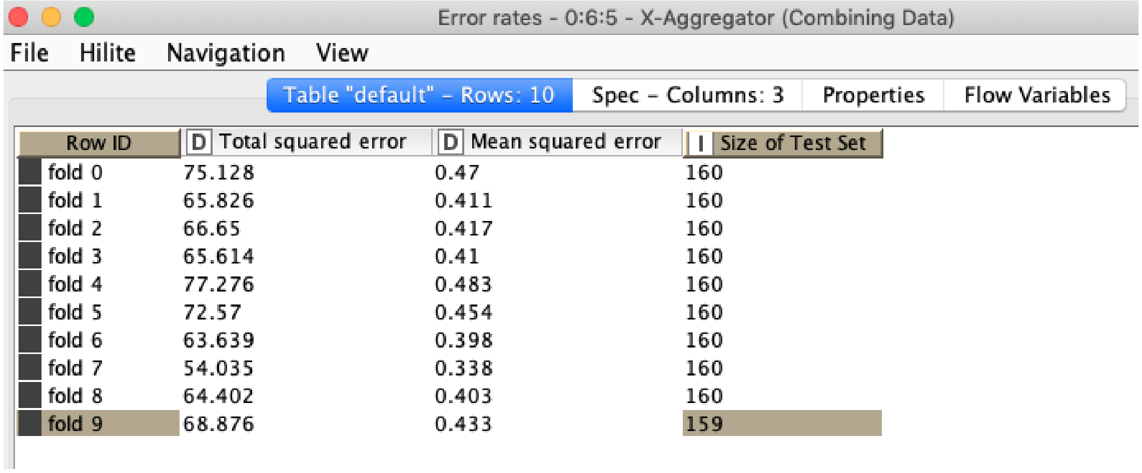
Replacement strategy: Closest permitted value

Figure 7: Fixing outliers configuration.


Error rates - 0:5:6 - X-Aggregator (Combining Data)
File
Hilite
Navigation
View
Table "default" – Rows: 10
Spec – Columns: 3
Properties
Flow Variables

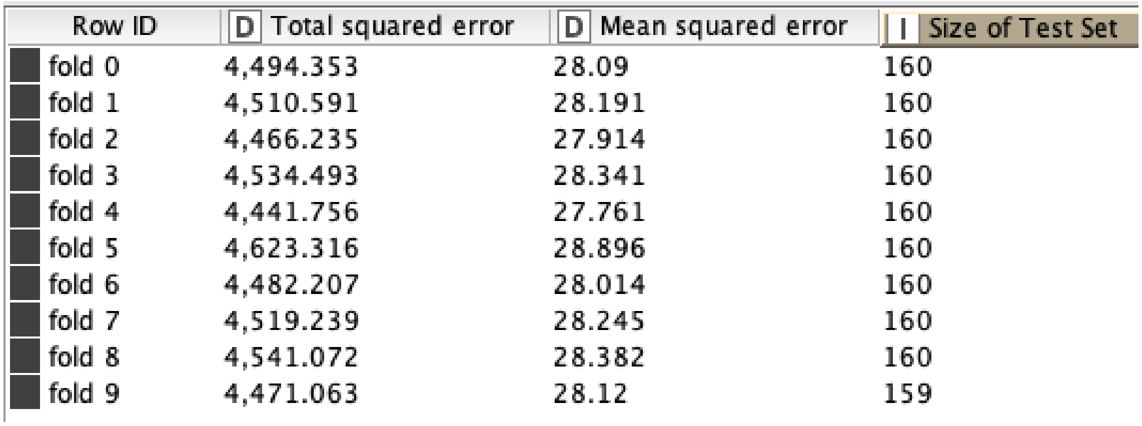
Row ID	D Total squared error	D Mean squared error	I Size of Test Set
fold 0	64.312	0.402	160
fold 1	69.704	0.436	160
fold 2	72.827	0.455	160
fold 3	65.735	0.411	160
fold 4	66.876	0.418	160
fold 5	74.022	0.463	160
fold 6	67.897	0.424	160
fold 7	70.37	0.44	160
fold 8	76.511	0.478	160
fold 9	61.872	0.389	159

Figure 8: Error rates linear regression.


