

CPS844 - Data Mining

Dr. Cherie Ding

Predict Abalone's Number of Rings

George Saade - 500867644

Siena Smith - 500775103

Getting the data ready

The dataset was downloaded from UCI Machine Learning Repository. To make a dataset readable by WEKA, the file must be converted to a compatible format. The data downloaded had a “.data” extension with an accompanying .names file, normally associated with C4.5-format data, but the .names file did not contain correctly formatted information. Opening the .data file using a text editor, we observed that it had a format associated with CSV files: each line contained one instance and the attributes were separated by a comma. After observing this, we found that we were able to load the dataset into WEKA by changing the .data file extension to .csv.

The preferred format for WEKA is ARFF. A WEKA ARFF file contains two distinct sections, the data information section and the header information section. The original file already contains the data section formatted properly. The header information was added by us manually through WEKA’s “Edit” function. This was done with the aid of the webpage the data was taken from, which contained the names of the attributes and the possible values and data types for each one of them. The continuous datatype for the attributes was treated as numeric as WEKA documentation² indicates.

After labelling the attributes appropriately, the file was saved with a “.arff” extension, resolving .

Dataset overview

Dataset	Data Types	Default Task	Attribute Types	Instances	Attributes	Year
Abalone	Multivariate	Classification	Categorical, Integer, Real	4177	8	1994

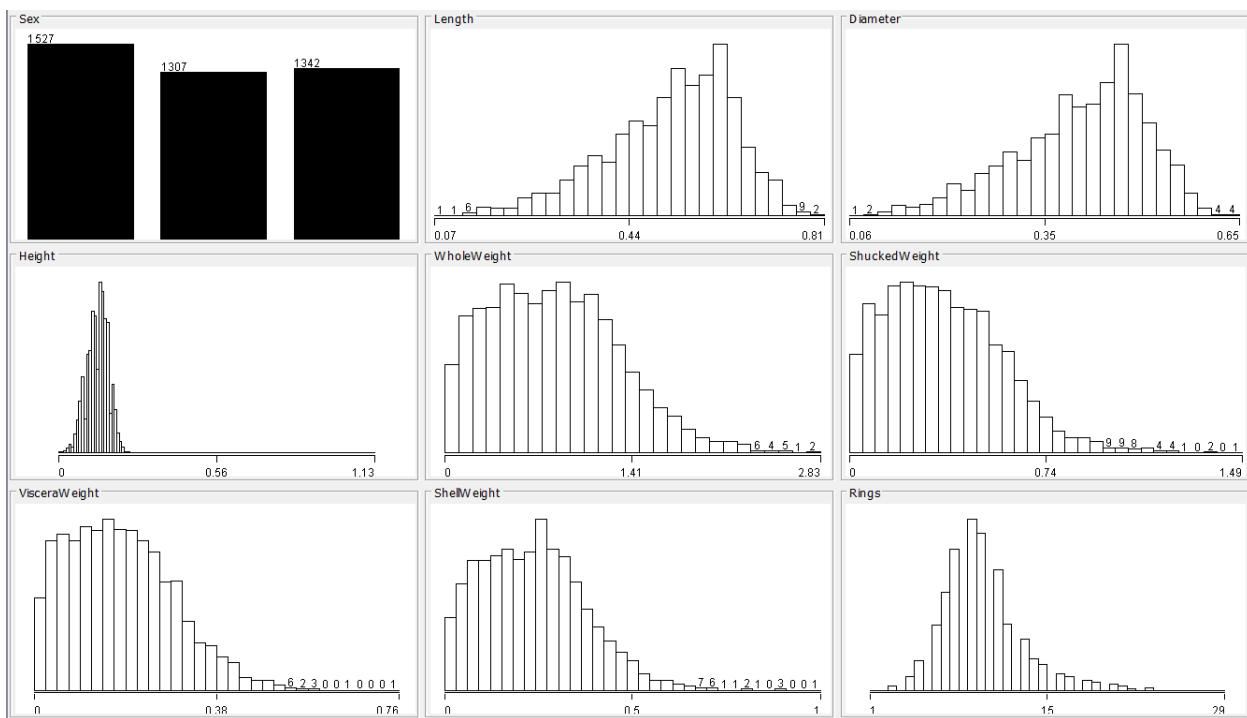
Abalone are marine snails that primarily inhabit cold coastal waters. The data collected in this dataset comes from abalone that inhabit the North Coast and Islands of Bass Strait in Australia, which is located just south of the city of Melbourne.

