Quarter 1 Project Report: Chlorophyll-a Concentrations in Water Bodies Predictive Model

Group Members:  
Jacob Dipasupil, Petr Kisselev, Nikhil Alladi

# 

# Table of Contents

[1 Data Information 3](#_zemelijkcb7q)

[2 Model and Rationale 3](#_3ef5sa3dh4wg)

[3 Preprocessing 3](#_5ctj50sz9a9h)

[4 Attribute Selection Algorithms and Model Classifiers 5](#_5bpczdj7j02m)

[4.1 Attribute Selection Algorithms Used 5](#_mtzm31972e8v)

[4.2 Classifier Models 9](#_onnswtpdsn9e)

[5 Results and Analysis 10](#_uqpnysrcmyaq)

[5.1 Results 10](#_a6hocvcwihy8)

[5.1a Information Gain Attribute Selection 10](#_j373gyqnwmbf)

[5.1b Learner Based Classification Attribute Selection 11](#_lwouqi6jafk8)

[5.1c Principal Component Analysis Attribute Selection 13](#_7ljl8t6banjc)

[5.1d OneR Classifier Attribute Selection 15](#_rxv6qmigaij6)

[5.1e Subjective/Hand-picked Attribute Selection 17](#_t92grxy7lz9s)

[5.2 Analysis 19](#_2m951gh67lla)

[6 Conclusion and Reproduction 20](#_oay4xq3s84qa)

[7 Team Members and Tasks Performed 20](#_jkwqs3252891)

[8 Appendix and Sources 20](#_hzegwz2xs9)

# 1 Data Information

Our dataset contains data compiled by the U.S. Environmental Protection Agency on certain characteristics of lakes[1]. The dataset has 67 different attributes, some of which include lake name, date of sampling, total phosphorus concentration, area of lake surface, monthly and yearly average precipitation across the watershed, annual average nitrogen from human waste, lake depth, log of chlorophyll-*a* concentration, and more. The meaning of every attribute is contained in the data dictionary found in the provided link to the dataset. Since there are 67 different attributes, and we are classifying a class attribute of our choosing, our dataset has a dimensionality of 66. There are 2,226 instances with 45 missing values for lake name, 41 missing values for both nitrogen concentration and phosphorus concentration, 75 missing values for depth, and 132 missing values for the log of chlorophyll-*a* concentration, our class. Since we are trying to classify chlorophyll-*a* concentration, we will have to remove those 132 instances where the values are missing. The distribution of data is slightly right skewed with a mean of 1.053 and a standard deviation of 0.563 ranging from 0.029 to 2.941. Since the log of chlorophyll-*a* is a numerical variable, we will discretize the data into three bins: low, medium, and high. The class distribution is quite heavily skewed to the right, with 1,084 instances in low (-inf-0.999795], 864 in medium (0.999795-1.970205], and 146 in high (1.970205-inf).

# 2 Model and Rationale

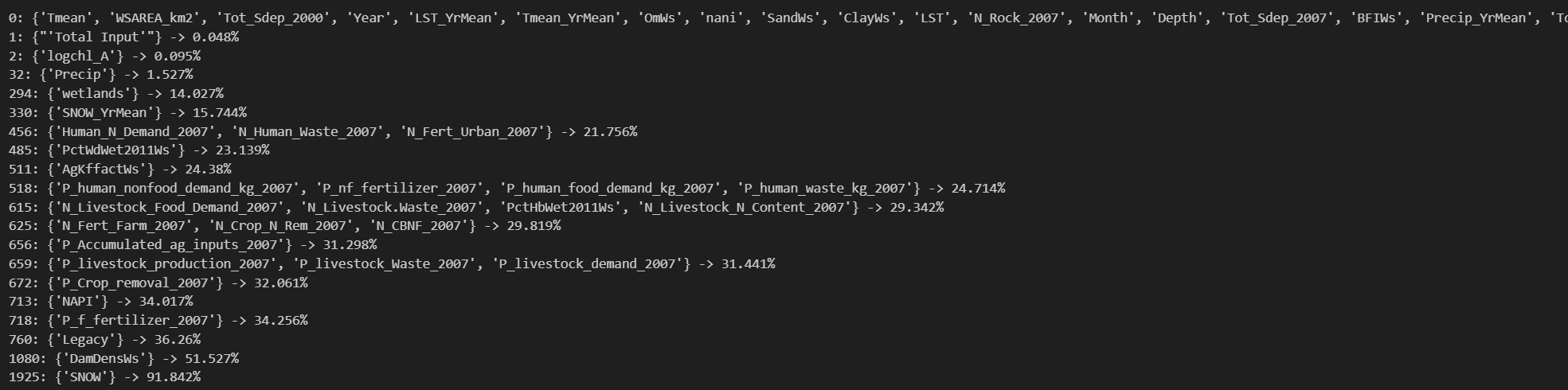
Our model will use data on lakes to predict if the concentration of chlorophyll-*a* is high, medium, or low in order to give us information about the state of the lake ecosystem. Chlorophyll-*a* concentrations can be used as a measure of the amount of algae growing in a water body and give us information on the trophic condition of a waterbody. High levels of chlorophyll-*a* concentrations and the subsequent algae growth can lead to harmful algal bloom, characterized by excessive algae growth producing toxins in water bodies, and hypoxia, which is when oxygen concentrations are too low for most organisms to survive in. Both of which are detrimental to the organisms living in and drinking from water bodies and can have harmful effects to the surrounding ecosystem. Being able to predict chlorophyll-*a* concentrations before permanent damage is done can help save some of these ecosystems.

# 3 Preprocessing

The first step of our preprocessing was done in Google Sheets. Many values in our dataset caused errors when trying to open in Weka. In order to allow Weka to open the dataset, all apostrophes in the *LAKENAME* attribute values were replaced with spaces. Additionally, there were 287 cells that contained one of the following values: “#NUM!”, “#DIV/0!”, “#VALUE!”. These obvious error values lead Weka to decide that certain attributes are string when they should be numeric attributes. To fix this we simply converted all data cells with those values into empty cells.

Pushing this data into WEKA, some further steps for preprocessing present themselves. To begin with, 130 instances in our dataset are missing values for our assigned class attribute *logchl\_A*. As supervised learning requires labeled class attributes, we removed these instances from our dataset (note that due to the values here being positive decimals, we set the split point to be above the maximum of 2.941 in this attribute so that the filter did not inadvertently remove valid instances as well). Additionally, we renamed the labels for the discretized class to low, medium, and high, instead of the ranges listed earlier. We also removed attributes that can be clearly reasoned to have no relation to the class attribute of any kind, including *LAKENAME, Survey Number, and SITE\_ID.*

Looking at this data in a spreadsheet view, we noticed that some attributes had a notably high amount of the value 0 in them. Due to their numerical basis, we took this 0 to be a default value, and analyzed the amount of zeros per attribute. In order to perform this analysis, we created a Python script using the Pandas library to load in the .csv version of our file that we got from the previous step and measure the percent of each attribute that consisted of zero values.

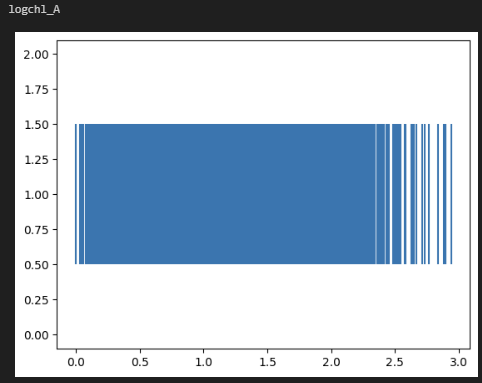
Doing so, we came to this result, seen below. 

We also analyzed the number of 0s per instance. This was also done using Pandas and Python on a Jupyter notebook file, resulting in this sort:



We chose to drop the *SNOW* attribute from this analysis as it had more than 70% of its values missing, and kept the instances intact.

Due to the extreme variance in magnitude of the data, we decided to normalize all attributes in the dataset. Some attributes had notable outliers, so we used z-score normalization for these. These attributes were as follows: *wsarea\_km2, lake\_area\_ha, fire, fire\_yrmean, lst, lst\_yrmean, precip\_yrmean, tmean, tmean\_yrmean, atmo\_pdep\_2002, atmo\_pdep\_2007, n\_cbnf\_2007, n\_crop\_n\_rem\_2007, n\_fert\_farm\_2007, n\_livestock.waste\_2007, n\_livestock- \_n\_content\_2007, p\_crop\_removal\_2007, p\_livestock\_demand\_2007, p\_livestock\_waste\_2007, p\_livestock\_production\_2007, p\_nf\_fertilizer\_2007, p\_human\_food\_demand\_kg\_2007, p\_human\_nonfood\_demand\_kg\_2007, p\_human\_waste\_kg\_2007, p\_accumulated\_ag\_inputs- \_2007, napi, total input'* [note that the ‘ here is not an accidental typo and is included in the name of the attribute], *legacy, damdensws, pcthbwet2011ws,* and *p2o5ws.* For all other attributes, we used min-max normalization. A quick plot of *logchl\_A* shows that the data is generally uniformly distributed, likely due to the log scale applied in this dataset (see below image), so we can use random sampling to split the dataset. We used 10-fold validation for this dataset, without it being stratified for the reason listed above. After preprocessing, we had this distribution:



# 

# 4 Attribute Selection Algorithms and Model Classifiers

After data cleaning and preprocessing, our dataset still had a dimension of 59. This is simply too large for classification algorithms to be used effectively as we would quickly run into issues typical of the curse of dimensionality. What this means is that the algorithms would have a hard time “finding” the trends within the data, as well as the model being far more complex and taking up more space. As the model would be more complex and take up more space, both in memory and in storage, the execution time would greatly increase as well. Thus, it is imperative that the dimensionality of the dataset is reduced. To do this we employed four attribute selection algorithms as well as choosings a set of attributes by hand through a subjective analysis. These attribute selection algorithms will be detailed below.

## 4.1 Attribute Selection Algorithms Used

### 4.1a Information Gain

For this approach to attribute selection, we used Weka for the computation. The concept of “gain” in a dataset means the amount of information that can be determined about one variable from another variable, randomly [4]. The methodology behind Information Gain attribute selection is as follows:

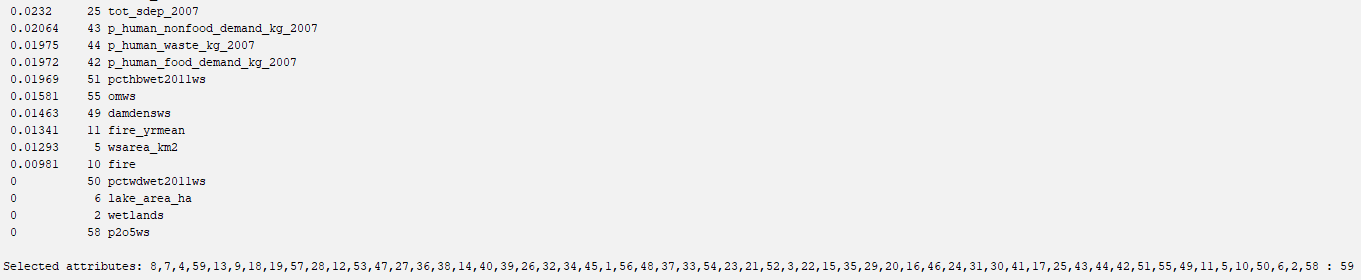
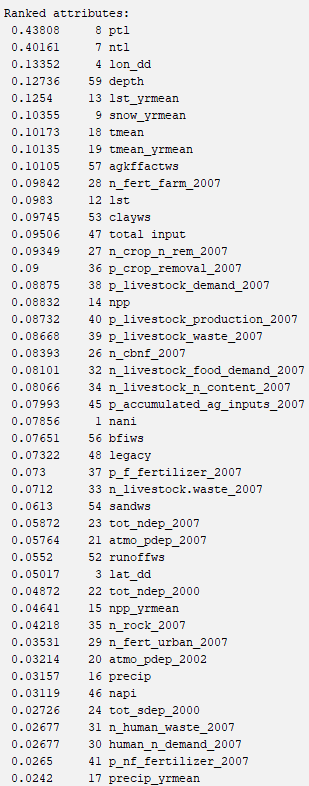
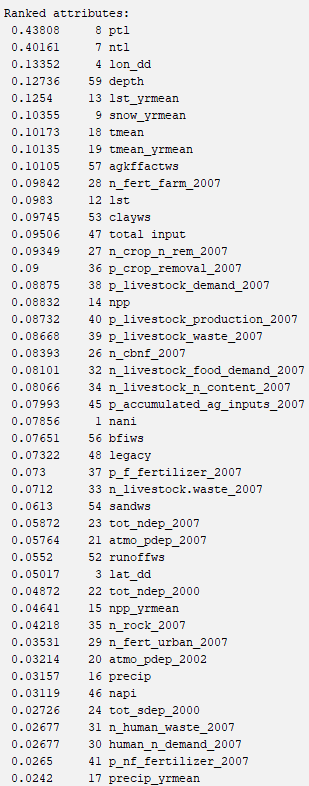
A given attribute is represented by *A*, *D* is the dataset and *pi* is the probability of a given tuple found in *D* to belong to the class *Ci*, *m* is the number of classes in the dataset

Information to classify a tuple in D:

Same, after splitting D in *v* partitions by attribute *A* where *Dj* is a given partition:

The gain of an attribute may be defined as:

Now, for every attribute in the dataset, its gain is calculated and they are ranked highest to lowest where the higher the gain, the better. The result of this can be seen below.



We chose a cutoff of 0.1, leaving us with nine attributes: *ptl*, *ntl*, *lon\_dd*, *depth*, *lst\_yrmean*, *snow\_yrmean*, *mean*, *tmean\_yrmean*, and *agkffactws*. The reason for this choice of cutoff is that is leaves us with a dimension of nine – neither too high as to induce the curse of dimensionality, nor too low as to leave no information for the classifier algorithms to use.

