Dan Hoogasian & Sungmin Jung

Air Quality Index Data Mining

CS699 Spring 2018

**Table of Contents**

1. [Cover page](#_Toc292085505)

2. [Statement of data mining goal 1](#_Toc292085506)

3. Description of dataset [2](#_Toc292085508)

4. Description of data mining tools and algorithms used [2](#_Toc292085512)

5. Description of data mining procedure  [3](#_Toc292085515)

a. Data extraction  [3](#_Toc292085516)

b. Data preprocessing  [3](#_Toc292085516)

c. Running attribute selection algorithms and classifiers in Weka  [5](#_Toc292085516)

6. Data mining results and evaluation  [5](#_Toc292085516)

a. Performance measures  [5](#_Toc292085516)

b. Best model  [1](#_Toc292085516)3

7. Application  [1](#_Toc292085516)4

8. Discussion and conclusion [1](#_Toc292085516)7

9. Credits and references [18](#_Toc292085517)

## 2. Statement of data mining goal

Our “data mining” goal was to create and classify a set of class attributes for the Air Quality Index dataset that would signify if an area’s air quality was poor enough for people to want to take note of. We were only looking to “flag” areas of possible concern. People would need to do more individualized research if air quality was a serious concern of theirs. We aimed to use software and various algorithms to calculate which attributes had the most significance in determining this classification, test various classifier algorithms on these chosen attribute selections and find a top performing classifier. The desire is to be able to apply our chosen best classifier to future datasets supplied by the agency and produce classifications. This would be useful because these datasets are initially hard to extract meaning from by the non-scientific population.

## 3. Description of dataset

Our dataset is an annual summary of the Air Quality Index (AQI) values recorded at 1053 locations throughout America for the year 2016. Each location represents each and every county in America. It is made available by the EPA. The EPA uses what they call the AQS Data Mart database to store all of the data they collect relating to the AQI. There are annual and daily datasets available that go back to 1990. They are easily downloadable “csv” files with headers.

The dataset contains 19 attributes. The first two attributes are state and county, respectively. The 3rd is the year, 2016. The 4th is the total number of days in 2016 that the AQI was measured. The 5th through and the 10th attribute represents the number of days that tuple had that fell under what the EPA titled “Good Days”, “Moderate Days”, “Unhealthy for Sensitive Groups Days”, “Unhealthy Days”, “Very Unhealthy Days” and “ Hazardous Days”. We couldn’t find anywhere on the website where it described the parameters of these classifications.

The 11th attribute is max AQI, representing the highest AQI recorded for that tuple during 2016. Again we couldn’t find how this was calculated. The 12th attribute is 90th percentile AQI, representing the value of which 90 percent of the years AQI values were below. The 13th attribute is the median annual AQI for that location.

The 14th attribute represents the numbers of days that location had a CO (carbon monoxide) level meeting their minimum threshold. Similarly, the 15th through 17th attribute represents the numbers of days that location had a NO2 (nitrous dioxide), ozone or SO2 (sulfur dioxide) level meeting their minimum threshold.

The 18th and 19th attributes represent the number of days that tuple had that met the threshold for the amount of a wide variety of toxic particulates of a size falling into the PM2.5 and PM10 classifications. On a webpage key, they list all the particulates, which equals seven for PM2.5 and eight for PM10.

## 4. Description of data mining tools and algorithms used

We used a total of four attribute selection algorithms (plus one set selected by ourselves) and six classification algorithms, all implemented through the Weka data mining software.

The first attribute selection algorithm we used was the correlation attribute evaluator. From what Weka describes, this algorithm measures the correlation coefficient (Pearson’s product moment correlation coefficient) between a numeric attribute and the class attribute, to calculate the weight of that attribute [1].

The second attribute selection algorithm was the gain ratio attribute evaluator. This algorithm takes the information gain of an attribute and weighs it against the number of values for that attribute to overcome the bias the information gain algorithm has for attributes with a large number of values [1].

The third attribute algorithm was the information gain attribute evaluator. This looks at the information gain of each attribute related to its class. It does this by first using a logarithmic function to quantify the information needed to classify a given tuple. Then it similarly quantifies the information needed to classify a given tuple if it “splits” the tuple on that attribute, ignoring the beforehand attributes. It then subtracts the individual info value from the total info value to get the information gained by that attribute [1].

The last attribute algorithm was the ReleifF attribute evaluator. This algorithm repeatedly samples tuples and looks at the k-nearest instance to quantify the worth of an attribute [1].