The dataset¹ uses 8 attributes to predict the number of rings on the abalone's shell. The number of rings can further be used to determine the age of the abalone, which is the main purpose of this dataset because the process of manually counting rings has many disadvantages. Below is a table of the attributes of the dataset, as shown in the "Abalone.names" file provided on the web page alongside the data.

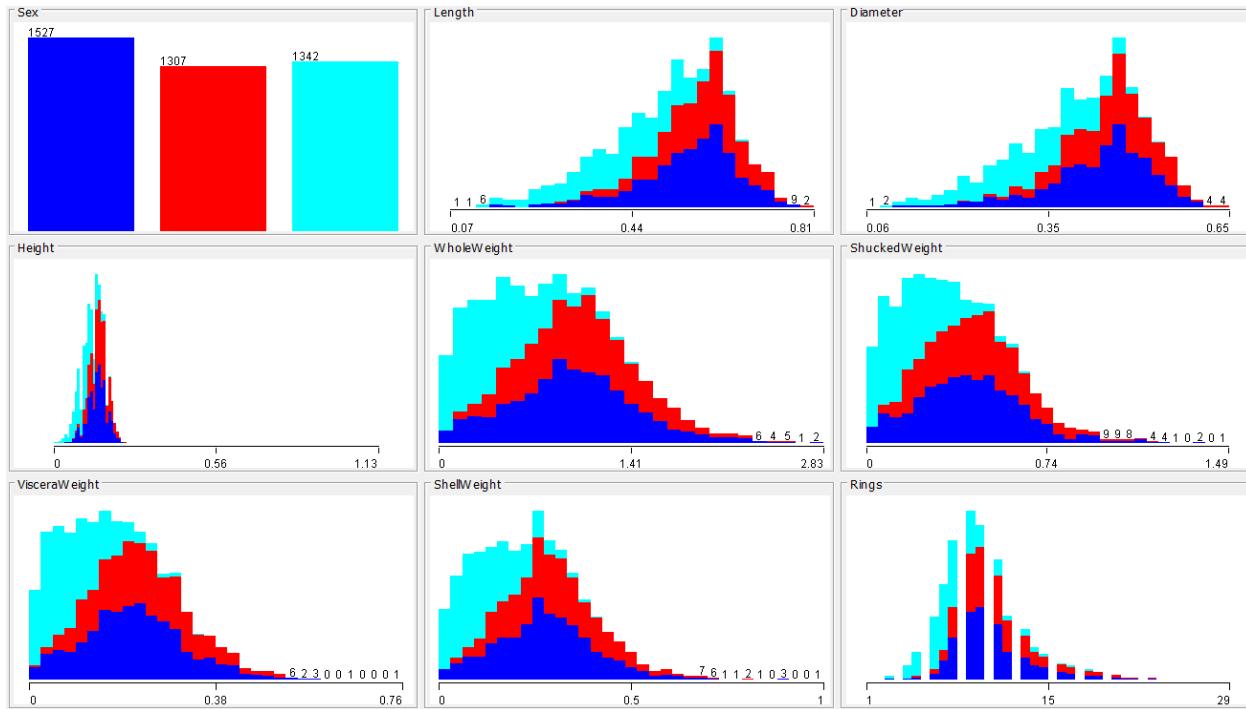
<u>Attribute</u>	<u>Datatype</u>	<u>Measurement</u>	<u>Description</u>
Sex	nominal	M, F, and I	(male, female, Infant)
Length	continuous	mm	longest shell measurement
Diameter	continuous	mm	perpendicular to length
Height	continuous	mm	with meat in a shell
Whole weight	continuous	grams	whole abalone
Shucked weight	continuous	grams	weight of meat
Viscera weight	continuous	grams	gut-weight (after bleeding)
Shell weight	continuous	grams	after being dried
Rings (class)	integer	rings on shell	adding 1.5 gives the age in years

The dataset in its original form had contained missing values, mainly belonging to the class column. However, the dataset was edited by the creator prior to uploading on the UCI database, which resolved this issue, and the dataset used for this report contained no missing values at all.

Below are histograms of attributes plotted with regards to their frequencies; WEKA Explorer was used to provide this preliminary visualization of the data.