### 4.2b Principal Component Analysis

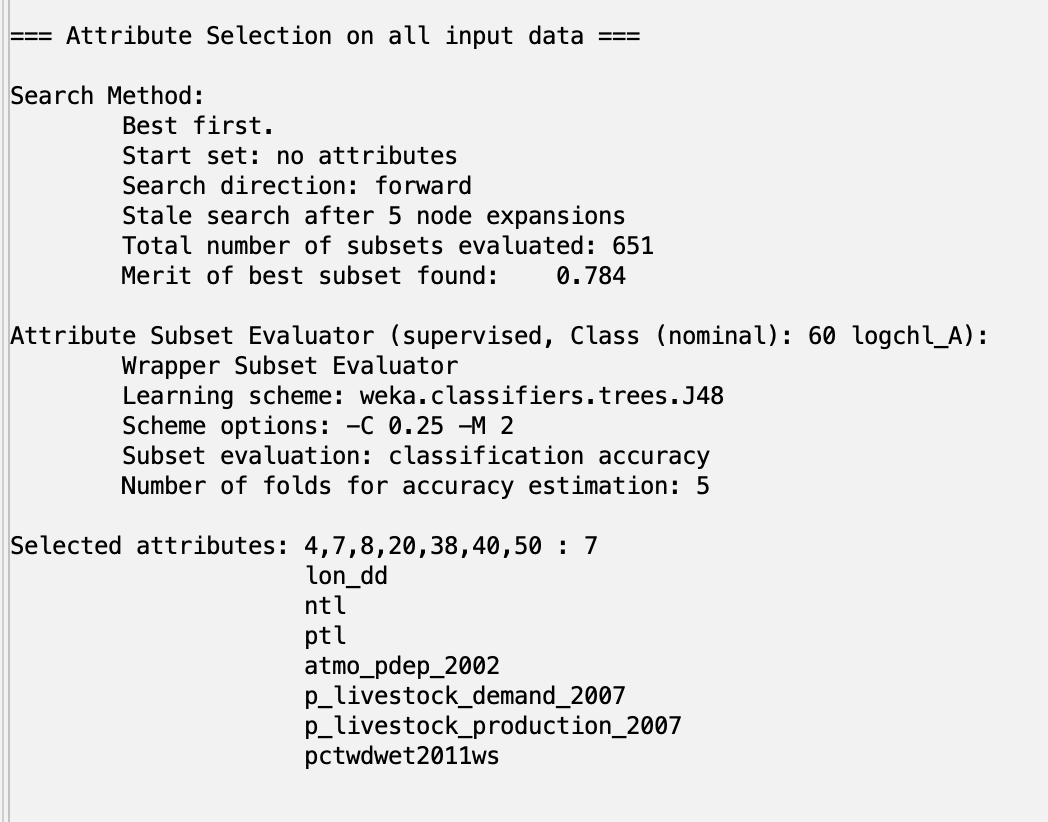
We once again used Weka here to perform all of the computations associated with PCA. In essence, what PCA does is that it transforms the initial dataset into a new one with new attributes where the values and attributes are selected in such a way as to maximize variance, with the highest variance attributes being ranked the highest [6]. The result of running PCA on our dataset can be found below.



Here we chose a threshold of 0.2, as this would leave with 11 attributes, not seriously deviating from the previous amount so as to maintain consistency and also continuing to walk the balance between the curse of dimensionality and oversimplification.

### 4.1c Learner Based w/ J48

Once again, we used Weka for this attribute selection algorithm. J48 is an open-source Java implementation of the popular C4.5 decision tree algorithm [7]. This algorithm utilizes gain as defined earlier to continually split the dataset and thus generate an effective decision tree. The algorithm then chooses the most important features for prediction and the results of this can be seen below.



### 4.1d OneR

For our final attribute selection algorithm, we chose OneR attribute evaluation. This algorithm produces a single rule for any given pairing of attribute and value and ranks these rules by accuracy to find the best one [5]. The pseudo code is below.

*For each attribute in the dataset*

*For each value in the current attribute*

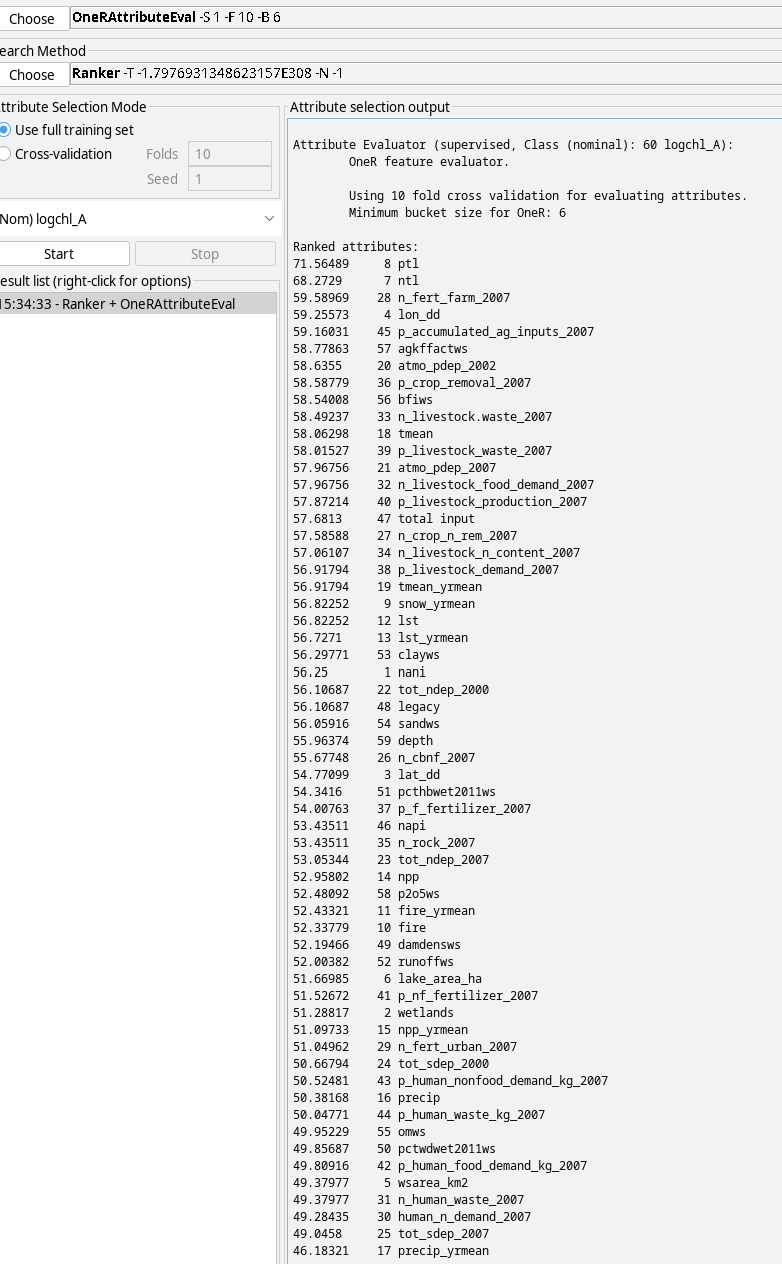
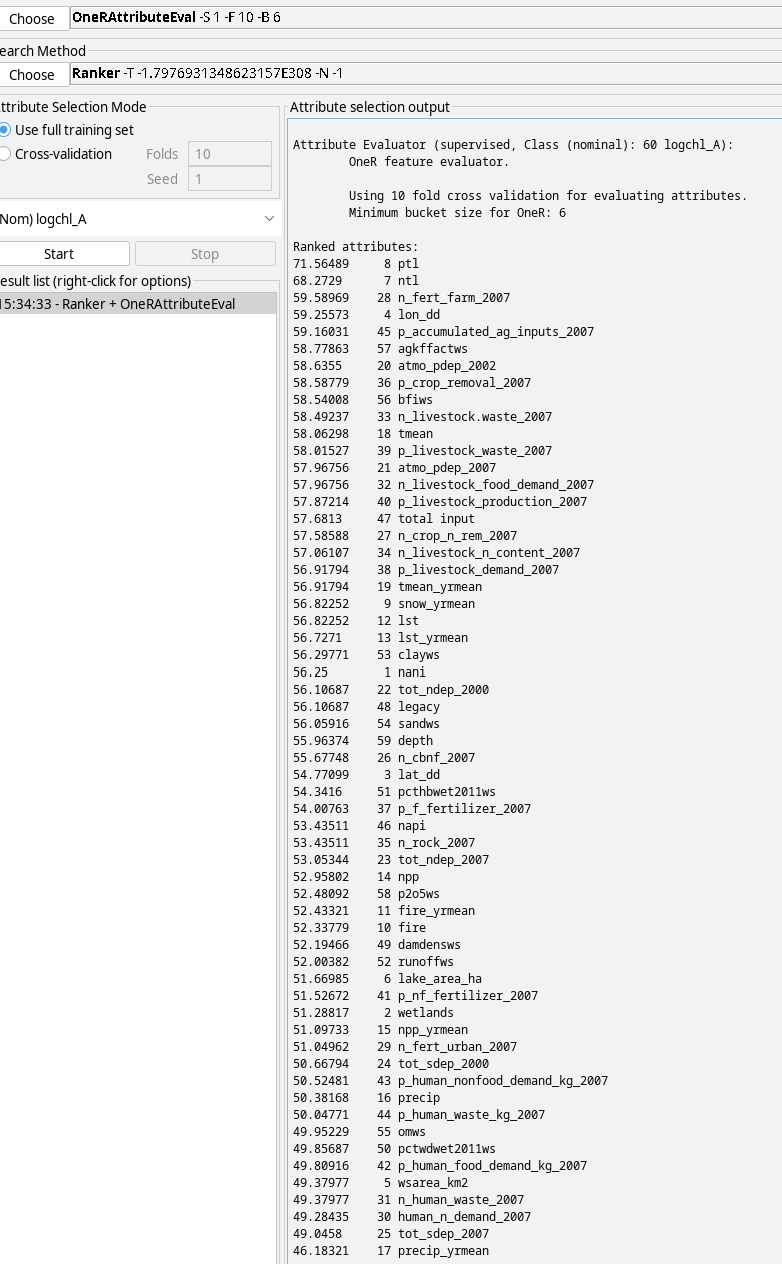
*Find most frequent class for given value of the given attribute*

*Create rule that assigns most frequent class to this attribute-value pairing*

*Compute error of feature by summing all rule error values*

*Rank attributes by error with lowest error rate being best*

The result of evaluating the features of our dataset using this algorithm are below.



We chose a threshold of 58.5, as this would leave us with 9 attributes – a reasonable amount and one that is similar to the amounts of attributes kept using other attribute selection algorithms, thus preventing too much inconsistency. We are left with *ptl*, *ntl*, *n\_fert\_farm\_2007*, *lon\_dd*, *p\_accumulated\_ag\_inputs\_2007*, *agkffactws*, *atmo\_pdep\_2002*, *p\_crop\_removal\_2007*, *bfiws*.

### 4.1e Subjective Analysis / Hand-picking

Finally, for the subjective approach we picked 10 attributes which we thought could contribute to chlorophyll-a levels. *lon\_dd*, the longitude of the lake would be important since longitude can indirectly imply climate patterns like distance from coastlines and the presence of mountain ranges. We also chose *lat\_dd*, the latitude of the lake, as it is another geographical measure and could give some indication as to the climate or overall temperature around a lake, as high latitudes are generally colder and drier. *ntl,* total nitrogen concentration, and *ptl,* total phosphorus concentration, would be a great indicator for algae growth as they both directly affect the rate of algae growth. Too much nitrogen or phosphorus will cause algae to grow faster than ecosystems can handle, which is what we are trying to prevent with chlorophyll-a concentrations. *atmo\_pdep\_2002,* the annual average phosphorus deposition in 2002, will also give our model more information on the amount of phosphorus in the lake. There is also *atmo\_pdep\_2007,* but we decided that only one indicator of phosphorus deposition would be necessary. *n\_human\_waste\_2007* and *n\_livestock\_waste\_2007* give the annual average of nitrogen from human and livestock waste*,* and  *p\_human\_waste\_2007* and *p\_livestock\_waste\_2007* give the annual average phosphorus from human and livestock demand. These four attributes were selected since we suspect that humans and animals farms have a large influence on the surrounding environments (i.e. lakes) and the phosphorus and nitrogen from our waste and demands are contributing to algae growth. Our last attribute selected was *runoffws*, the mean runoff within the given watershed, which we suspected contributed to algae growth as runoff would carry nutrients, like phosphorus and nitrogen, to the lakes to feed the algae.

## 4.2 Classifier Models

### 4.2a Naive Bayes

The Naive Bayes classifier works as follows:

Given a training set of labeled tuples, *D*, some tuple *X* with *n* attributes (*x1*, *x2*, *x3*, ..., *xn*), where *xi* is the value for the attribute *Ai* and *m* classes are represented by *C1*, *C2*, *C3*, ..., *Cm*

The probability of *X* belonging to class *Ck* can be predicted recursively as:

When assuming no dependence between attributes the formula can be simplified to:

This can be used to predict the class of an instance by performing this probability calculation on every class (*C1 -> Ck*) and taking the class with the highest probability as the prediction [9].

### 4.2b Logistic Regression

This algorithm works through the estimation of the parameters of a logistic model. This is somewhat similar to a linear regression except that the parameters are and *s* instead of *m* and *b*. In the logistic model controls the location of the midpoint while *s* controls the scale of the curve. These two values are optimized to minimize the error. After this, predictions are made by using the output of the model as the prediction and the input into the function as the input for the unlabeled instances [10].

### 4.3c Learner Based w/ J48

The J48 classification algorithm works in much the same way as the same attribute selection algorithm, except instead of using the decision tree to determine the most important attributes, the decision tree is used to actually predict the class for new instances[7].

### 4.4d RandomTree

The random tree algorithm is in a way similar to the J48/C4.5 algorithm in that it constructs a tree, but in this case at each node there are *k* branches. What gives the algorithm is “random” name is that at each node, it considers *k* attributes entirely randomly [8]. This then builds out into a decision tree that can be used for prediction of class on unlabeled data.