For our personally selected attributes, we simply eyeballed all the calculated attribute weights and made a selection based on which ones had the highest relative weights. Our guess performed very well so we decided to leave it as is.

The first classification algorithm we used was the Naïve Bayes classifier. This uses a simplified formula utilizing Bayes Theorem and probabilistic prediction to come up with classifications.

The second classifier was the random forest tree ensemble. This is an ensemble method classifier that uses a wide variety of decision tree algorithms. Each individual algorithm’s classification is combined to create a more accurate ultimate classification.

The third classifier was the J48 tree algorithm. This J48 decision tree is an extension of the ID3 tree algorithm, which is based on the C4.5 algorithm [2]. “The additional features of J48 are accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc.” [2].

The fourth classifier was the OneR rule classifier. This is an easy to use the algorithm that simply makes classifications based on one attribute. It takes the error rates for each attribute and then uses the one with the lowest rate to make classifications.

Our fifth classifier was the IBK algorithm. This is a so-called “lazy” classifier that utilizes a k-nearest neighbor type algorithm. This means that it looks at the classifications of similar tuples to come up with a classification estimate for the specific tuple.

Our sixth and final classifier was the logistic function classifier. This uses a polynomial logistic regression model with the addition of a ridge estimator, which adds a greater weight to large coefficients. It does this to minimalize over-fitting [3].