Error rates - 0:6:5 - X-Aggregator (Combining Data)
File
Hilite
Navigation
View
Table "default" – Rows: 10
Spec – Columns: 3
Properties
Flow Variables

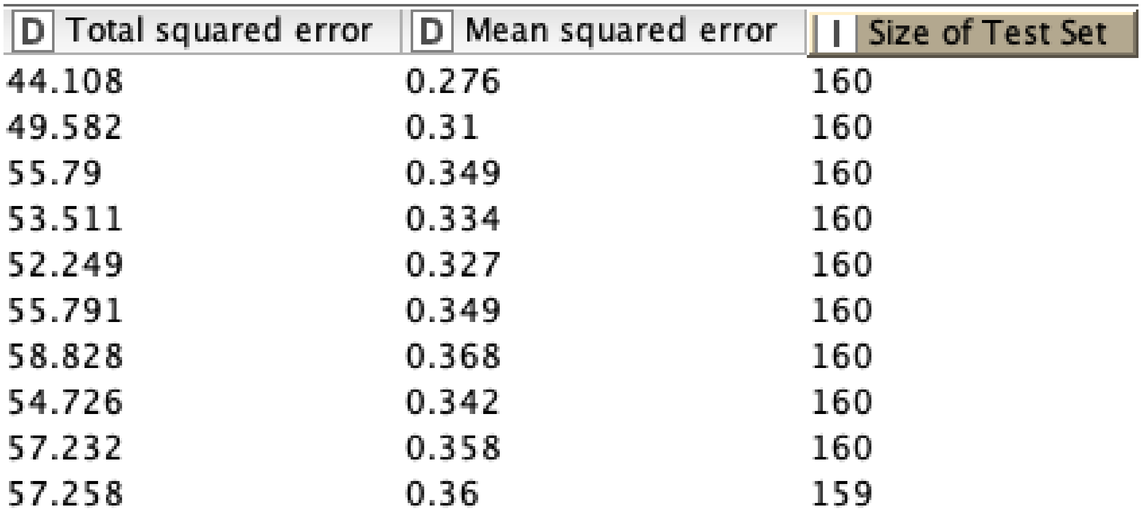
Row ID	D Total squared error	D Mean squared error	I Size of Test Set
fold 0	75.128	0.47	160
fold 1	65.826	0.411	160
fold 2	66.65	0.417	160
fold 3	65.614	0.41	160
fold 4	77.276	0.483	160
fold 5	72.57	0.454	160
fold 6	63.639	0.398	160
fold 7	54.035	0.338	160
fold 8	64.402	0.403	160
fold 9	68.876	0.433	159

Figure 9: Error rates polynomial regression.