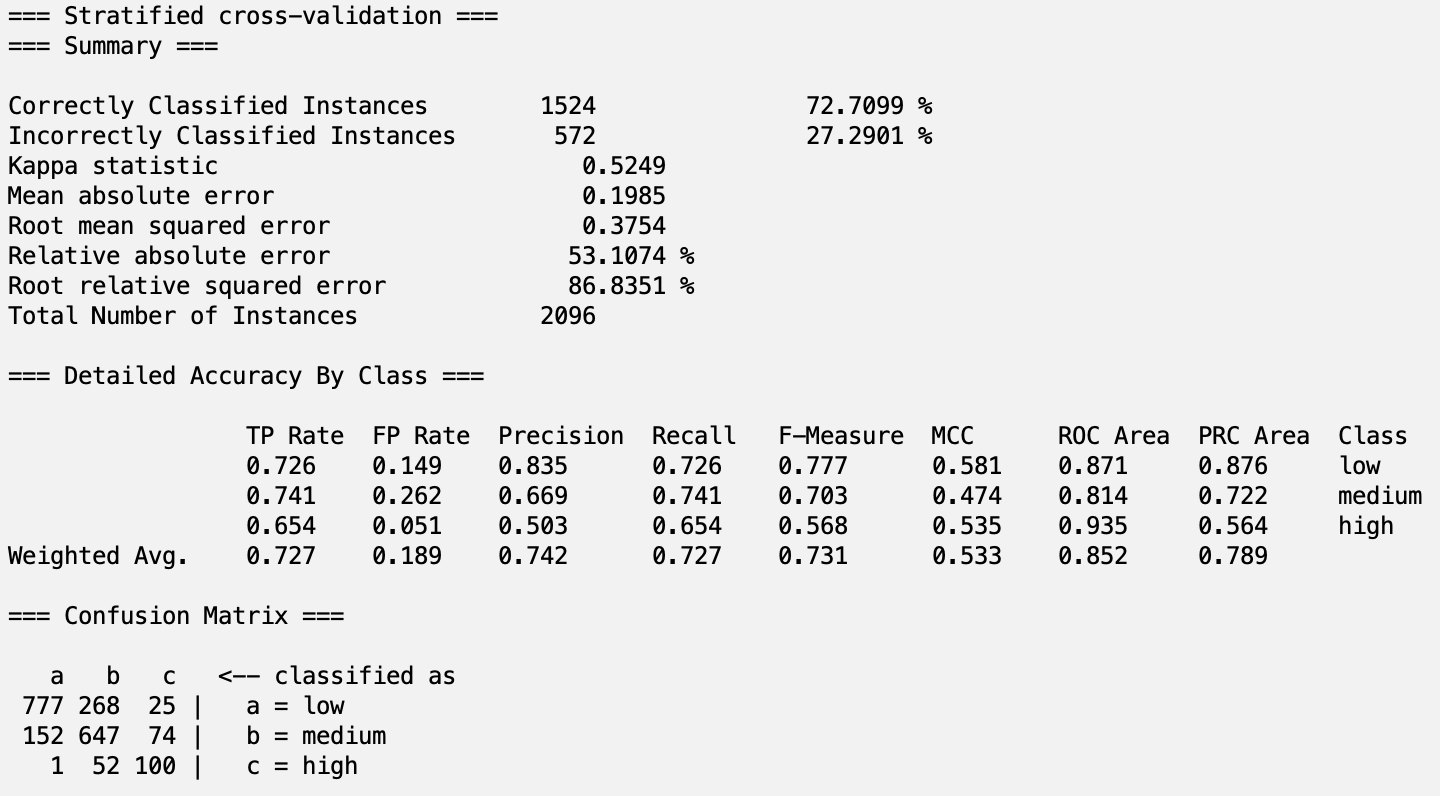
# 5 Results and Analysis

## 5.1 Results

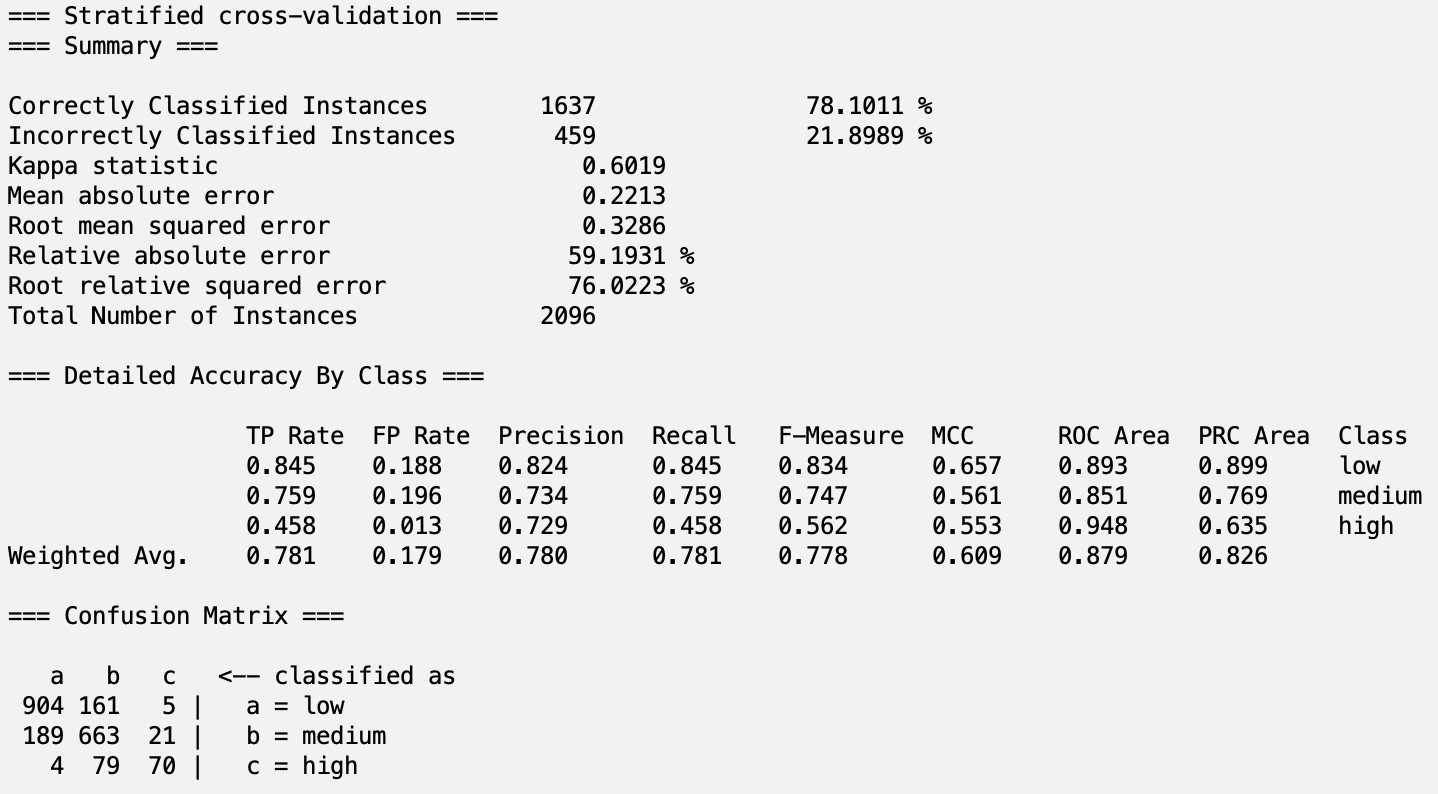
Results ordered by attribute selection algorithm:

### 5.1a Information Gain Attribute Selection

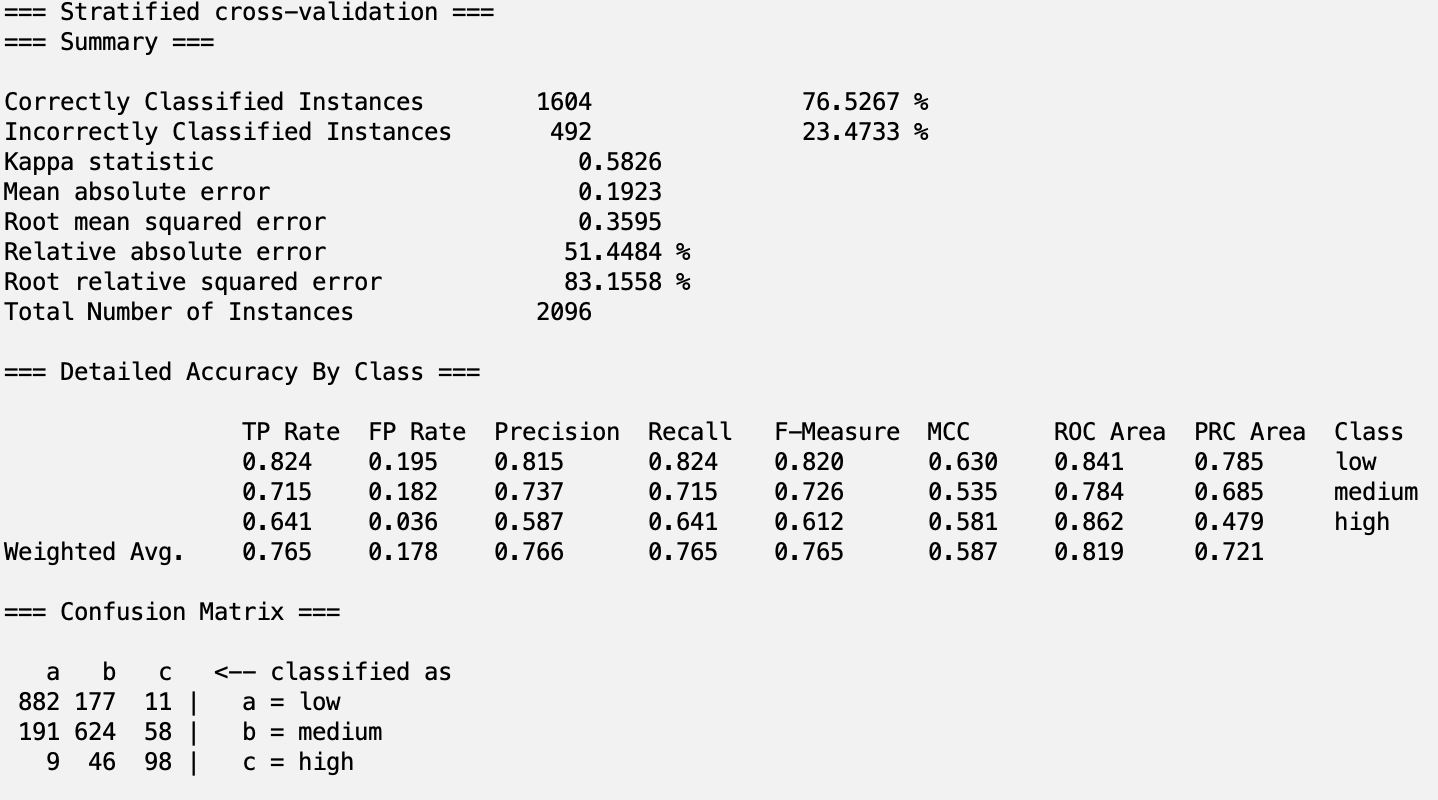
InfoGainAttributeEval with Naive Bayes



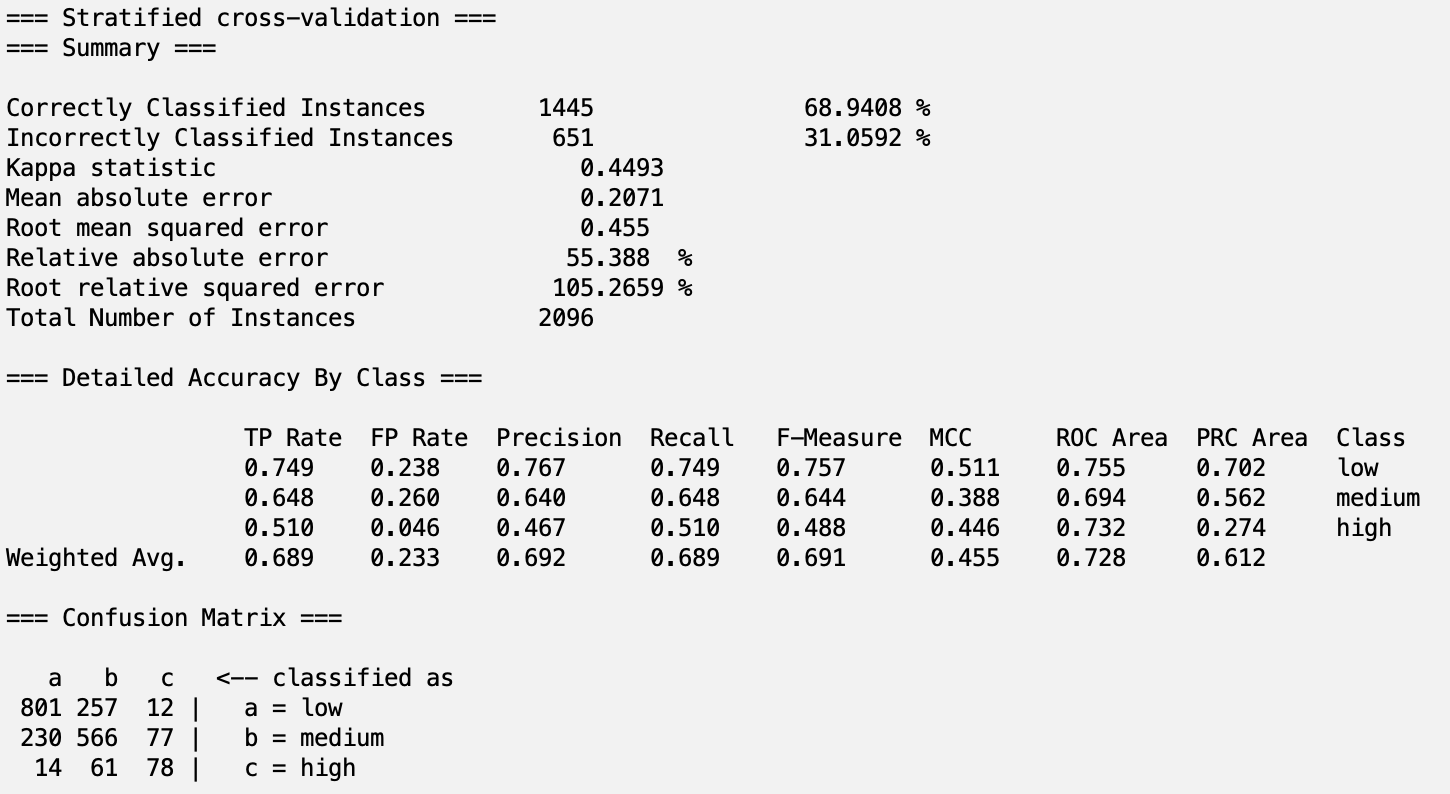
InfoGainAttributeEval with Logistic



InfoGainAttributeEval with J48



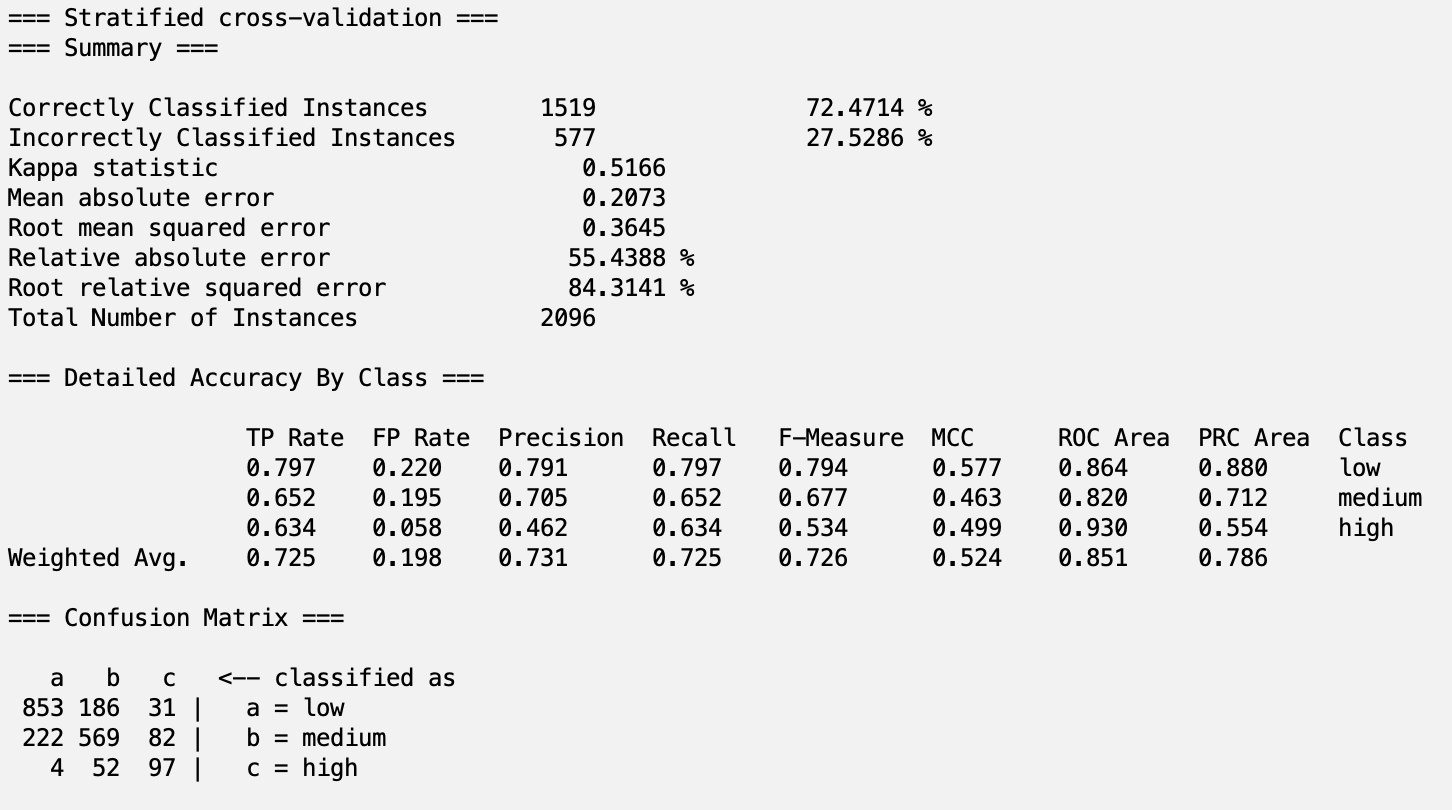
InfoGainAttributeEval with RandomTree



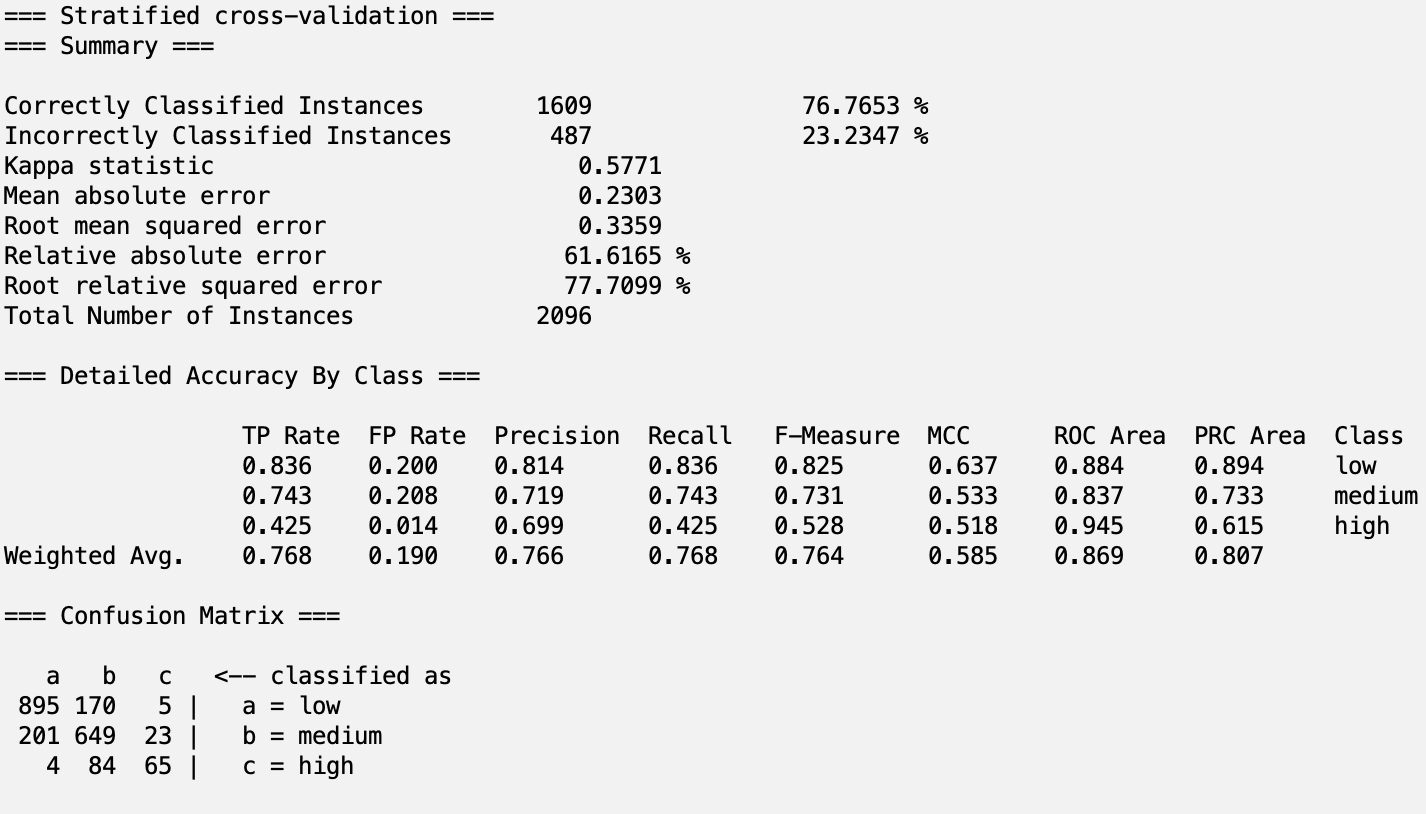
### 

### 5.1b Learner Based Classification Attribute Selection

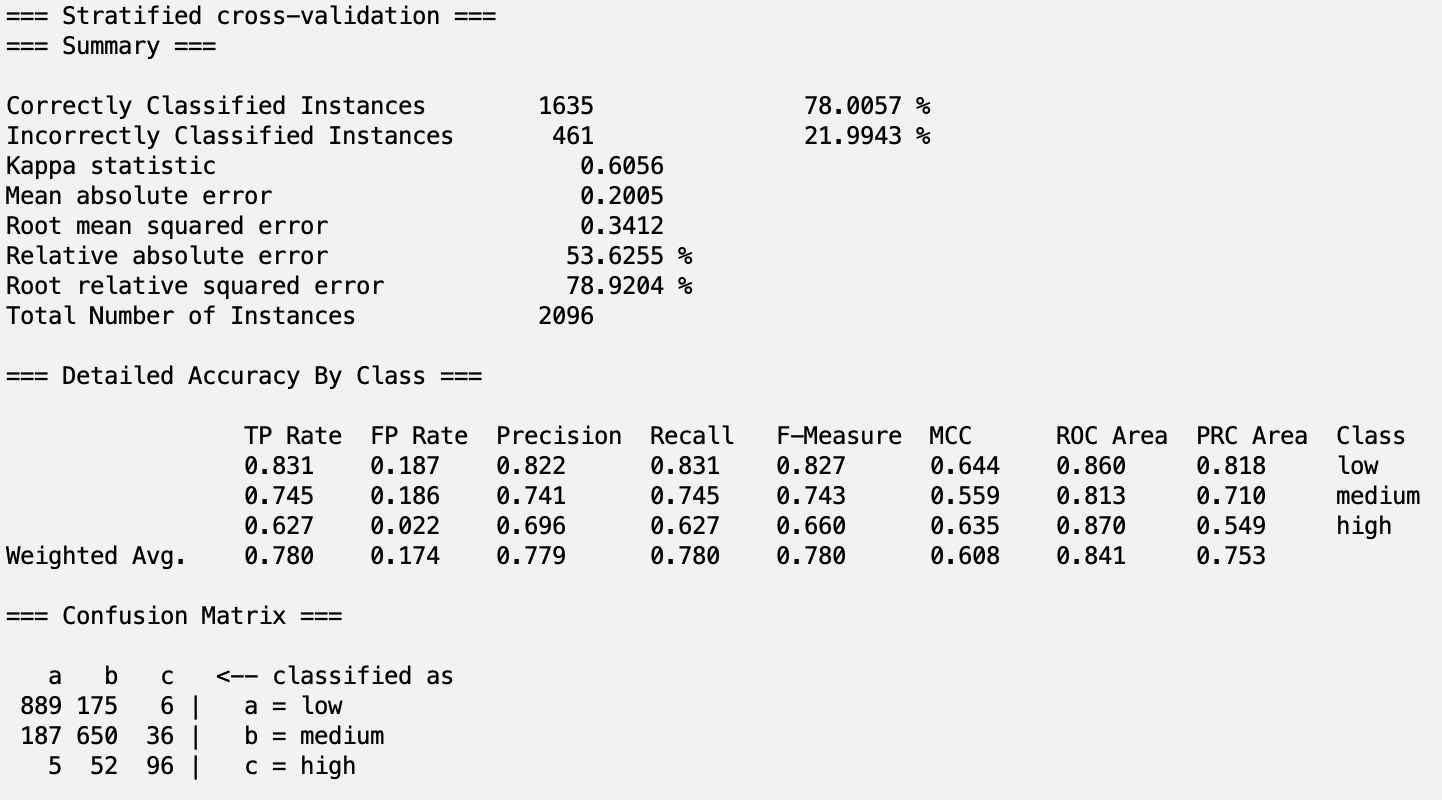
LearnerBased (J48) with Naive Bayes



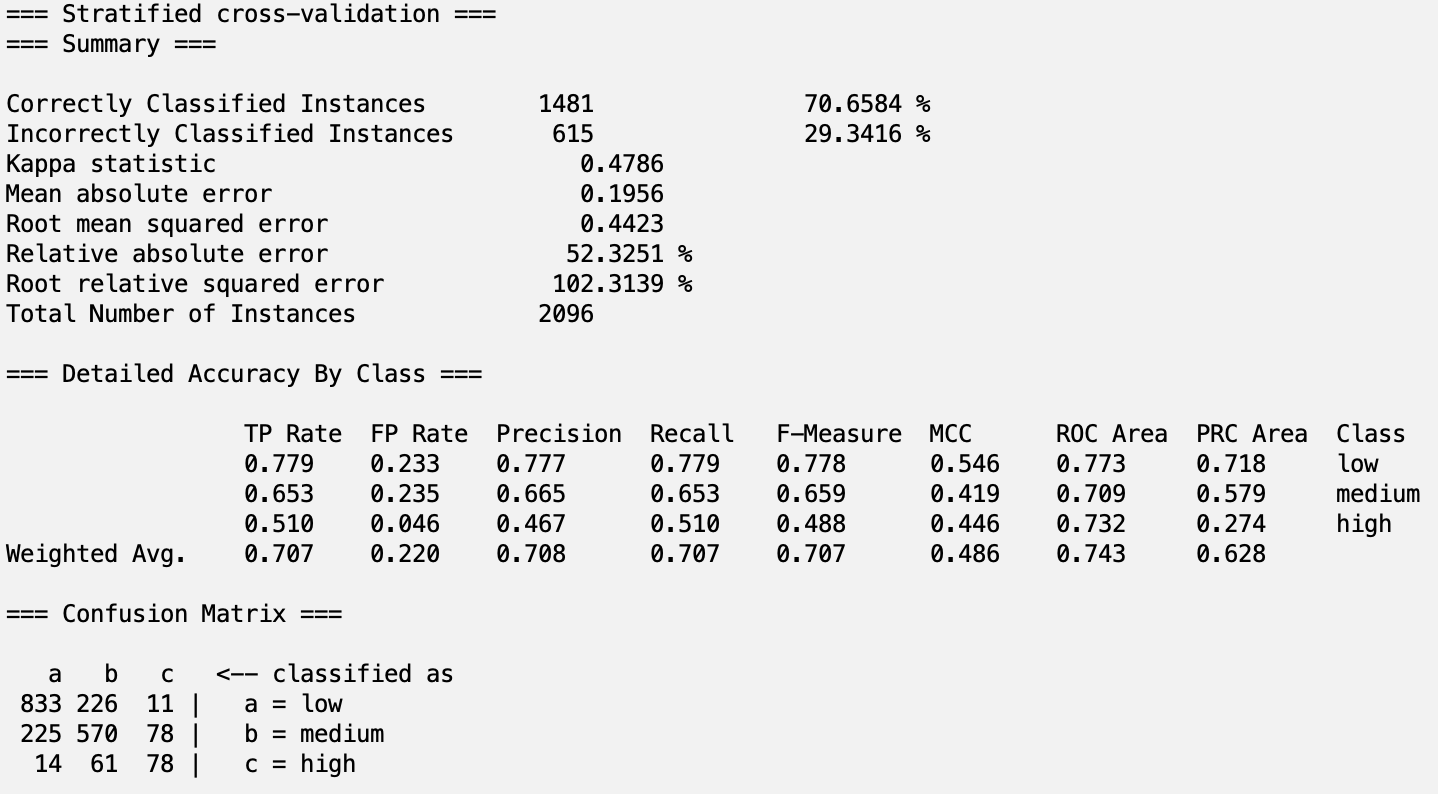
LearnerBased (J48) with Logistic



LearnerBased (J48) with J48

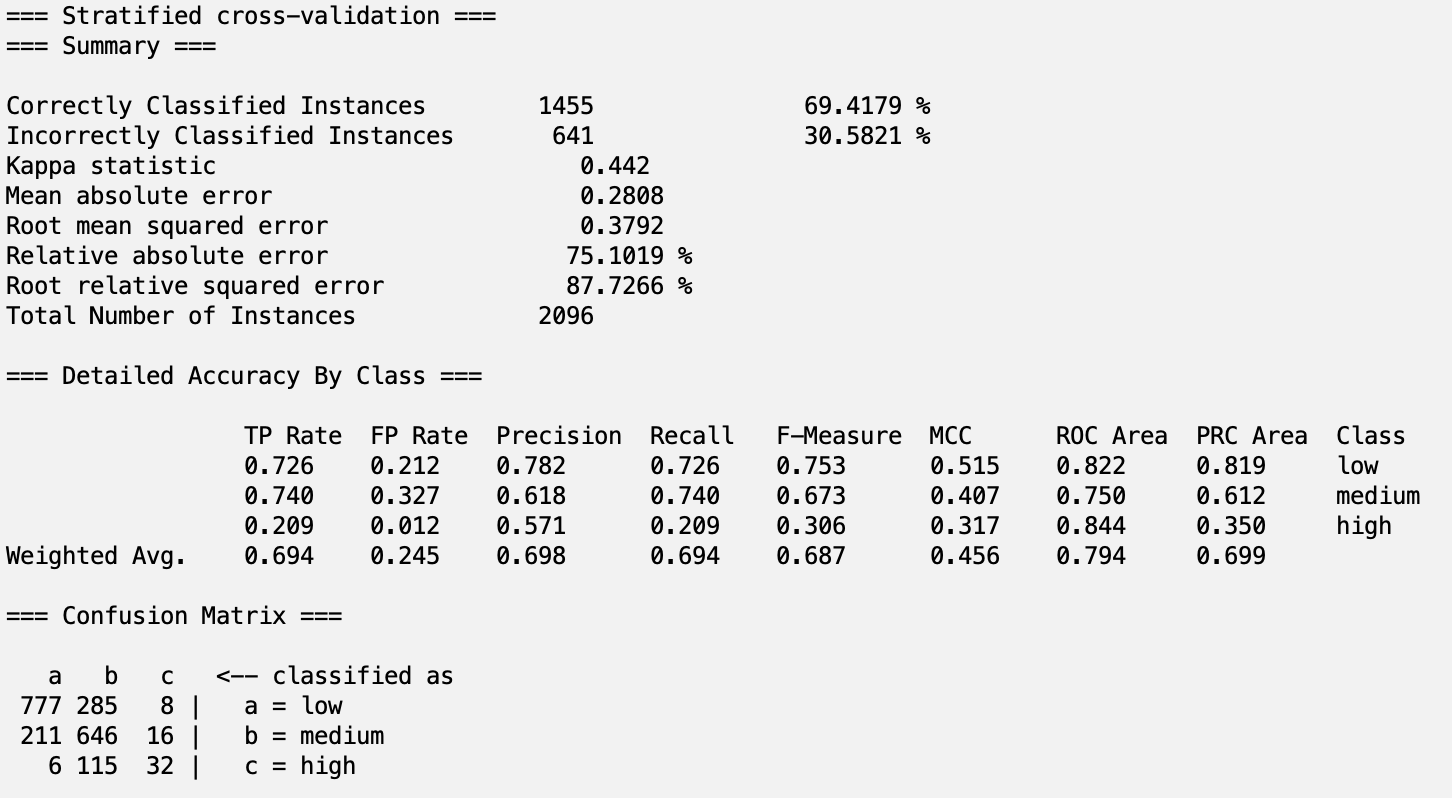


LearnerBased (J48) with RandomTree

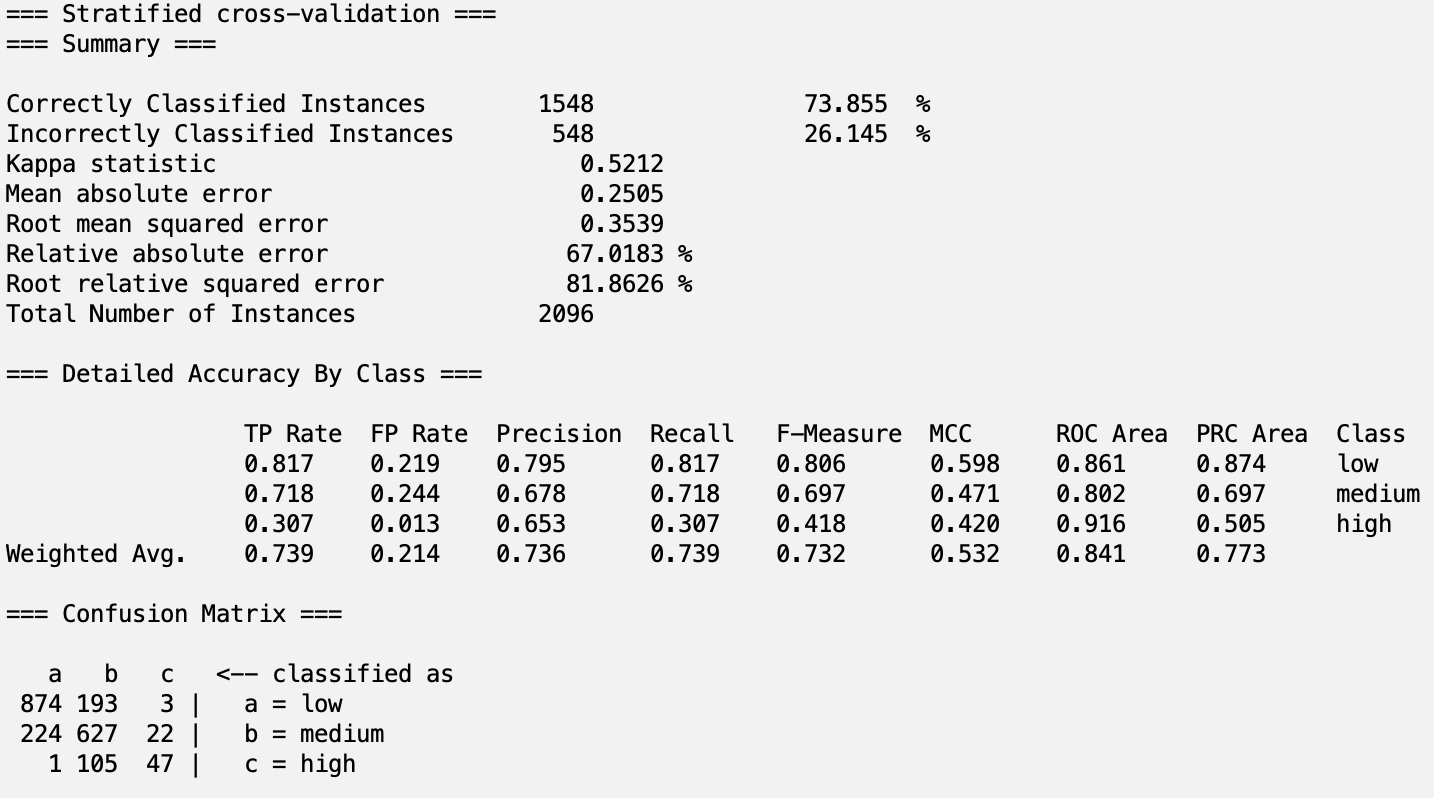


### 5.1c Principal Component Analysis Attribute Selection

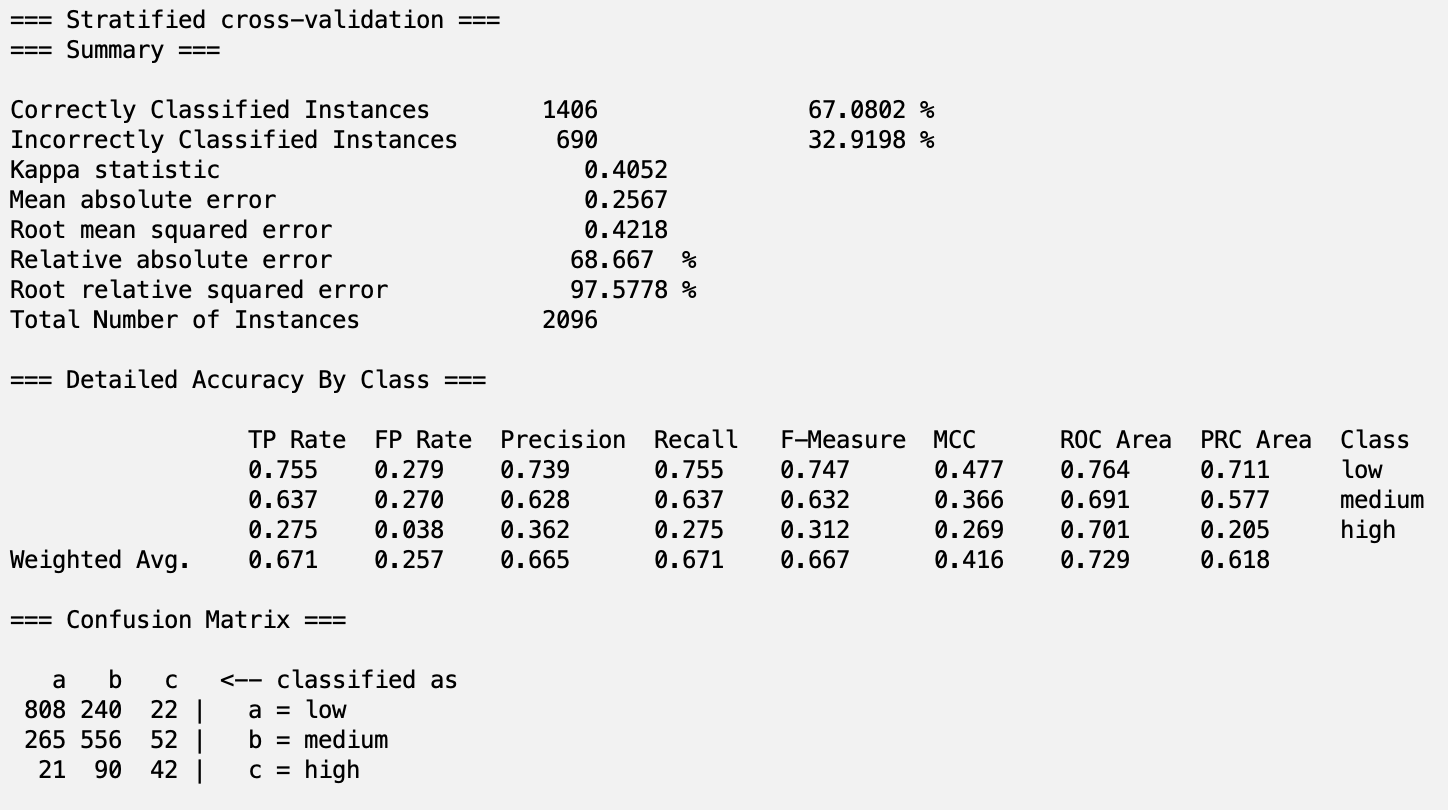
PCA with Naive Bayes



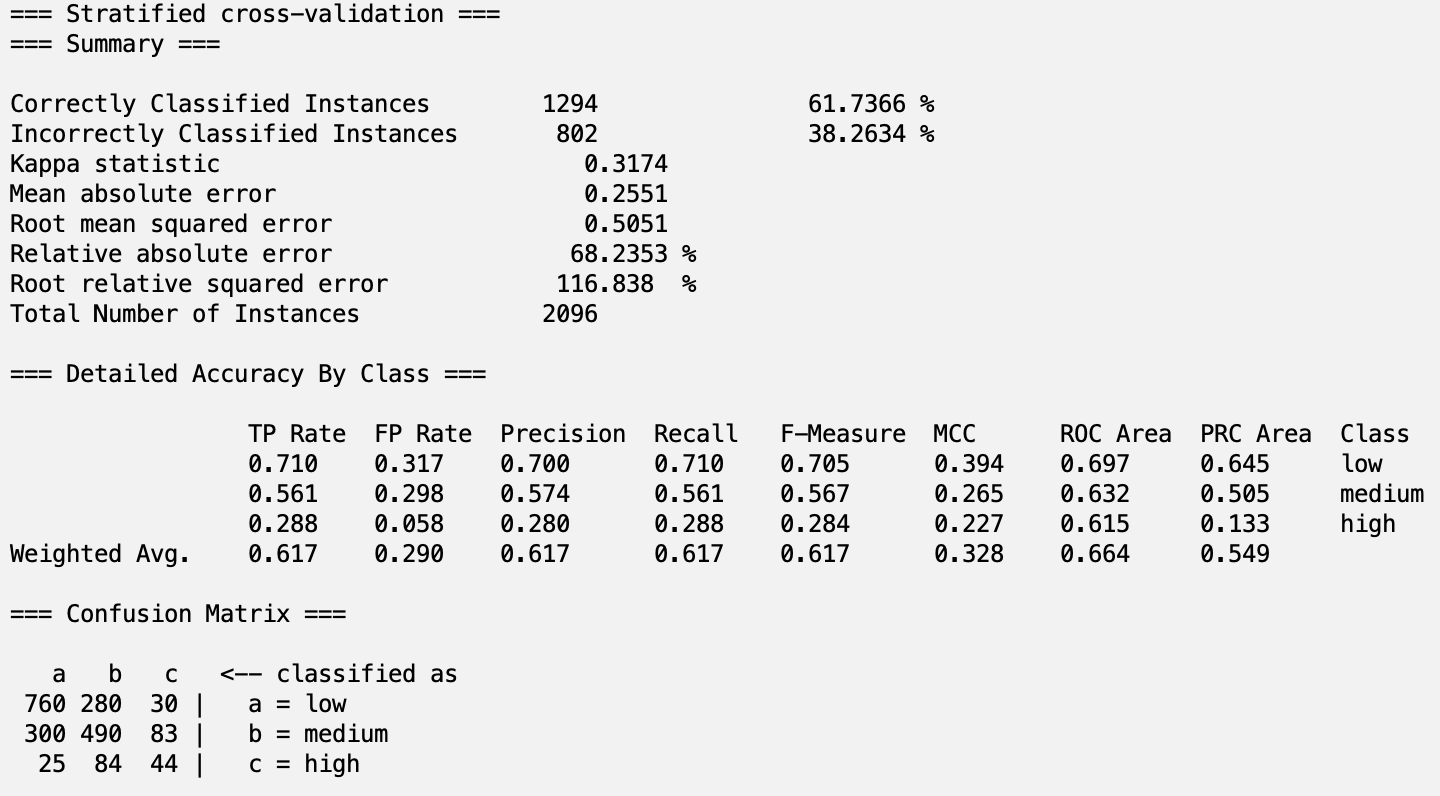
PCA with Logistic



PCA with J48

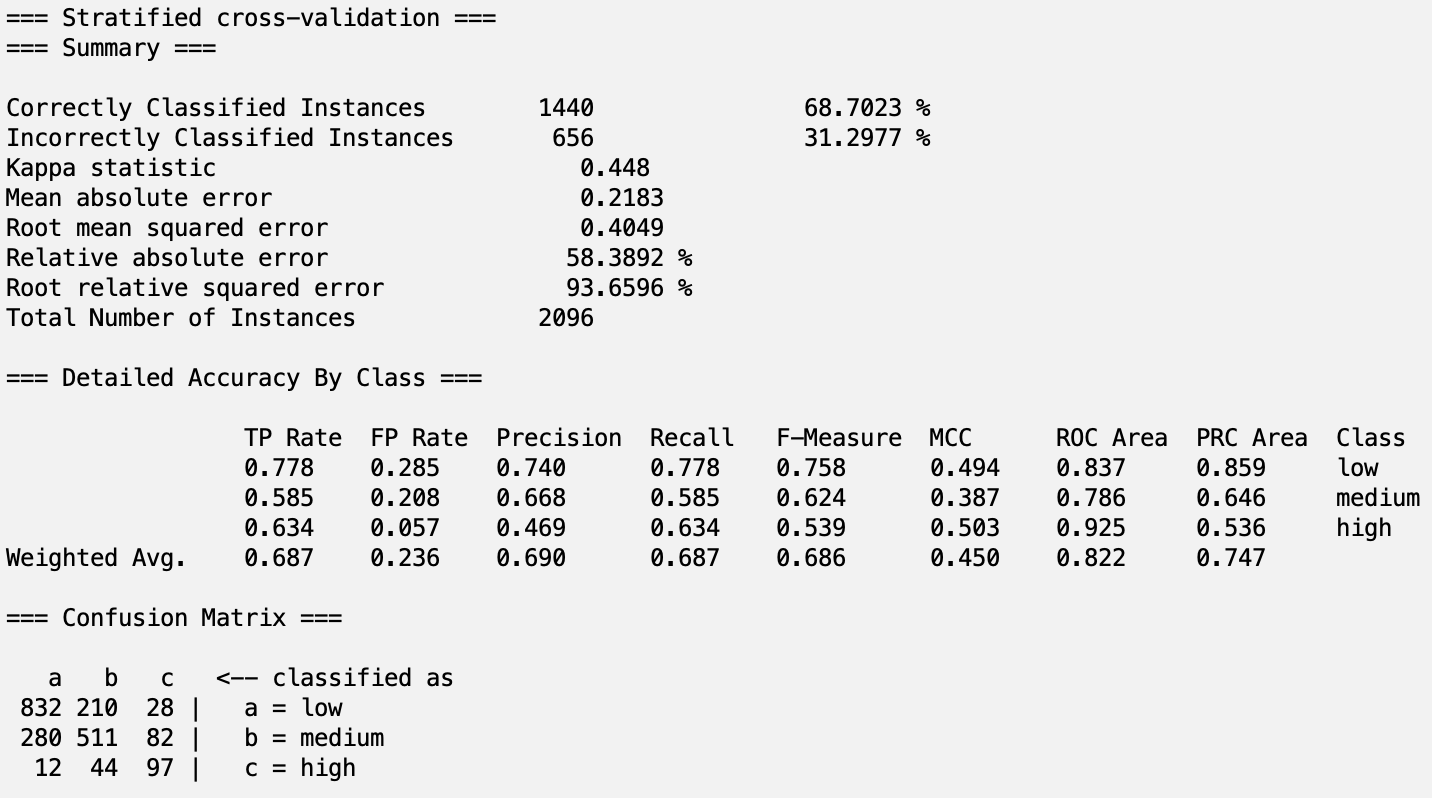


PCA with RandomTree

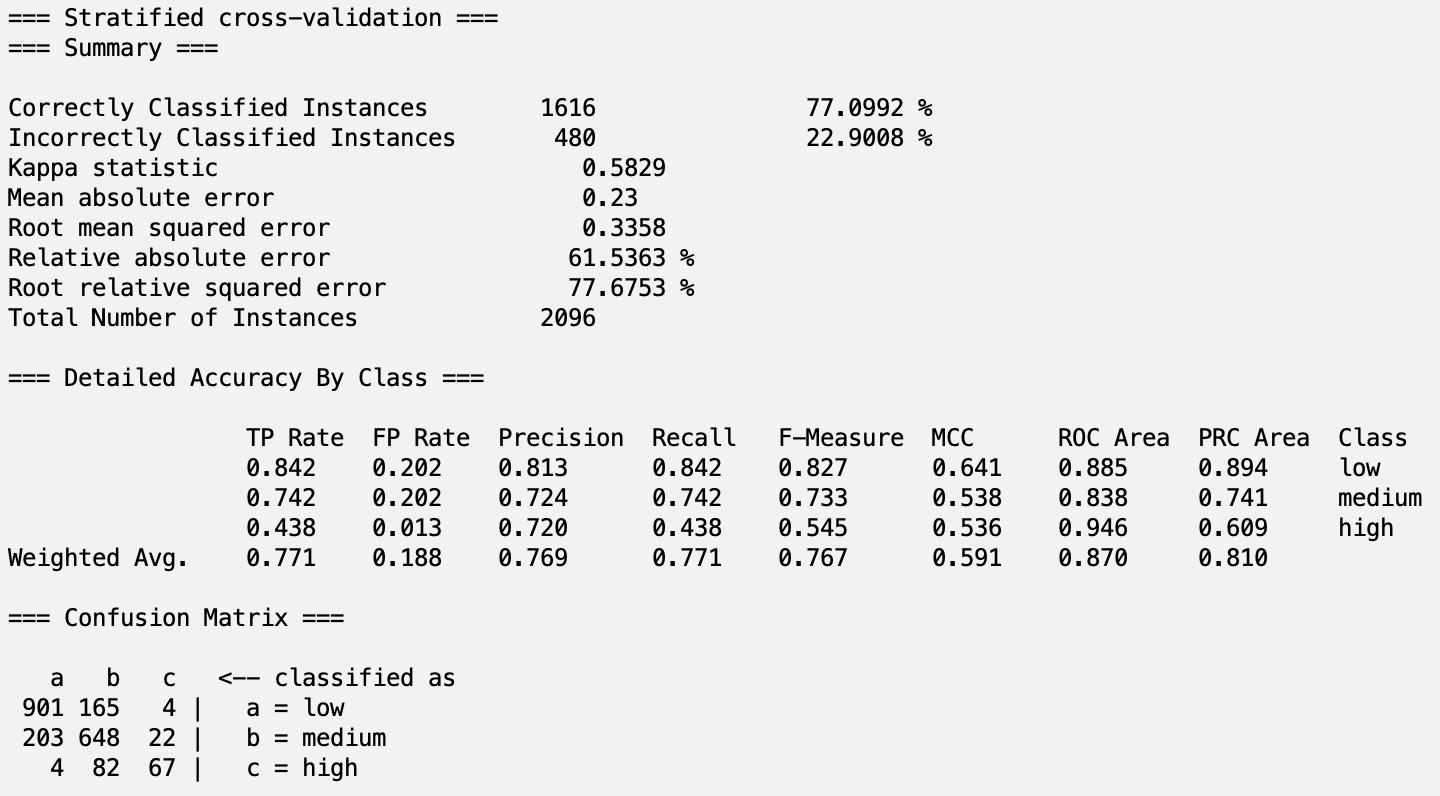


### 5.1d OneR Classifier Attribute Selection

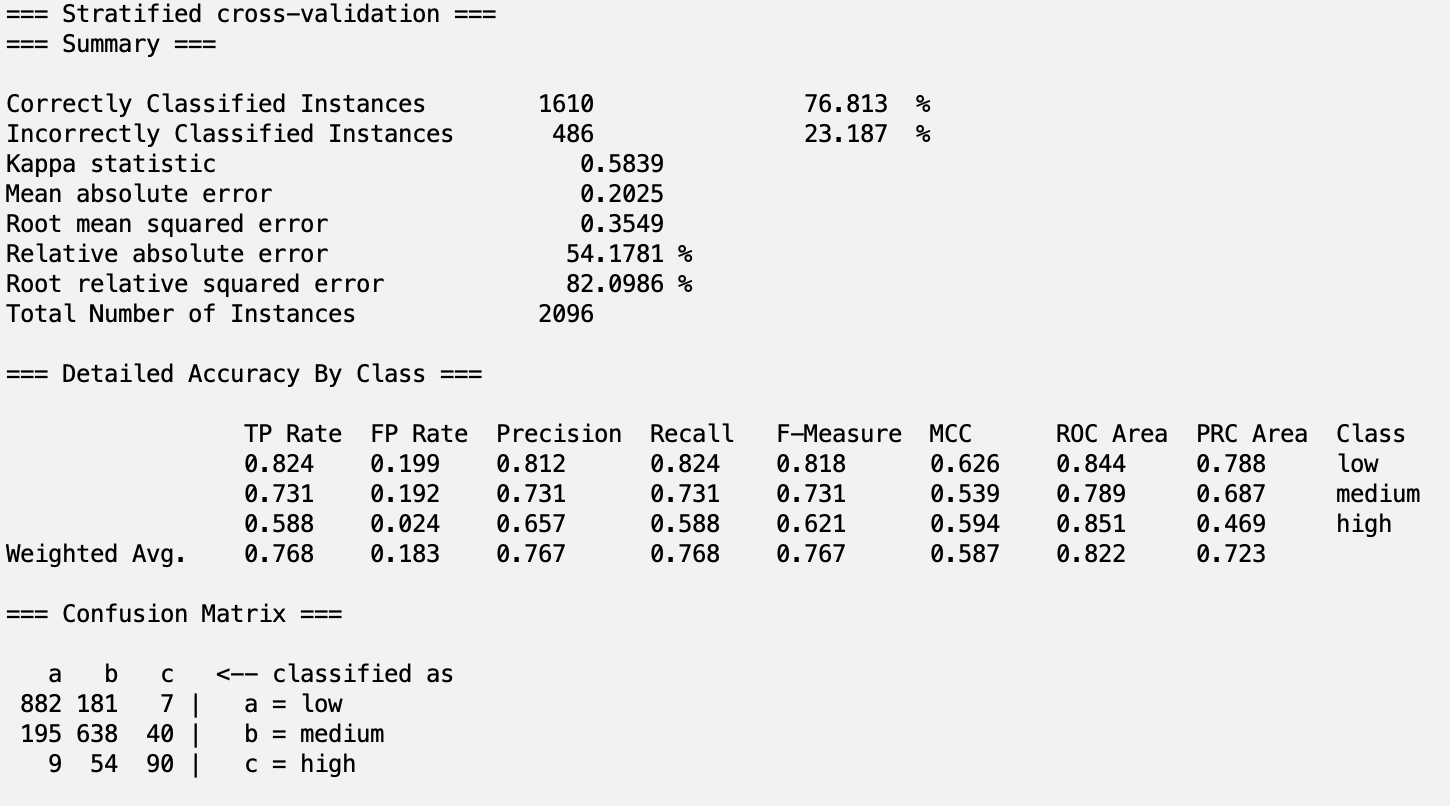
OneR with Naive Bayes



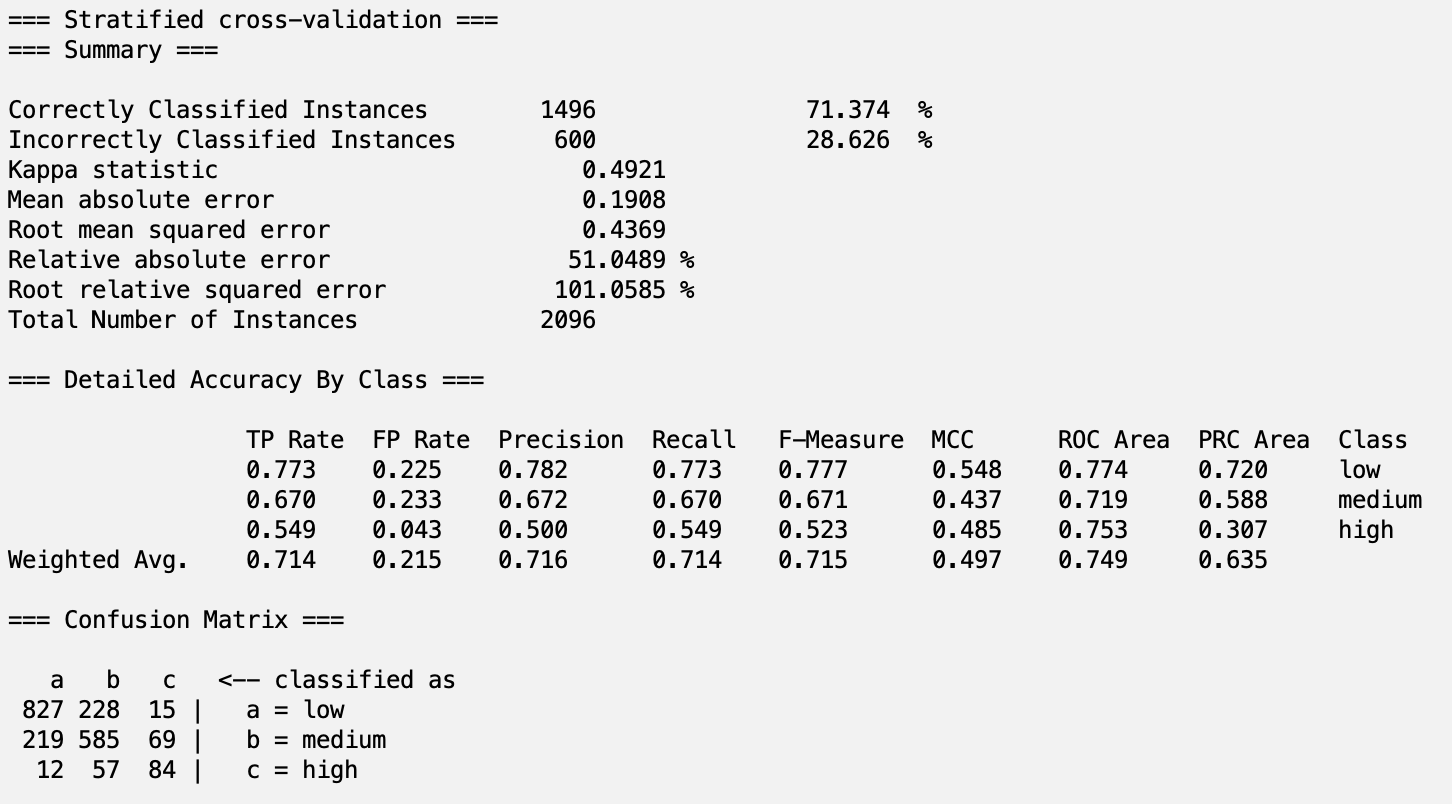
OneR with Logistic