## 5. Description of data mining procedure

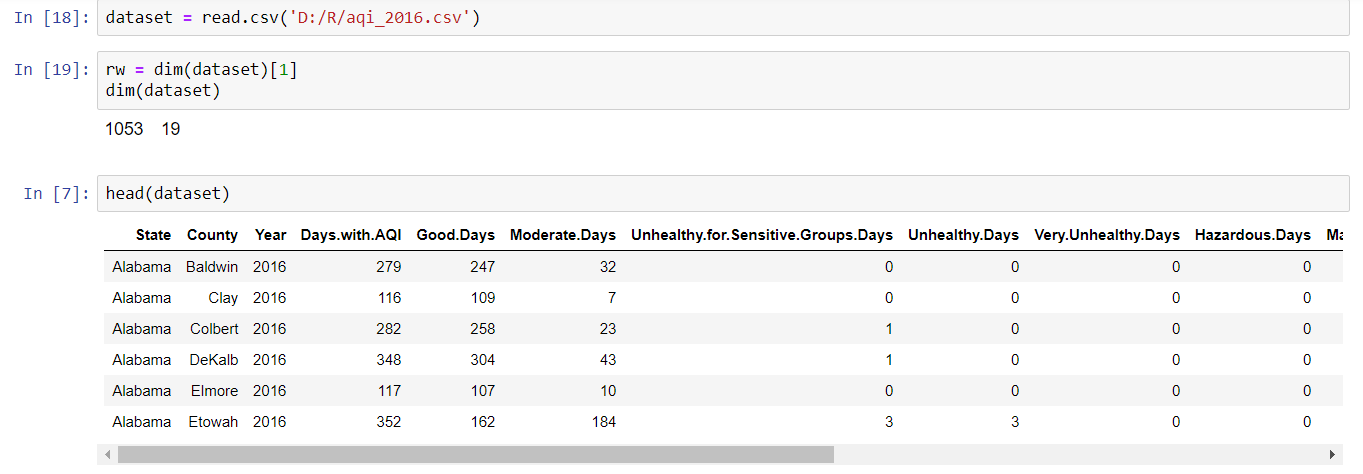
## a. Data extraction

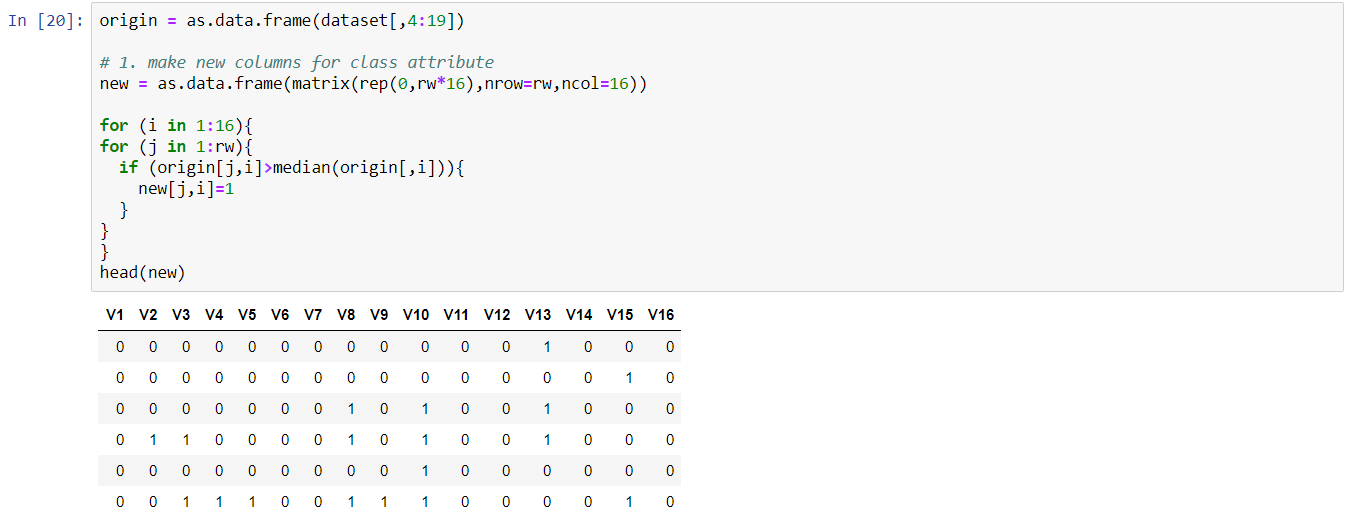
We downloaded the data from this EPA webpage: <https://aqs.epa.gov/aqsweb/airdata/download_files.html>.