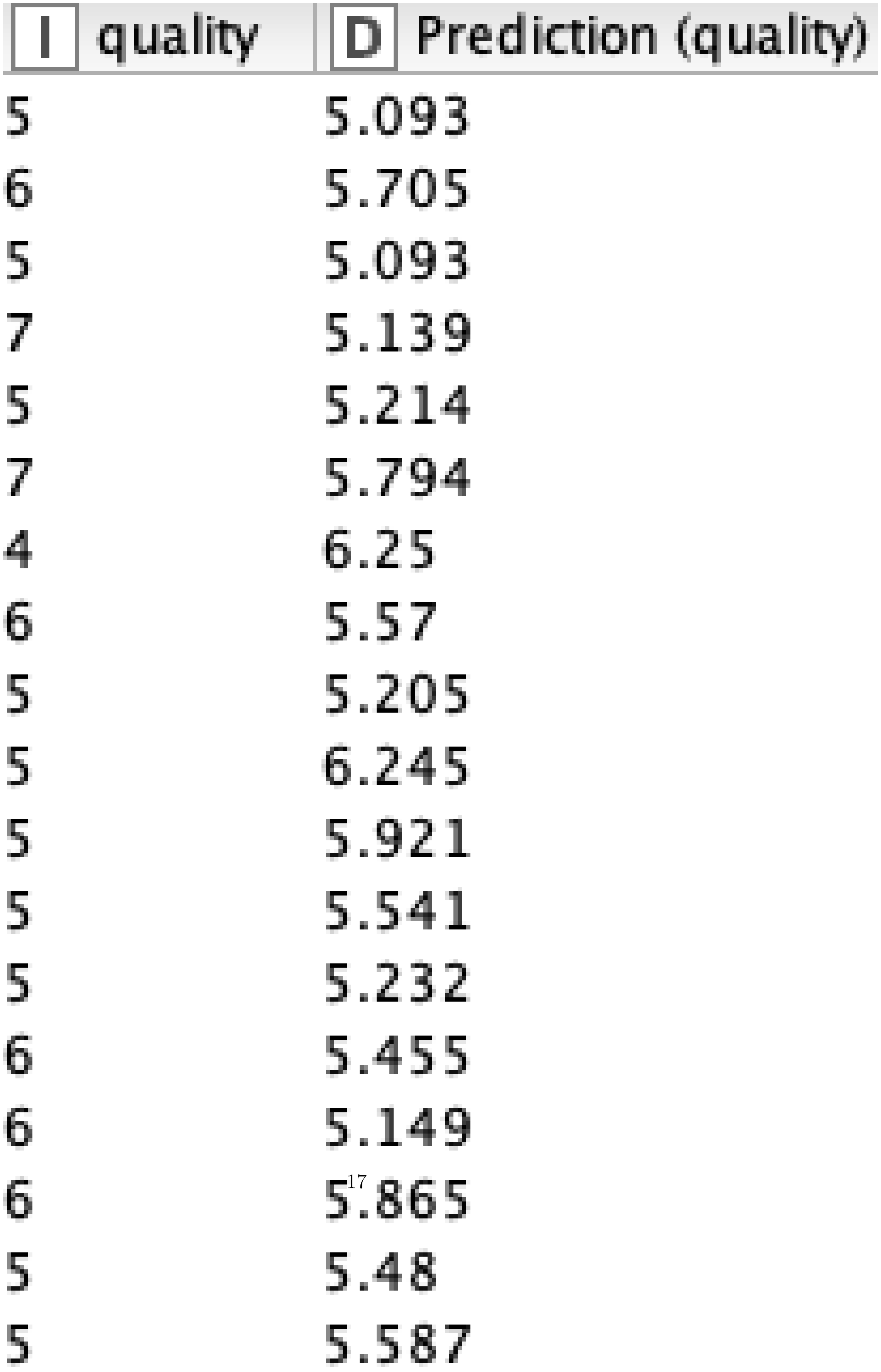
Row ID	D Total squared error	D Mean squared error	I Size of Test Set
■ fold 0	4,494.353	28.09	160
■ fold 1	4,510.591	28.191	160
■ fold 2	4,466.235	27.914	160
■ fold 3	4,534.493	28.341	160
■ fold 4	4,441.756	27.761	160
■ fold 5	4,623.316	28.896	160
■ fold 6	4,482.207	28.014	160
■ fold 7	4,519.239	28.245	160
■ fold 8	4,541.072	28.382	160
■ fold 9	4,471.063	28.12	159

Figure 10: Error random forest.