Attributes frequency (no sex coloring)

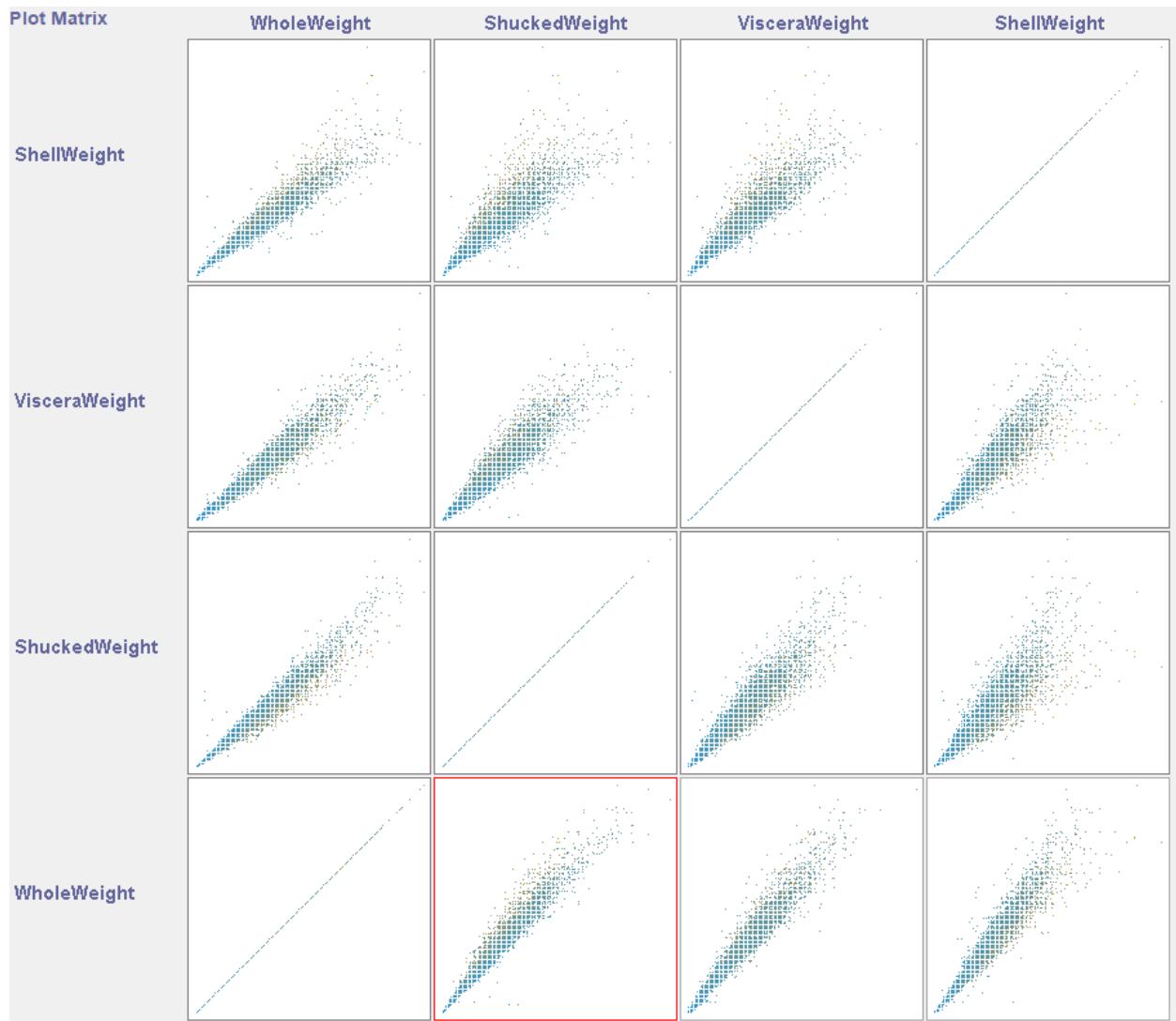


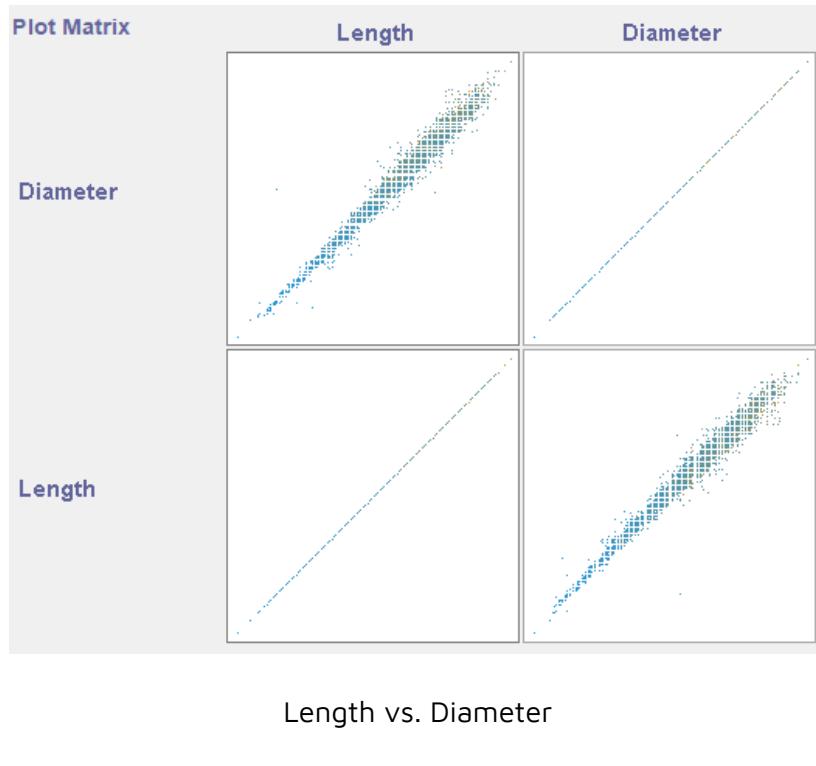
Attributes frequency (with sex coloring)

Upon closer observation, there seems to be a high correlation between some of the attributes. Attributes "length" and "diameter" roughly follow the same trend. They are rising steadily until they peak somewhere at the end of the 3rd quarter.

It is very apparent that the attributes "whole weight", "shucked weight", "viscera weight", and "shell weight" show signs of correlation as well. All four attributes start with a large value, maintain the same level throughout the 1st quarter, only to start declining at the beginning of the 2nd quarter and maintain a very low level until the very end.

The two figures below and on the next page show the attributes we suspect to be correlated, plotted against each other in order to visually demonstrate our claims.





Length vs. Diameter

Such high correlations are critical because they may skew the algorithm's accuracy: statistical redundancy such as that observed here can essentially double the weight of a value. Discarding the redundant attribute is often essential for accurate classification. These observations serve as a rough first look; they will be further addressed and analyzed later on using various tools provided by WEKA Explorer.

Data Analysis

(Predicting the number of rings)

Algorithms

We ran 15+ classification algorithms and recorded the relative absolute error and the root relative squared error into a spreadsheet, accessible at [this link](#). Both these metrics are important measures of the relationship between the predicted class and the actual class. There is a difference between these measures (usually within one or two percentage points), but both are error measures normalized by dividing by the error given by a simple predictor. When ranking the overall performance of a given classifier, we used the mean of the two metrics (which we will refer to as the "average error" going forward).

All attributes are numerical except for the sex attribute. Therefore, the data needs to be discretized before certain algorithms can be run on it, because such algorithms only take nominal classes as input. RegressionByDiscretization meta-classifier was run in order to obtain the desired results. The algorithms that this discretization process was applied to include ZeroR, OneR, NaiveBayes, J48, and Vote. Here is a table showcasing how well this selection of algorithms did.