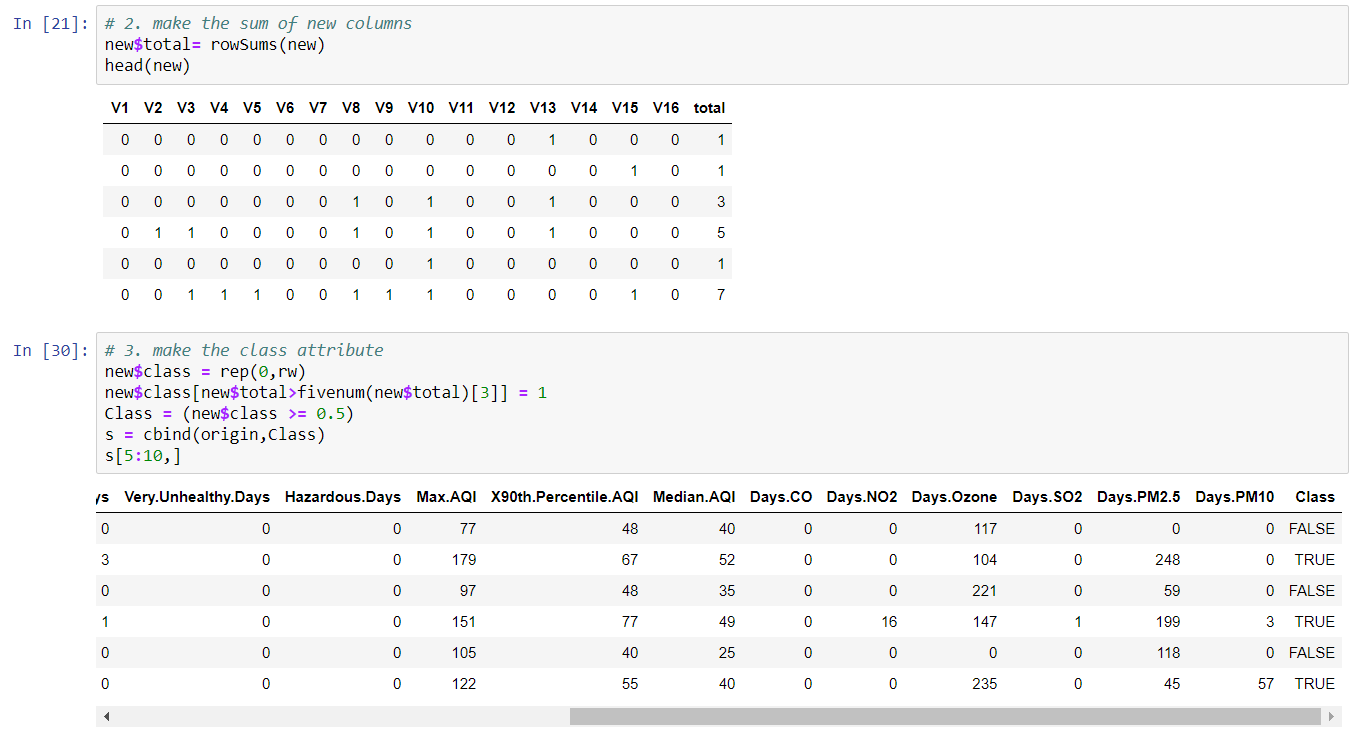
**b. Data preprocessing**

The 2017 Air Quality Index file has a 1053\*19 matrix. Since there is no class attribute in the data, we created the class attribute. First, we eliminated the first three columns related to location and year, and used the numeric variable columns four through nineteen to create a new attribute column. This new column held numeric variables that represented the overall level of air quality. We then created a new column of our class attributes. The variables were of an asymmetrical binary type that held values of “True” and “False”. We split the classification based on if the value of the new numeric column was more or less than its median. “True” indicated air is polluted and “False” indicated air quality is good.

R code for data preprocessing:







**c. Running attribute selection algorithms and classifiers in Weka**

With the new class attribute column added, we saved the data as a “csv” file through Microsoft Excel and then read it into Weka. The first thing we did in Weka was to remove the header and save this new file. Then we ran four attribute evaluators and made a selection of the top four or five attributes for each evaluator, based on their ranking. Then we chose five attributes for our personally selected attribute set. For all the evaluators we had the attribute selection mode set to the default of using the full training set. We saved all of these selections as “arff” files.

Next we ran six different classifiers on each of our attribute combinations and saved the screenshots of the relevant performance measures. For all the classifiers we had the test options set to cross-validation of ten folds.

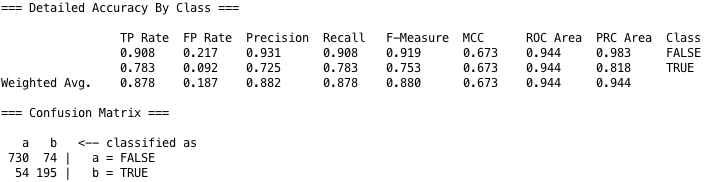
## 6. Data mining results and evaluation

## a. Performance measures

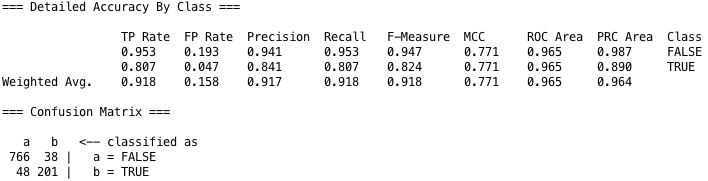
|  |  |
| --- | --- |
| **Attribute Evaluator** | **Selected attributes** |
| CorrelationAttributeEval | 0.6371 3 Moderate.Days  0.5108 9 X90th.Percentile.AQI  0.434 10 Median.AQI  0.4071 4 Unhealthy.for.Sensitive.Groups.Days  0.3291 1 Days.with.AQI |
| GainRationAttributeEval | 0.1955 3 Moderate.Days  0.1944 6 Very.Unhealthy.Days  0.1882 9 X90th.Percentile.AQI  0.1878 5 Unhealthy.Days |
| InfoGainAttributeEval | 0.3746 3 Moderate.Days  0.306 9 X90th.Percentile.AQI  0.2879 4 Unhealthy.for.Sensitive.Groups.Days  0.2642 8 Max.AQI |
| ReliefFAttributeEval | 0.129451 1 Days.with.AQI  0.120633 2 Good.Days  0.092016 15 Days.PM2.5  0.085883 13 Days.Ozone  0.067579 3 Moderate.Days |
| Personally selected attributes | 1 Days.with.AQI  2 Good.Days  3 Moderate.Days  4 Unhealthy.for.Sensitive.Groups.Days  13 Days.Ozone |

**i. Correlation Attribute Evaluator**

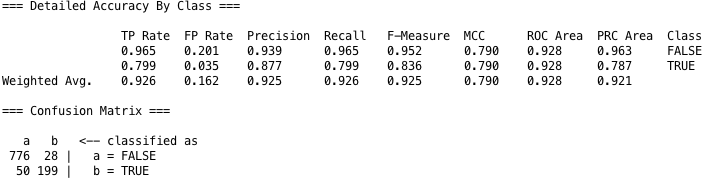
Naive Bayes Classifier:



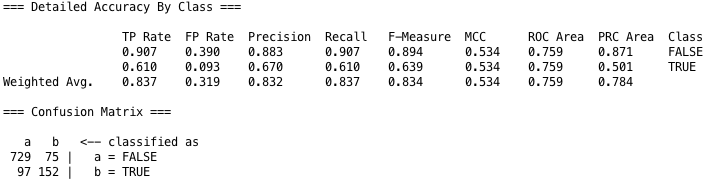
Random Forest Classifier:



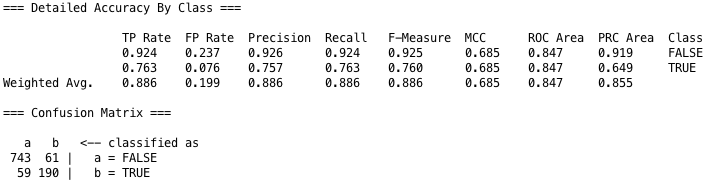
J48:



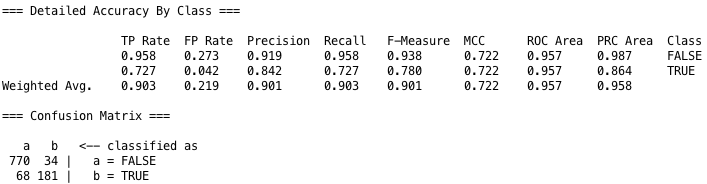
OneR (One Rule):



IBK (k-Nearest Neighbor):

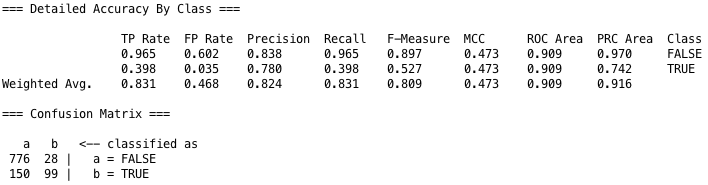


Logistic:

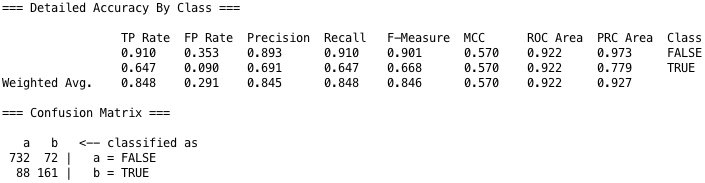


**ii. Gain Ratio Attribute Evaluator**

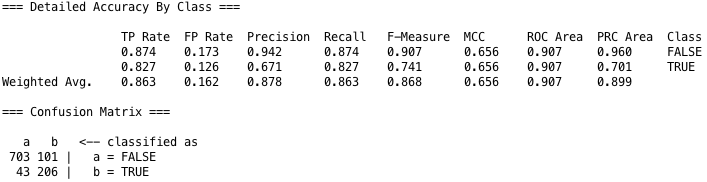
Naive Bayes Classifier:



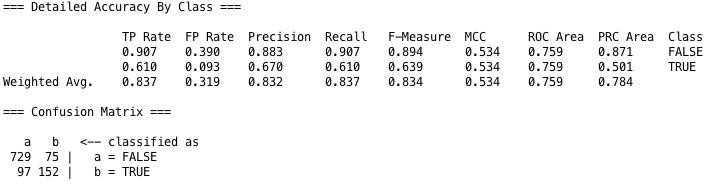
Random Forest Classifier:



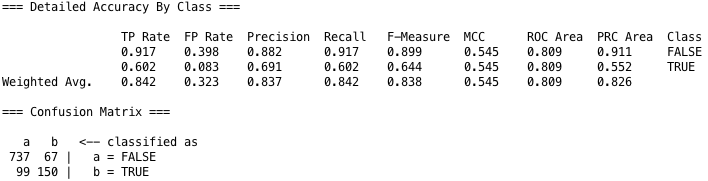
J48:



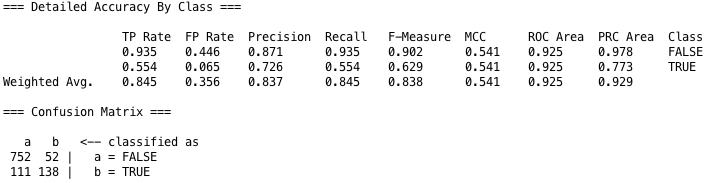
OneR (One Rule):