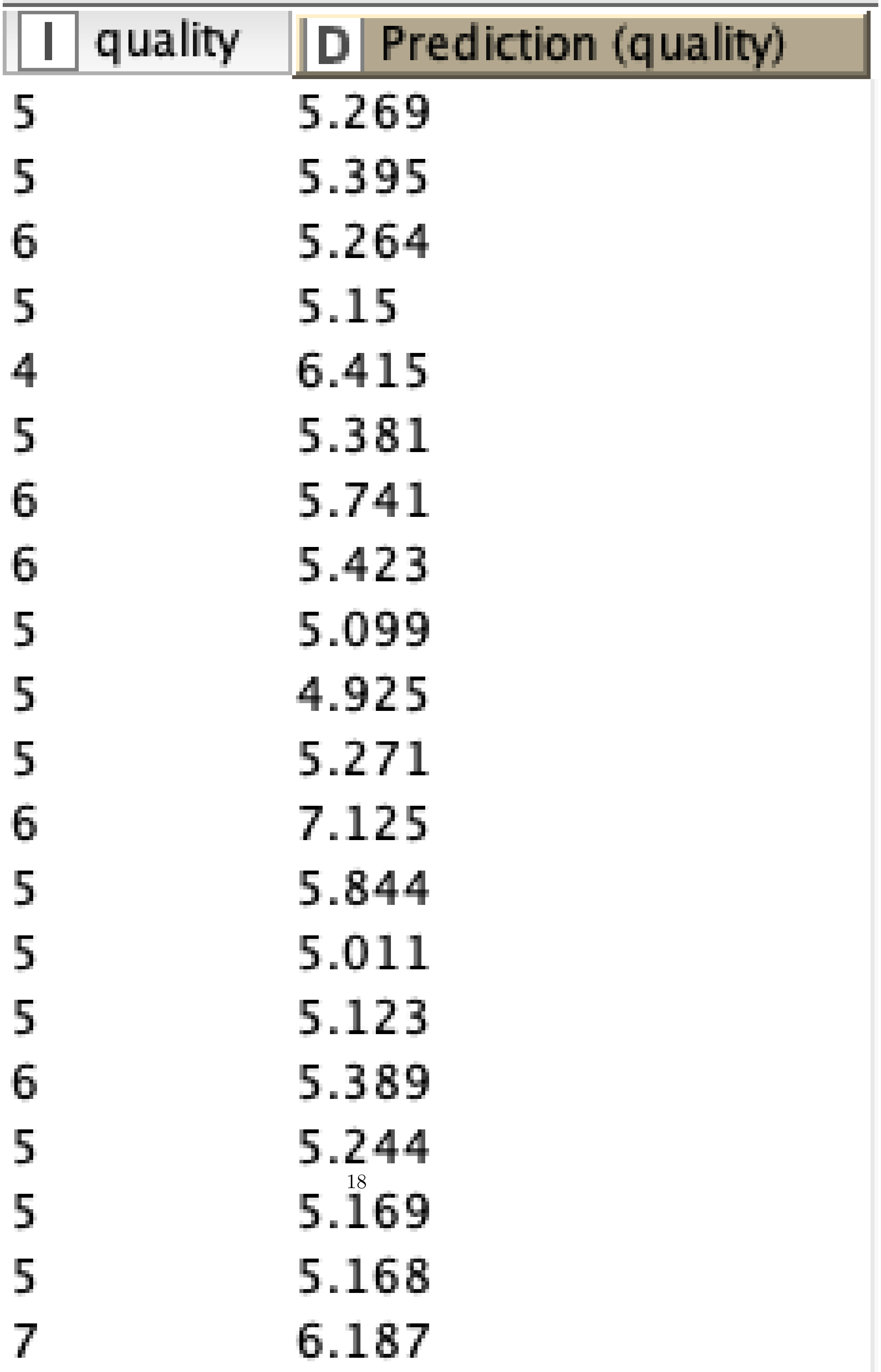


D Total squared error	D Mean squared error	I Size of Test Set
44.108	0.276	160
49.582	0.31	160
55.79	0.349	160
53.511	0.334	160
52.249	0.327	160
55.791	0.349	160
58.828	0.368	160
54.726	0.342	160
57.232	0.358	160
57.258	0.36	159