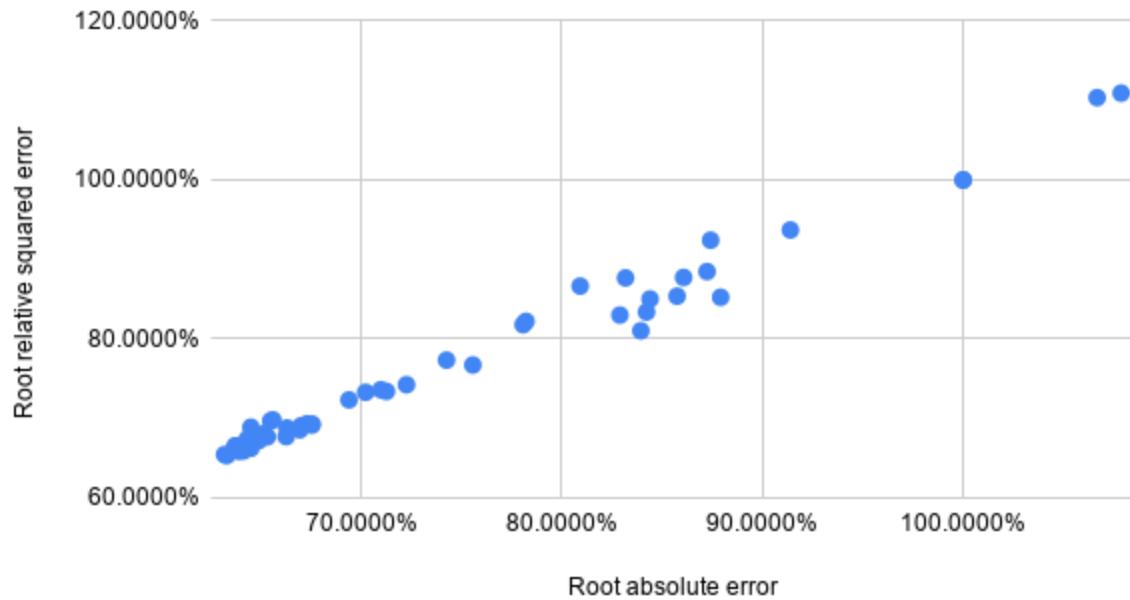
algorithm	Relative Absolute Error	Relative Root Squared Error	Additional comments
ZeroR	100.0016%	100.0009%	ZeroR predicts class value: '(6.6-9.4]'
OneR	80.9174%	86.6501%	OneR picks "Diameter" as the best attribute
NaiveBayes	107.8740%	110.9197%	Performed worst out of all algorithms including ZeroR.
J48	78.0761%	81.7980%	Performs decently well.
Vote*	63.3089%	65.3459%	The best classifier tested.

*this metafunction was applied to the LMT and Logistic Regression classifiers

All test options were run on 10-fold cross-validation as it was shown in class to be the most reliable and unbiased test model.

In terms of the average between RAE and RRSE, our results ranged from 64.3274% (an undiscretized dataset classified by the Vote metafunction applied to the M5Rules and M5P tree algorithms) to 109.3969% (NaiveBayes run on a discretized dataset). On the following graph, we have plotted all tested classifiers' RAE and RRSE statistics.

RRSE vs. RAE



(Please note that neither axis starts at zero, and the grid lines are 10 percentage points apart on the x-axis but 20 percentage points apart on the y-axis).

We see here that although there is a large range of inaccuracy, the 26 most accurate are all clustered close together in the 63-68 and 65-70 range (for RAE and RRSE respectively).

Attribute selection algorithms

Two methods of attribute selection were applied: gain ratio attribute selection and correlation-based feature subset attribute selection. Gain ratio attribute selection was only able to run on nominal-class datasets, so it was deployed in conjunction with a discretization function.

Running both of these selection methods on a discretized version of the data with the logistic model tree classifier gave an average error of 71.7606% for CFSubset (~6.6 percentage points lower than without any attribute selection) and 65.1405% for gain ratio attribute selection (the same value as without any attribute selection). The gain ratio algorithm ranked the attributes in this order (from most to least relevant): "shell weight", "height", "diameter", "whole weight", "viscera weight", "length", "shucked weight", and "sex". The CFSubset algorithm selected the "sex", "diameter", "height", "viscera weight", and "shell weight" attributes, leaving out length and shucked weight. The identical error between no attribute selection and gain ratio attribute selection implies that the ranking function of the gain ratio attribute selection either has *no* impact, or has the same impact of the standard logistic model tree classifier (i.e. the tree construction does not change).

Running a classifier subset evaluator on the dataset with an M5P tree takes out "length" and "viscera weight" attributes, implying redundancy. Curiously, running an M5P tree after manually deleting these attributes gives a relative absolute error of 63.7802% and root relative squared error of 66.1582% (compared to 64.036% and 66.3545% with all 9 attributes included). This is by far the best result without using any metafunctions, and the fourth best result overall.

The CFSubset attribute selection algorithm was also applied through a WEKA filter to the M5Rules algorithm (run on the original dataset without discretization),

yielding an average error of 76.1503% (10.95 percentage points higher than without attribute selection). The attributes selected by this algorithm were height and shell weight. Over 10 percentage points is a drastic difference in accuracy; perhaps understandable given the dataset's reduction to two attributes. When comparing this to the excellent results we obtained from manually deleting the two least relevant attributes, we observe that attention must be paid to keeping a good balance of attributes.