OneR with J48

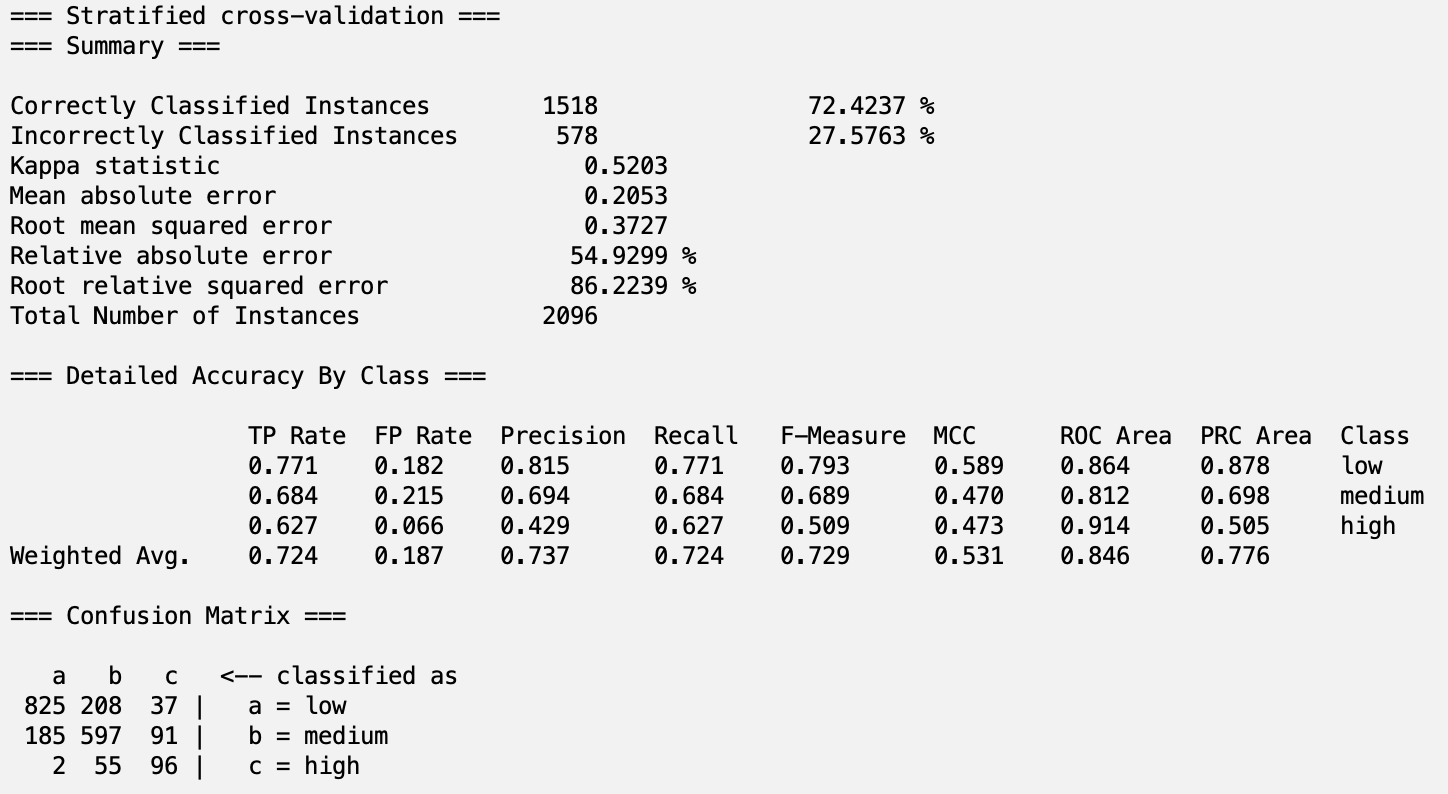


OneR with RandomTree

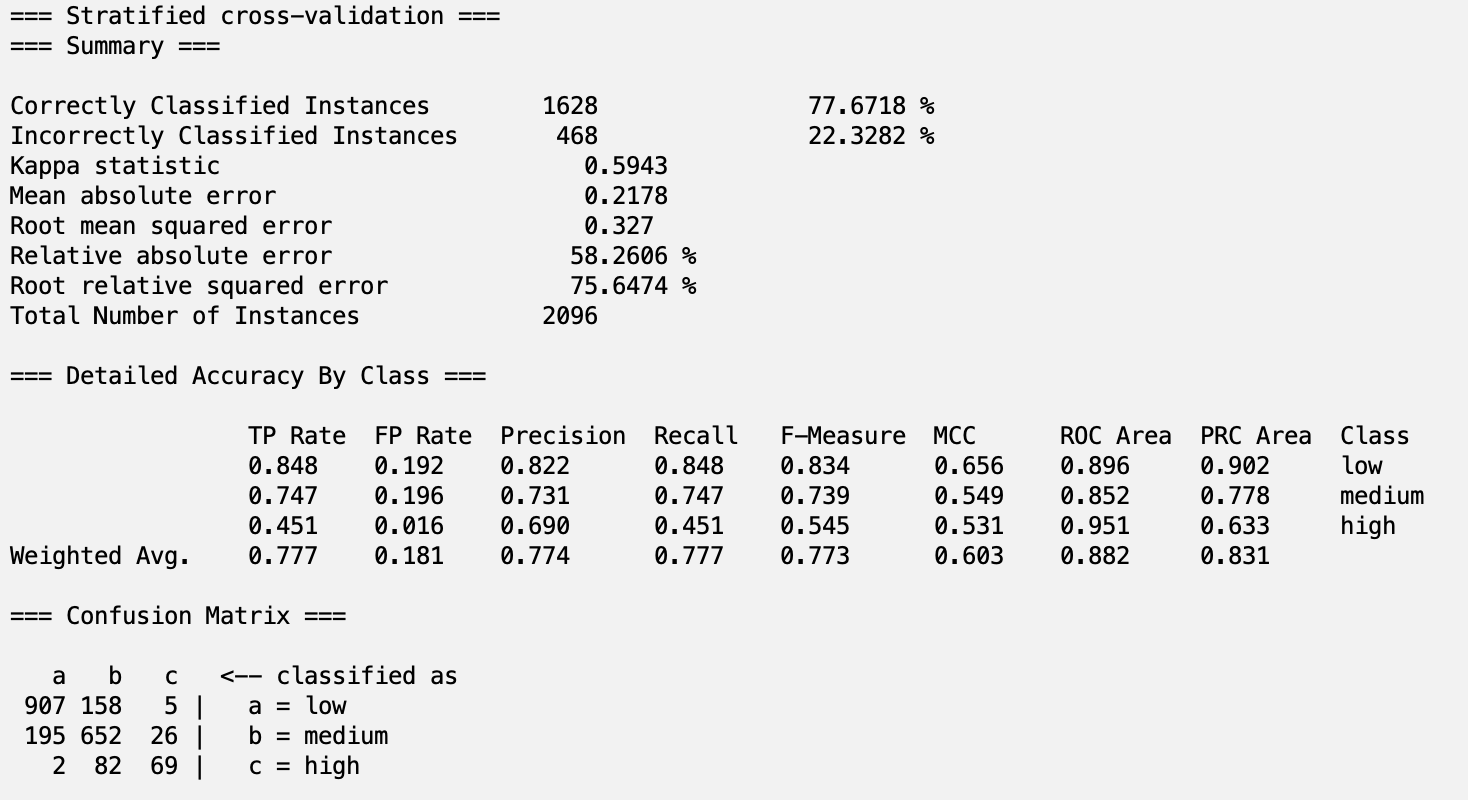


### 5.1e Subjective/Hand-picked Attribute Selection

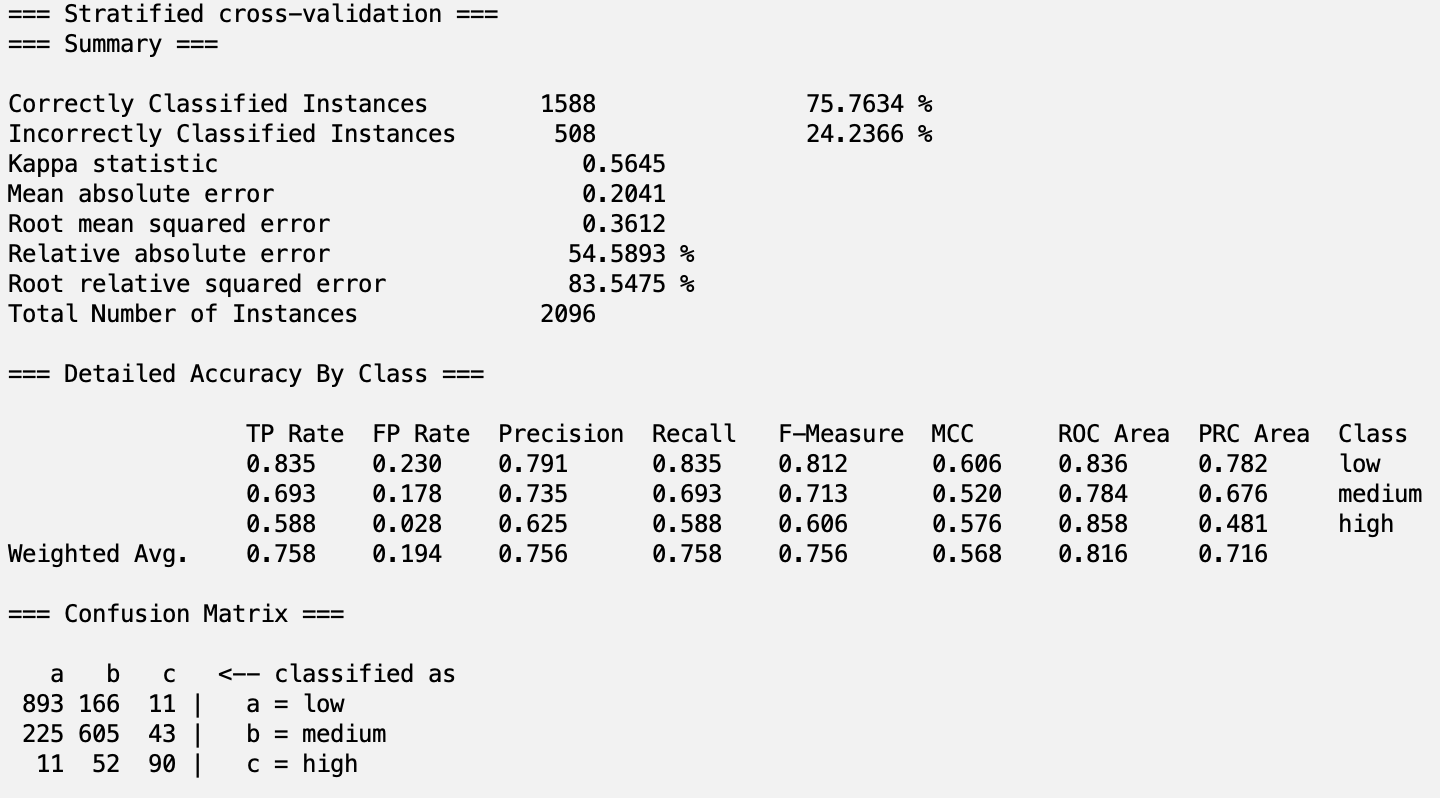
Hand-picked with Naive Bayes



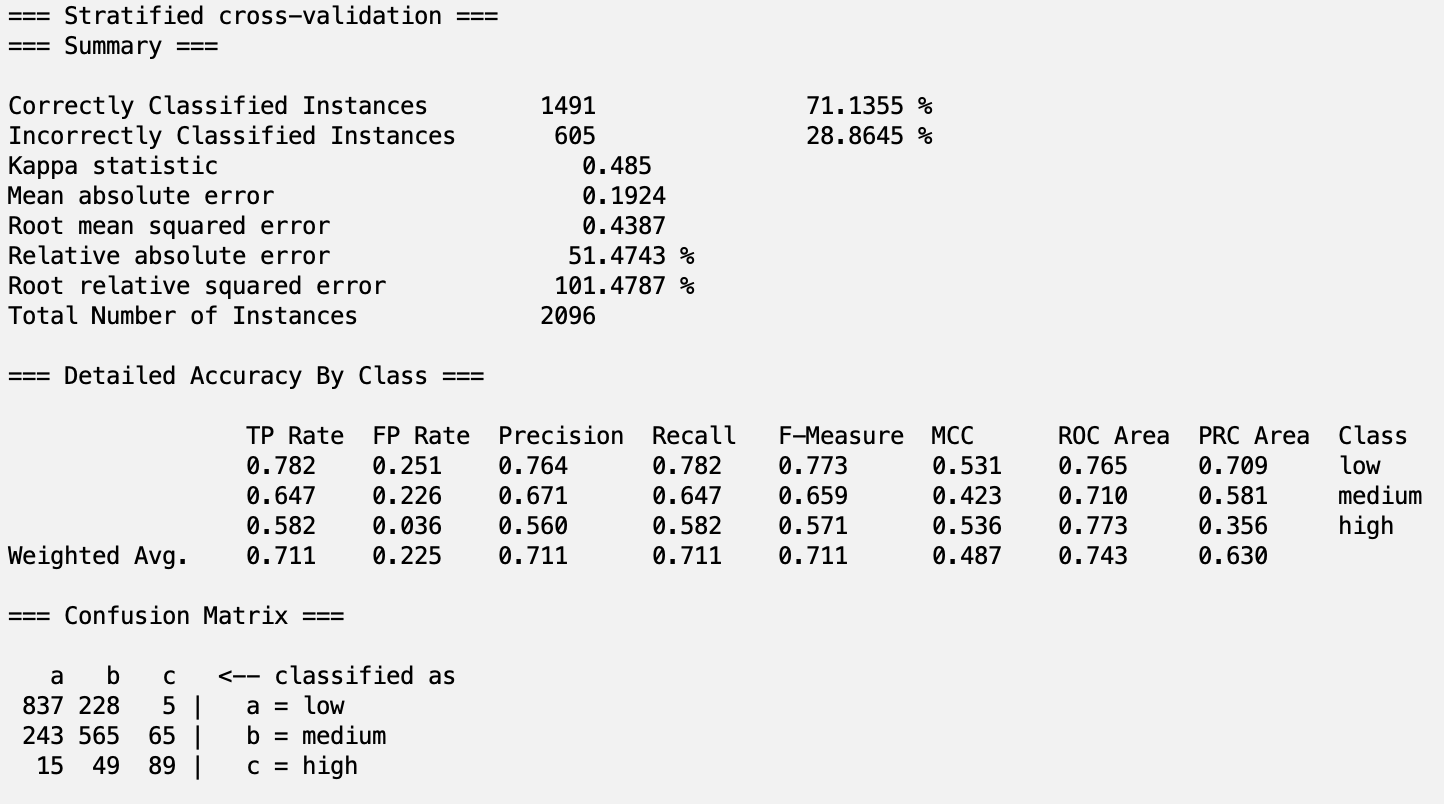
Hand-picked with Logistic



Hand-picked with J48



Hand-picked with RandomTree



## 5.2 Analysis

| Model | Accuracy (%) | TPR High | FPR High | ROC High | TPR Weighted Avg. | FPR Weighted Avg. | ROC Weighted Avg. |
| --- | --- | --- | --- | --- | --- | --- | --- |
| InfoGain-Bayes | 72.71% | 0.654 | 0.051 | 0.935 | 0.727 | 0.189 | 0.852 |
| InfoGain-Logistic | 78.10% | 0.458 | 0.013 | 0.948 | 0.781 | 0.179 | 0.879 |
| InfoGain-J48 | 76.53% | 0.641 | 0.036 | 0.862 | 0.765 | 0.178 | 0.819 |
| InfoGain-Tree | 68.94% | 0.51 | 0.046 | 0.732 | 0.689 | 0.233 | 0.728 |
| PCA-Bayes | 69.42% | 0.209 | 0.012 | 0.844 | 0.694 | 0.245 | 0.794 |
| PCA-Logistic | 73.86% | 0.307 | 0.013 | 0.916 | 0.739 | 0.214 | 0.841 |
| PCA-J48 | 67.08% | 0.275 | 0.038 | 0.701 | 0.671 | 0.257 | 0.729 |
| PCA-Tree | 61.74% | 0.288 | 0.058 | 0.615 | 0.617 | 0.29 | 0.664 |
| Learner-Bayes | 72.47% | 0.634 | 0.058 | 0.93 | 0.725 | 0.198 | 0.851 |