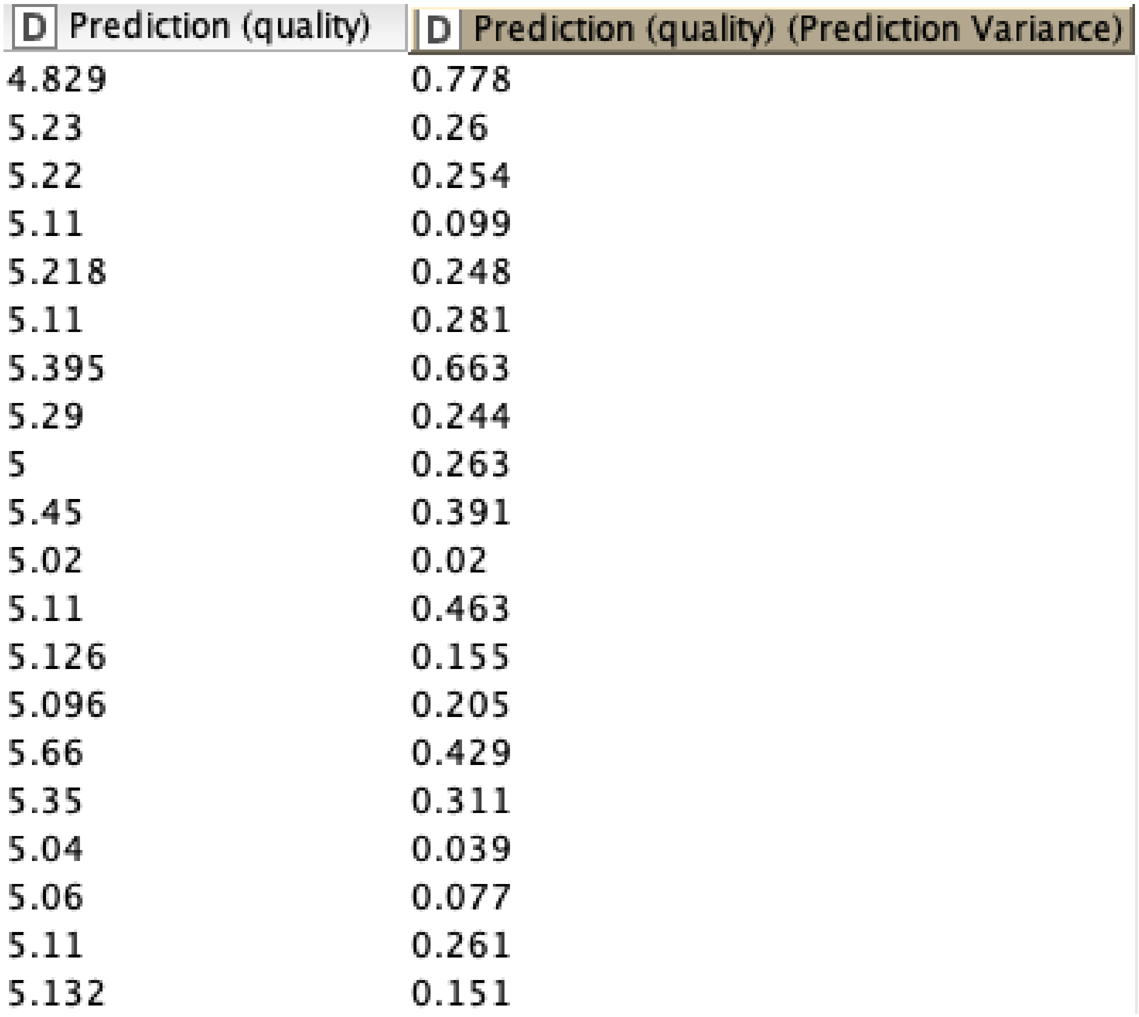
Figure 11: Error tree ensemble.



I quality	D Prediction (quality)
5	5.093
6	5.705
5	5.093
7	5.139
5	5.214
7	5.794
4	6.25
6	5.57
5	5.205
5	6.245
5	5.921
5	5.541
5	5.232
6	5.455
6	5.149
6	5.865
5	5.48
5	5.587