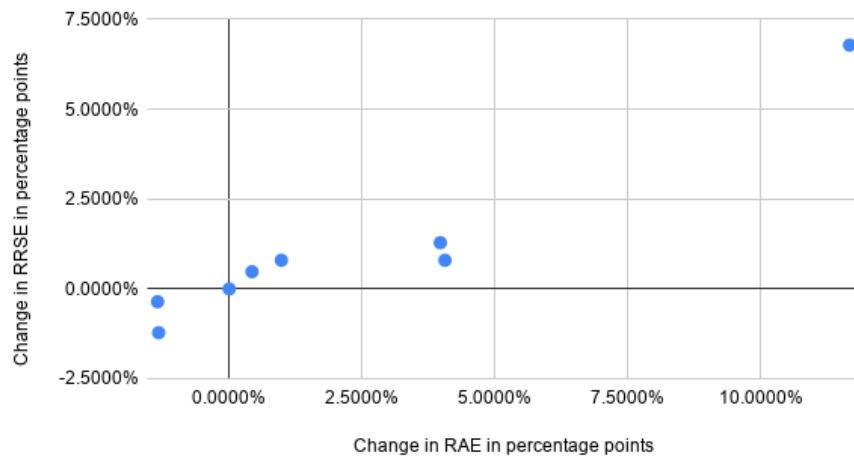
Conclusions

The best classifier found was the "Vote" metaclassifier used to combine the M5 rules and M5P tree classifiers on undiscretized data. It makes sense that strategically combining approaches of different types (rule classifiers and tree classifiers in this case) would improve accuracy, but it is also interesting to note that both classifiers use the M5 tree algorithm as a base. The worst-performing classifier tested was Naive Bayes performed on discretized data. We hypothesize that the inability to recognize ordinal relationships between class values in discretized data is particularly damaging to accuracy in this case, as Bayesian probability deals with the distributions of variables across classes. However, since metafunctions were used on only the best-performing classifiers, finding worse-performing classifiers would likely be trivial.

On a broader scale, we examine the effects of metafunctions and discretization. Below, we see a graph depicting the change in RAE and RRSE when

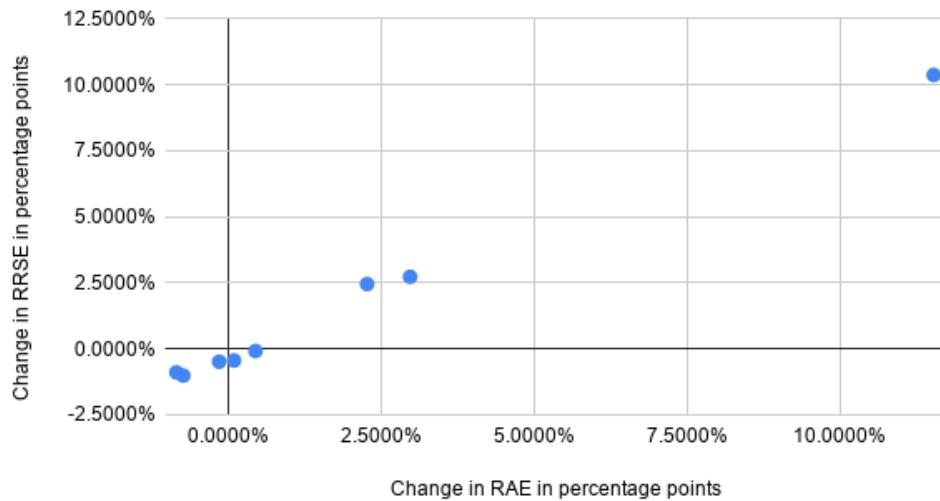
applying a discretization filter (for classifiers usable with both nominal and numeric classes).

Effects of Discretization On Functions



We see here that a large majority of classifiers perform the same or worse than they did on the undiscretized data; only two perform better (locally weighted learning and decision stump, neither of which perform particularly well either way). Discretizing data often dampens accuracy by removing orderedness from numerical attributes, and any benefits it may confer are offset by that loss of information.

Effects of Metaprograms Applied To Original Dataset



As depicted in the graph above, almost half of the metaprograms applied to the original dataset improve upon the best results from that dataset, and the other portion performs about as well or worse. (None of the metaprograms applied to discretized data improved upon the best discretized results.)

This shows that metaprograms are well worth trying (one bringing us the best results we found), with an impressive chance of improving results.

References

1. Dua, Dheeru and Graff, Casey. (UCI) Machine Learning Repository
<https://archive.ics.uci.edu/ml/datasets/Abalone>
2. NA. Arff stable - Weka Wiki
https://waikato.github.io/weka-wiki/formats_and_processing/arff_stable/