| Learner-Logistic | 76.77% | 0.425 | 0.014 | 0.945 | 0.768 | 0.19 | 0.869 |
| Learner-J48 | 78.01% | 0.627 | 0.022 | 0.87 | 0.78 | 0.174 | 0.841 |
| Learner-Tree | 70.66% | 0.51 | 0.046 | 0.732 | 0.707 | 0.22 | 0.743 |
| OneR-Bayes | 68.70% | 0.634 | 0.057 | 0.925 | 0.687 | 0.236 | 0.822 |
| OneR-Logistic | 77.10% | 0.438 | 0.013 | 0.946 | 0.771 | 0.188 | 0.87 |
| OneR-J48 | 76.81% | 0.588 | 0.024 | 0.851 | 0.768 | 0.183 | 0.822 |
| OneR-Tree | 71.37% | 0.549 | 0.043 | 0.753 | 0.714 | 0.215 | 0.749 |
| HandPicked-Bayes | 72.42% | 0.627 | 0.066 | 0.914 | 0.724 | 0.187 | 0.846 |
| HandPicked-Logistic | 77.67% | 0.451 | 0.016 | 0.951 | 0.777 | 0.181 | 0.882 |
| Handpicked-J48 | 75.76% | 0.588 | 0.028 | 0.858 | 0.758 | 0.194 | 0.816 |
| HandPicked-Tree | 71.14% | 0.582 | 0.036 | 0.773 | 0.711 | 0.225 | 0.743 |

Note: Highlighted top ten scores in each column of measure

After running 4 models on our 5 datasets, we found results with accuracies ranging from 61-79%. We noted six different measures of our results aside from accuracy: true positive rate (TPR), false positive rate (FPR), and ROC area of high concentrations, and the weighted averages of true positive rate, false positive rate, and ROC area. In the table above, our results with these measures are shown. We highlighted the top ten highest accuracies, highest TPR, lowest FPR, and highest ROC for both high concentrations and the weighted averages for each model. We want to find the most well-rounded model with good results for each measure, so we noted the models with the most amount of good measure scores (top ten scores). Some well-rounded models include InfoGain-Logistic, Learner-Logistic, OneR-J48, and HandPicked-Logistic which have top ten measures in 6/7 of the columns. All of these four models except OneR-J48 do not achieve top ten in TPR of high concentration, which is not ideal for our model as we are trying to predict if a lake will have a high concentration of chlorophyll-a or not. Because of this, another strong model would be InfoGain-Bayes, which has the highest TPR for high concentration (0.654) and top ten scores for accuracy, ROC area of high concentration, FPR for the weighted average, and ROC area of the weighted average. Although InfoGain-Bayes has one of the worst FPR for high concentration (0.051), this is not a major problem as it is better to check a healthy lake for signs of harmful algal bloom.