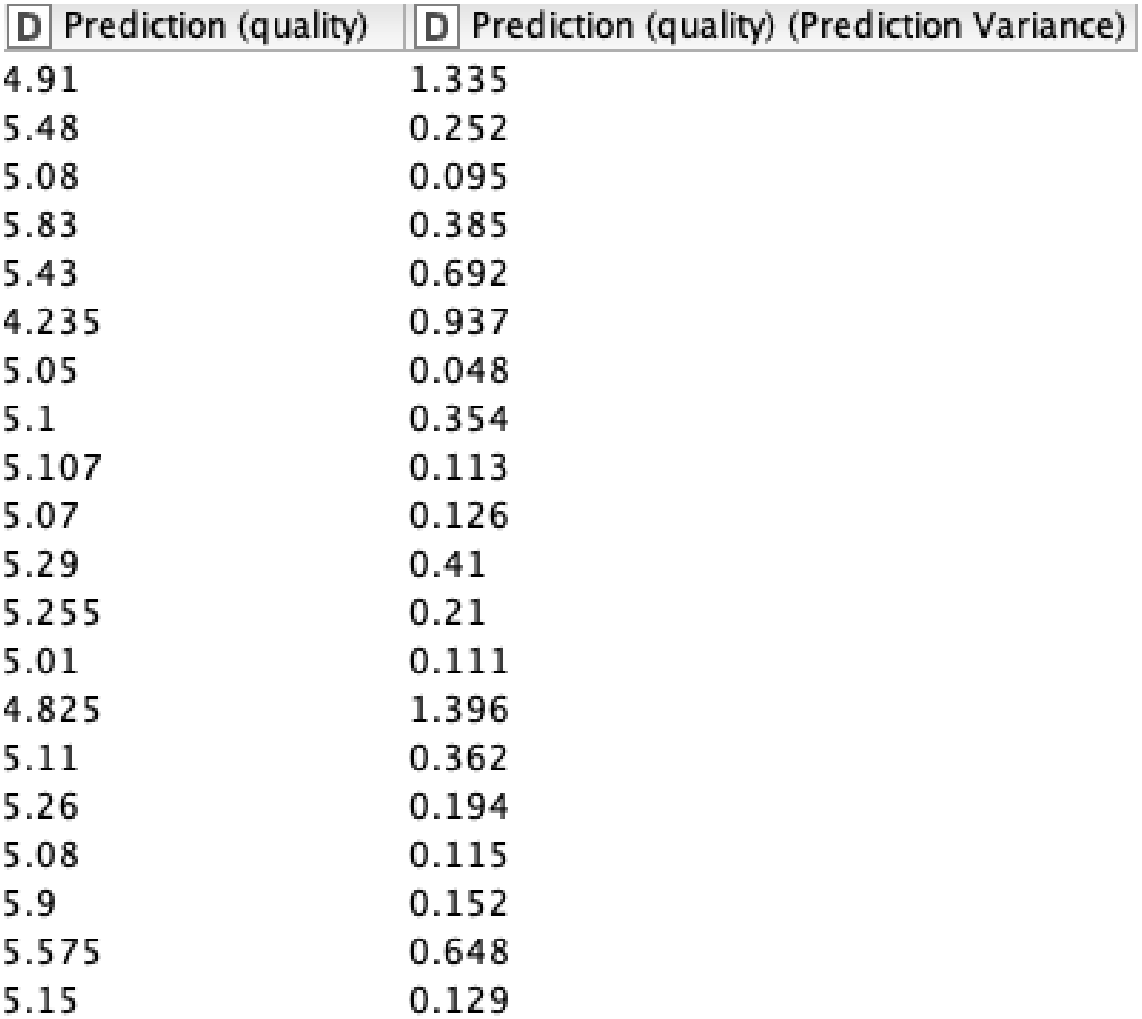


I quality	D Prediction (quality)
5	5.269
5	5.395
6	5.264
5	5.15
4	6.415
5	5.381
6	5.741
6	5.423
5	5.099
5	4.925
5	5.271
6	7.125
5	5.844
5	5.011
5	5.123
6	5.389
5	5.244
5	5.169
5	5.168
7	6.187



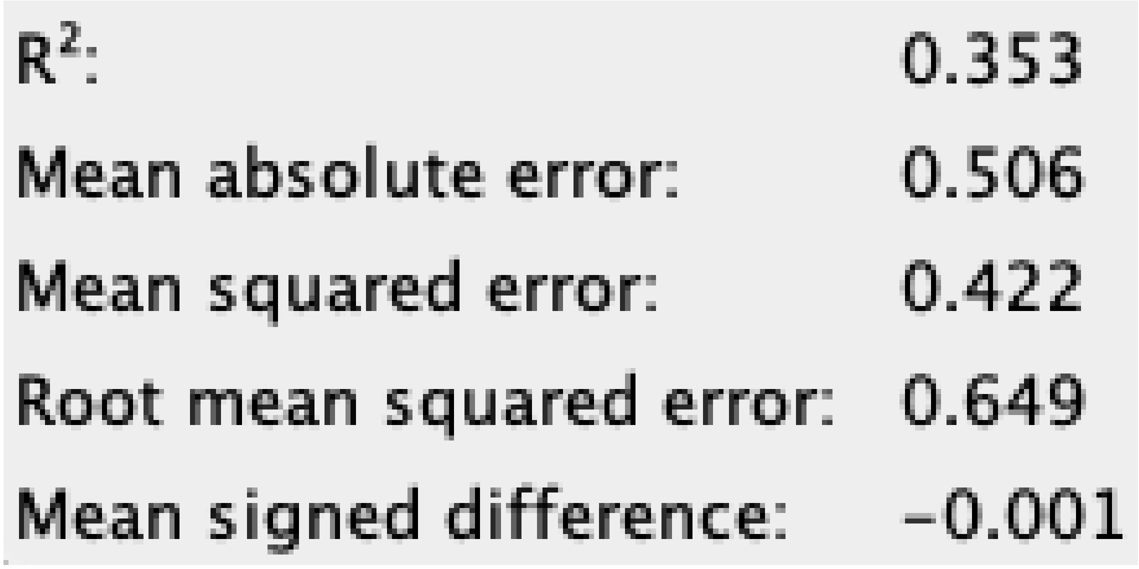
D Prediction (quality)	D Prediction (quality) (Prediction Variance)
4.829	0.778
5.23	0.26
5.22	0.254
5.11	0.099
5.218	0.248
5.11	0.281
5.395	0.663
5.29	0.244
5	0.263
5.45	0.391
5.02	0.02
5.11	0.463
5.126	0.155
5.096	0.205
5.66	0.429
5.35	0.311
5.04	0.039
5.06	0.077
5.11	0.261
5.132	0.151

Figure 14: Results random forest.



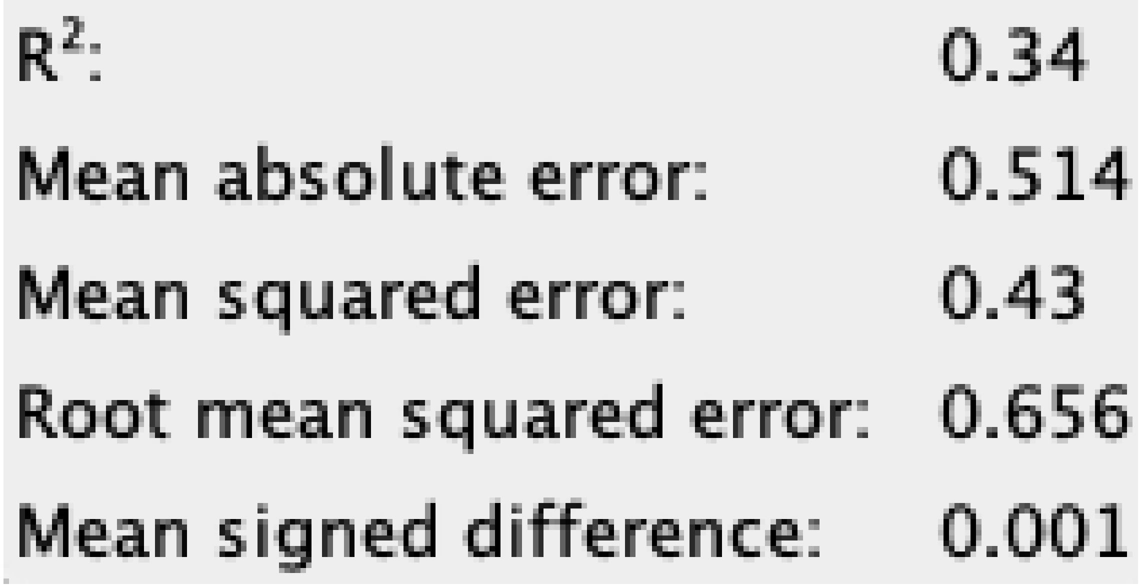
D Prediction (quality)	D Prediction (quality) (Prediction Variance)
4.91	1.335
5.48	0.252
5.08	0.095
5.83	0.385
5.43	0.692
4.235	0.937
5.05	0.048
5.1	0.354
5.107	0.113
5.07	0.126
5.29	0.41
5.255	0.21
5.01	0.111
4.825	1.396
5.11	0.362
5.26	0.194
5.08	0.115
5.9	0.152
5.575	0.648
5.15	0.129

Figure 15: Results tree ensemble.