We found that the Learner-Based attribute selection algorithm with the J48 classification algorithm produced the best results. Learner-J48 was the only model with top ten measures in every column. Learner-J48 had the second highest accuracy, fifth highest TPR for high concentrations, seventh lowest FPR for high concentrations, tenth highest ROC area for high concentrations, second highest TPR for the weighted average, tenth lowest FPR for weighted average, and seventh highest ROC area for the weighted average. Although InfoGain-Bayes has a higher TPR for high concentrations, we have confidence that the Learner-J48 model will give us better predictions overall because of its higher accuracies for all seven measures.

For future reference, we can observe which attributes were most important in each attribute selection algorithm to gain an understanding of which factors play the most important role in ecosystem health.

| InfoGain | OneR | LearnerBased (J48) |
| --- | --- | --- |
| 0.43808 8 ptl  0.40161 7 ntl  0.13352 4 lon\_dd  0.12736 59 depth  0.1254 13 lst\_yrmean  0.10355 9 snow\_yrmean  0.10173 18 tmean  0.10135 19 tmean\_yrmean  0.10105 57 agkffactws  0.09842 28 n\_fert\_farm\_2007  0.0983 12 lst  0.09745 53 clayws  0.09506 47 total input  0.09349 27 n\_crop\_n\_rem\_2007  0.09 36 p\_crop\_removal\_2007  0.08875 38 p\_livestock\_demand\_2007  0.08832 14 npp  0.08732 40 p\_livestock\_production\_2007  0.08668 39 p\_livestock\_waste\_2007  0.08393 26 n\_cbnf\_2007  0.08101 32 n\_livestock\_food\_demand\_2007  0.08066 34 n\_livestock\_n\_content\_2007  0.07993 45 p\_accumulated\_ag\_inputs\_2007  0.07856 1 nani  0.07651 56 bfiws  0.07322 48 legacy  0.073 37 p\_f\_fertilizer\_2007  0.0712 33 n\_livestock.waste\_2007  0.0613 54 sandws  0.05872 23 tot\_ndep\_2007  0.05764 21 atmo\_pdep\_2007  0.0552 52 runoffws  0.05017 3 lat\_dd  0.04872 22 tot\_ndep\_2000  0.04641 15 npp\_yrmean  0.04218 35 n\_rock\_2007  0.03531 29 n\_fert\_urban\_2007  0.03214 20 atmo\_pdep\_2002  0.03157 16 precip  0.03119 46 napi  0.02726 24 tot\_sdep\_2000  0.02677 31 n\_human\_waste\_2007  0.02677 30 human\_n\_demand\_2007  0.0265 41 p\_nf\_fertilizer\_2007  0.0242 17 precip\_yrmean  0.0232 25 tot\_sdep\_2007  0.02064 43 p\_human\_nonfood\_demand\_kg\_2007  0.01975 44 p\_human\_waste\_kg\_2007  0.01972 42 p\_human\_food\_demand\_kg\_2007  0.01969 51 pcthbwet2011ws  0.01581 55 omws  0.01463 49 damdensws  0.01341 11 fire\_yrmean  0.01293 5 wsarea\_km2  0.00981 10 fire  0 50 pctwdwet2011ws  0 6 lake\_area\_ha  0 2 wetlands  0 58 p2o5ws | 71.56489 8 ptl  68.2729 7 ntl  59.58969 28 n\_fert\_farm\_2007  59.25573 4 lon\_dd  59.16031 45 p\_accumulated\_ag\_inputs\_2007  58.77863 57 agkffactws  58.6355 20 atmo\_pdep\_2002  58.58779 36 p\_crop\_removal\_2007  58.54008 56 bfiws  58.49237 33 n\_livestock.waste\_2007  58.06298 18 tmean  58.01527 39 p\_livestock\_waste\_2007  57.96756 21 atmo\_pdep\_2007  57.96756 32 n\_livestock\_food\_demand\_2007  57.87214 40 p\_livestock\_production\_2007  57.6813 47 total input  57.58588 27 n\_crop\_n\_rem\_2007  57.06107 34 n\_livestock\_n\_content\_2007  56.91794 38 p\_livestock\_demand\_2007  56.91794 19 tmean\_yrmean  56.82252 9 snow\_yrmean  56.82252 12 lst  56.7271 13 lst\_yrmean  56.29771 53 clayws  56.25 1 nani  56.10687 22 tot\_ndep\_2000  56.10687 48 legacy  56.05916 54 sandws  55.96374 59 depth  55.67748 26 n\_cbnf\_2007  54.77099 3 lat\_dd  54.3416 51 pcthbwet2011ws  54.00763 37 p\_f\_fertilizer\_2007  53.43511 46 napi  53.43511 35 n\_rock\_2007  53.05344 23 tot\_ndep\_2007  52.95802 14 npp  52.48092 58 p2o5ws  52.43321 11 fire\_yrmean  52.33779 10 fire  52.19466 49 damdensws  52.00382 52 runoffws  51.66985 6 lake\_area\_ha  51.52672 41 p\_nf\_fertilizer\_2007  51.28817 2 wetlands  51.09733 15 npp\_yrmean  51.04962 29 n\_fert\_urban\_2007  50.66794 24 tot\_sdep\_2000  50.52481 43 p\_human\_nonfood\_demand\_kg\_2007  50.38168 16 precip  50.04771 44 p\_human\_waste\_kg\_2007  49.95229 55 omws  49.85687 50 pctwdwet2011ws  49.80916 42 p\_human\_food\_demand\_kg\_2007  49.37977 5 wsarea\_km2  49.37977 31 n\_human\_waste\_2007  49.28435 30 human\_n\_demand\_2007  49.0458 25 tot\_sdep\_2007  46.18321 17 precip\_yrmean | lon\_dd  ntl  ptl  atmo\_pdep\_2002  p\_livestock\_demand\_2007  p\_livestock\_production\_2007pctwdwet2011ws |

*ptl, ntl,* and *lon\_dd* were all among the highest in each attribute selection algorithm. *ptl,* total phosphorus concentration, and *ntl*, total nitrogen concentration, intuitively would correlate with chlorophyll-a concentration as excessive algae growth can be caused by high nitrogen or phosphorus concentrations. Other attributes such as *snow\_yrmean, n\_fert\_farm\_2007, atmo\_pdep\_2002,* and *p\_livestock\_demand\_2007,* all relate to sources of phosphorus or nitrogen as well. *lon\_dd,* longitude, is slightly less intuitive, but we can infer that the surrounding climate and environment show similar trends for certain longitudes. Some climates are better for ecosystem health than others, with differences in temperatures and sources of runoff carrying nutrients, like phosphorus and nitrogen, for algal bloom.

# 

# 6 Conclusion and Reproduction

The J48 model with Learner-based attribute selection gave us the most well-rounded model out of the 20 different models of this project. Using k-fold validation, we found that our model had a 78.01% average accuracy for ten folds. Although this is not a high accuracy, we are confident that our model could be useful in the environmental health sphere. The ability to predict high chlorophyll-*a* concentrations can help to warn environmentalists if there is a problem in the ecosystem in the form of excessive algae growth, hypoxia, or harmful algal bloom. These problems can be detrimental to the organisms living within and surrounding the lake ecosystem which is why it is important to take preventative measures towards ecosystems which exhibit certain significant attributes found by our models. For future work, we suggest compiling more recent data, as our dataset mainly contains data from the years 2002 and 2007. By using more recent data, our model will be able to capture the patterns found in the current ecosystem which is heavily influenced by climate change. We would also suggest using data with a more even class distribution, as our model is heavily right skewed with the majority of instances in the low concentration levels. Since we wish to predict high or medium concentrations to alert us of any concerns in the ecosystem, we would want more data on those classes to gain a better understanding of the patterns and trends associated with those concentration levels. We would also encourage exploring the combination of different attributes as many of the attributes in our dataset are concerned with phosphorus or nitrogen. Combining these could save both space and time while keeping a good amount of the information from the data.

**Steps to Reproduce our J48 model with Learner-Based Selection:**

1. Open *original\_data.csv* in Google Sheets.
2. Under the Edit tab, select find and replace.
   1. In the Find box, type ‘ and type a space into the Replace with box.
   2. Select ‘This sheet’ or ‘All sheets’ in the Search drop down menu.
   3. Click Replace all, and repeat by replacing the following values with blank (leave the Replace box empty)
      1. “#NUM!”, “#DIV/0!”, “#VALUE!”, “#VALUE”, and “#NUM”
   4. Click Done
   5. On line 2,227 and the last column (BO), type -1 into the empty cell–this is necessary for weka to open the CSV file without error
3. Save as *replaced\_data.csv*
4. Open Weka Explorer, and open *replaced\_data.csv*.
   1. Click Edit.. to open the Viewer
   2. Scroll down to the final instance, right click, and delete the instance (should have the name DIPPER LAKE)
   3. Click OK
   4. Under Filter, click Choose, weka > filters > unsupervised > instance > RemoveWithValues
   5. Click on the horizontal bar with **RemoveWithValues**, set attributeIndex to last, matchMissingValues to True, and click OK
   6. Click Apply (the number of instances should have gone down to 2095)
5. In the Attributes box, select attributes *LAKENAME, Survey Number, SITE\_ID, Year, Month, Day,* and *SNOW* (These attributes have little correlation with our class or do not have enough values)
   1. Click Remove
6. Discretize the data
   1. Under Filter, click Choose, weka > filters > unsupervised > attribute > Discretize
   2. Set attributeIndices to 8, bins to 3, and ignoreClass to True
   3. Click OK, and click Apply
7. Save the file as *discretized\_data.arff*
8. Steps for normalization:
   1. For convenience, move the data to a .csv file.
   2. Open the .csv in *pandas*, then use the formulas for z-score normalization for those attributes that need to be z-score normalized (list in the *Preprocessing* section) or min-max normalization for all others.
      1. These formulas are as follows:
      2. min-max: data[attr] = (data[attr] - data[attr].min()) / (data[attr].max() - data[attr].min())
      3. Z-score: data[attr] = (data[attr] - data[attr].mean())/data[attr].std(ddof=0)
      4. Attr here represents the attributes, looped using a simple “for attr in data”.
   3. Save the file as *normalized\_data.csv* using data.to\_csv(“normalized\_data.csv”)
9. In Weka, open the *normalized\_data.csv* file
10. In the Select Attributes tab, under Attribute Evaluator click Choose > attributeSelection > WrapperSubsetEval
    1. Click on the horizontal bar with **WrapperSubsetEval** and click Choose next to classifier and select J48 under the trees folder
    2. Click OK
    3. Select Use full training set in Attribute Selection Mode
    4. Choose BestFirst for the Search Method
    5. Select (Nom) logchl\_A as the class
    6. Click Start
11. Drop unwanted attributes
    1. Back in the preprocess tab, select attributes LON\_DD, NTL, PTL, Atmo\_Pdep\_2002, P\_livestock\_demand\_2007, P\_livestock\_production\_2007, PctWdWet2011Ws, and logchl\_A
    2. Click Invert, then click the Remove button at the bottom; you should be left with eight attributes including the class, logchl\_A
12. Save the file as *selected\_data.arff*
13. In the classify tab, choose J48 under the trees folder
    1. For test options, choose Cross-validation with 10 folds, or click Supplied test set and upload your own test set
    2. Set logchl\_A as the class
    3. Press start
    4. Optional: Save the model to your local device by right clicking the result in the results list and selecting save model

# 7 Team Members and Tasks Performed

Jacob Dipasupil

* Dataset selection
* Attribute Selection
* Hand-picked attribute selection
* Classification
* Report writing
  + Section 1, 5, 6

Petr Kisselev

* Data Cleaning
* Attribute Selection
* Report writing
  + Section 2, 4, 7, 8
* Presentation Creation

Nikhil Alladi

* Preprocessing
* Data Cleaning
* Attribute Selection
* Report writing
  + Section 3, 4
* Presentation Creation

# 8 Appendix and Sources

Certain steps of data cleaning were performed through the use of Python and the Pandas library on Jupyter Notebooks.

*Weka* was used for all classification and attribute selection.

## 8.1 Citations

[1] Data source website: [Estimates of lake nitrogen, phosphorus, and chlorophyll-a concentrations to characterize harmful algal bloom risk across the United States](https://catalog.data.gov/dataset/estimates-of-lake-nitrogen-phosphorus-and-chlorophyll-a-concentrations-to-characterize-har)

[2] <https://weka.sourceforge.io/doc.dev/weka/attributeSelection/package-summary.html>

[3] <https://weka.sourceforge.io/doc.dev/weka/classifiers/package-summary.html>

[4]<https://weka.sourceforge.io/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html>

[5] <https://weka.sourceforge.io/doc.dev/weka/attributeSelection/OneRAttributeEval.html>

[6] <https://weka.sourceforge.io/doc.dev/weka/attributeSelection/PrincipalComponents.html>

[7] <https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/J48.html>

[8] <https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/RandomTree.html>

[9] <https://weka.sourceforge.io/doc.dev/weka/classifiers/bayes/NaiveBayes.html>

[10] <https://weka.sourceforge.io/doc.dev/weka/classifiers/functions/Logistic.html>