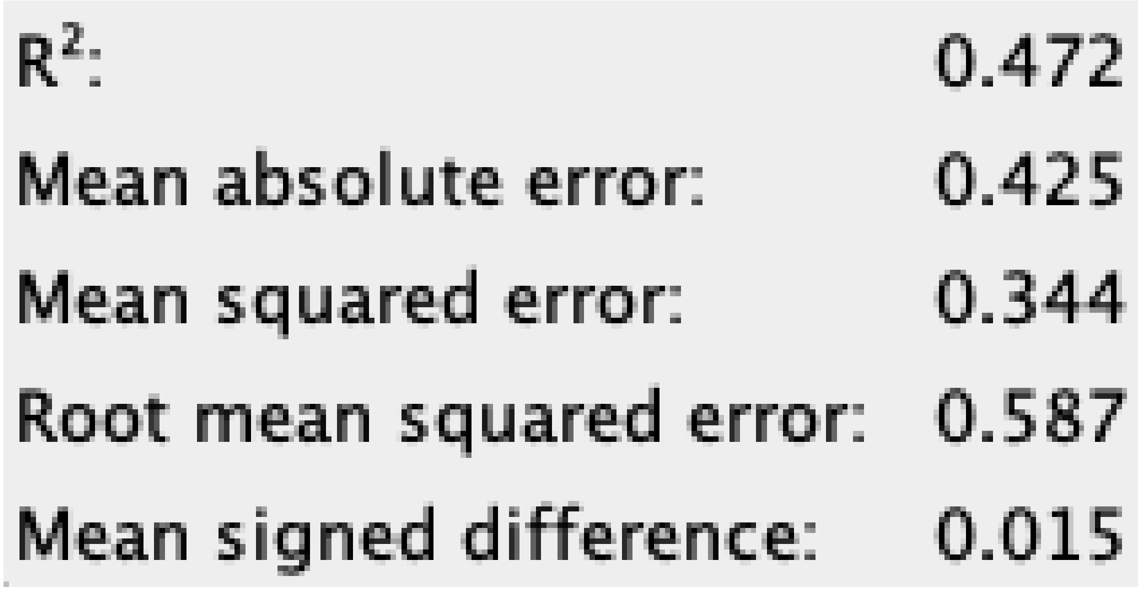
R²:	0.353
Mean absolute error:	0.506
Mean squared error:	0.422
Root mean squared error:	0.649
Mean signed difference:	–0.001

Figure 16: Score polynomial regression.



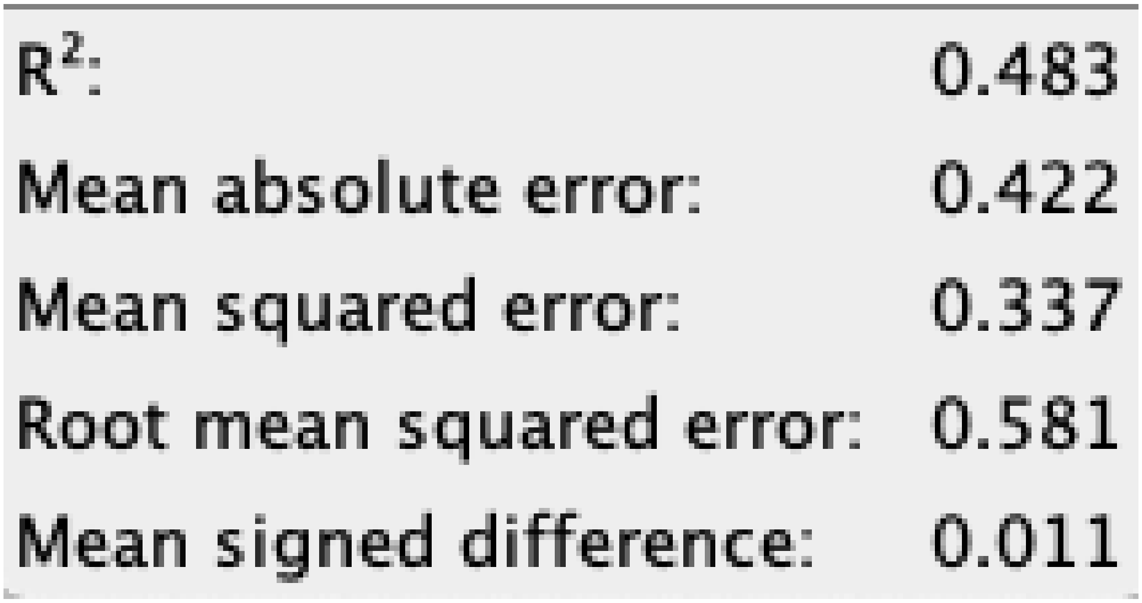
R²:	0.34
Mean absolute error:	0.514
Mean squared error:	0.43
Root mean squared error:	0.656
Mean signed difference:	0.001

Figure 17: Score linear regression.



```
R^2:                       0.472
Mean absolute error:       0.425
Mean squared error:        0.344
Root mean squared error:   0.587
Mean signed difference:    0.015
```

Figure 18: Score random forest.



```
R^2:                       0.483
Mean absolute error:       0.422
Mean squared error:        0.337
Root mean squared error:   0.581
Mean signed difference:    0.011
```

Figure 19: Score tree ensemble