IBK (k-Nearest Neighbor):

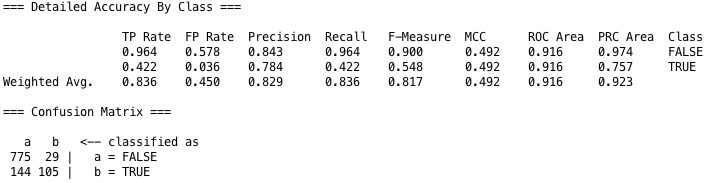


Logistic:

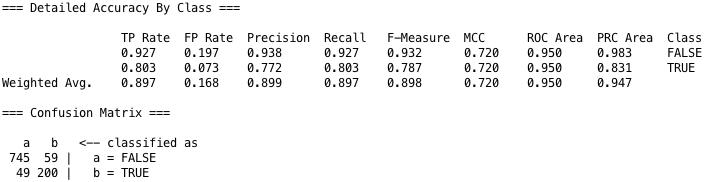


**iii. Info Gain Attribute Evaluator**

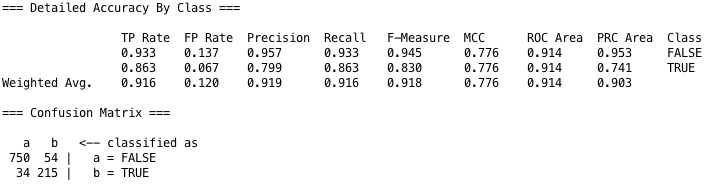
Naive Bayes Classifier:



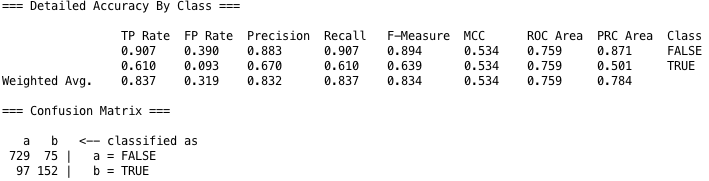
Random Forest Classifier:



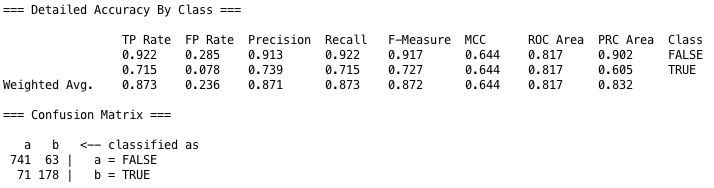
J48:



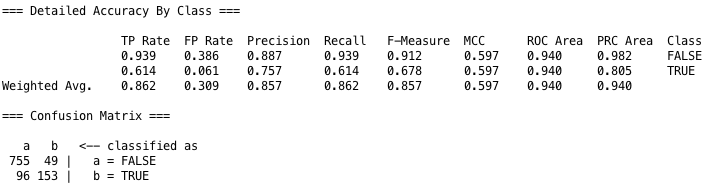
OneR (One Rule):



IBK (k-Nearest Neighbor):

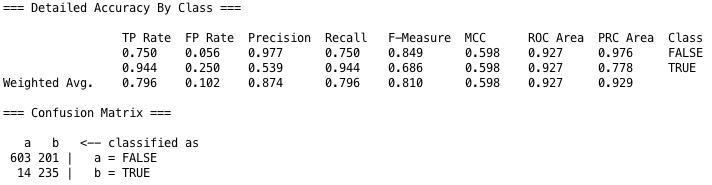


Logistic:

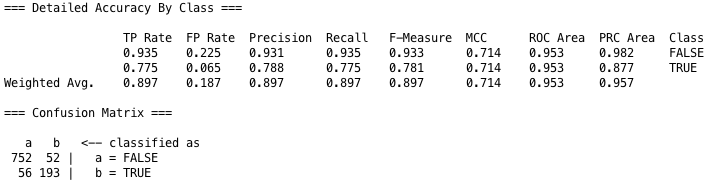


**iv. ReliefF Attribute Evaluator**

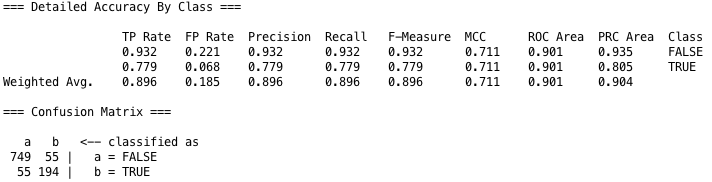
Naive Bayes Classifier:



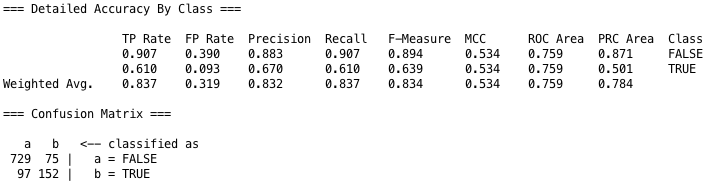
Random Forest Classifier:



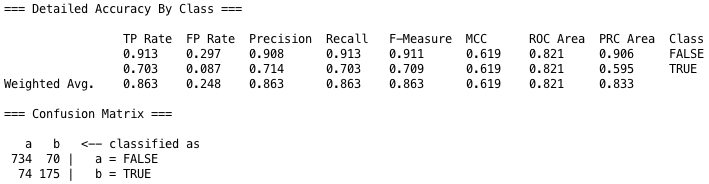
J48:



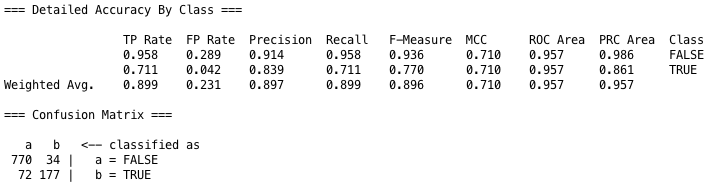
OneR (One Rule):



IBK (k-Nearest Neighbor):

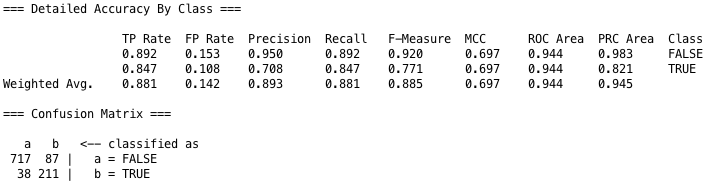


Logistic:

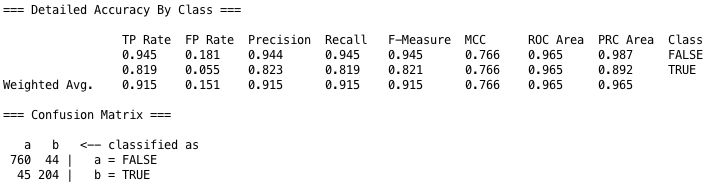


**v. Personally selected attributes**

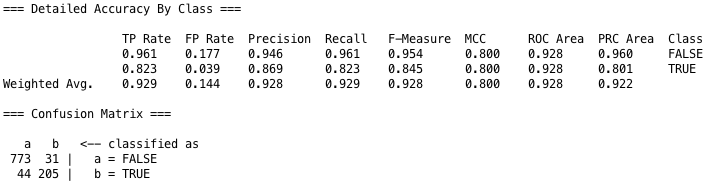
Naive Bayes Classifier:



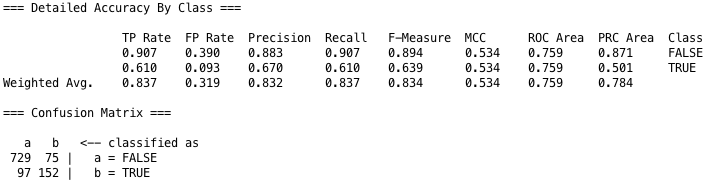
Random Forest Classifier:



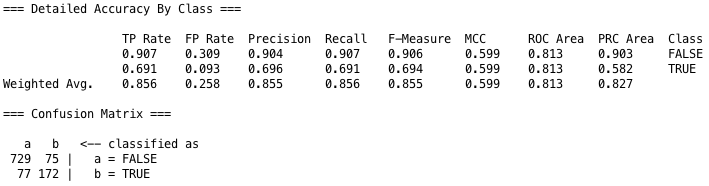
J48:



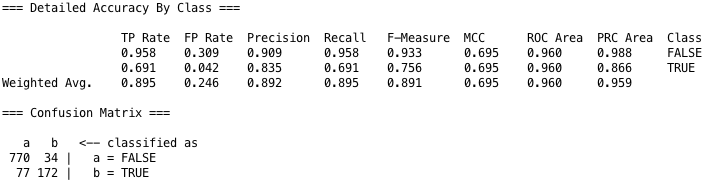
OneR (One Rule):



IBK (k-Nearest Neighbor):



Logistic:

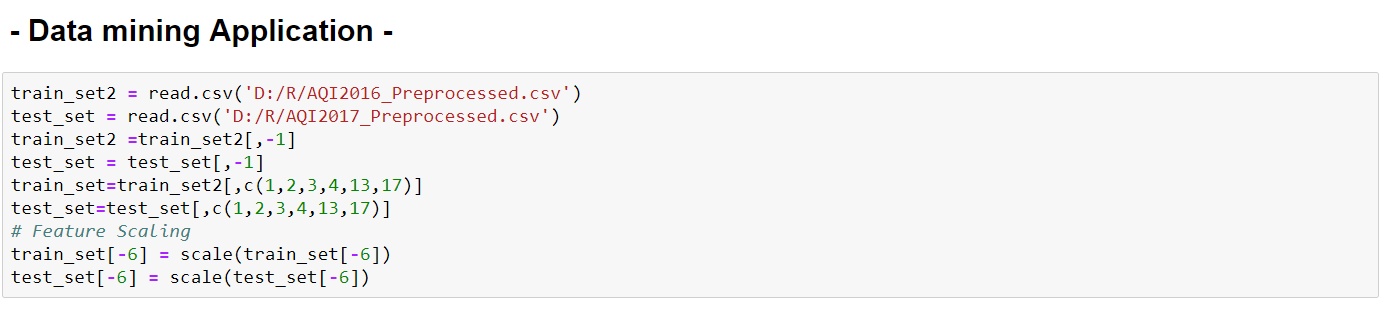


1. **Best algorithm**

Based on all of the classification performance measures, we chose the Random Forest and the J48 as the top two. This was based on the Random Forest scoring the highest AUC (area under the curve) of .965 on both our correlation attribute evaluator and personally selected sets of attributes. Looking at just the accuracy, running the J48 on our personally selected attributes scored the best at .9288. This was only slightly better than the accuracy of the Random Forest, whereas the AUC of the Random Forest was significantly better than the J48’s. For this reason we chose Random Forest to be the overall best model.

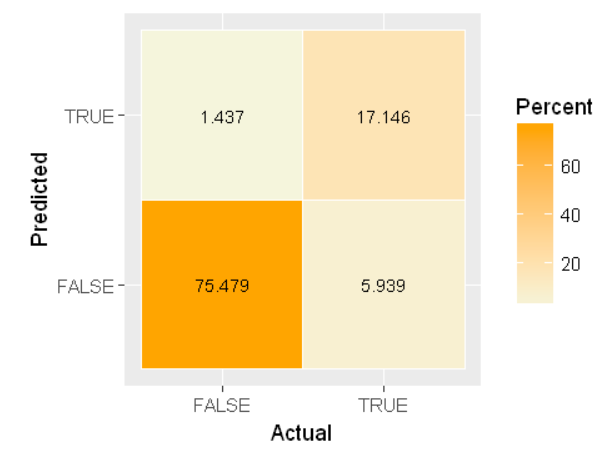
**7. Application**

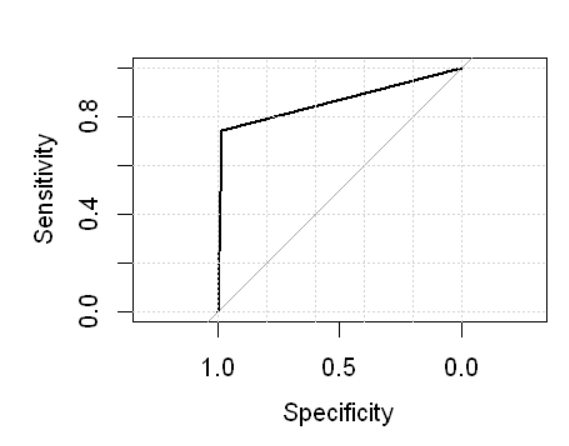
We used the Random Forest and the J48 models built from the 2016 dataset, on the 2017 dataset. The J48 performed better than the Random Forest when looking at both the accuracy and AUC.





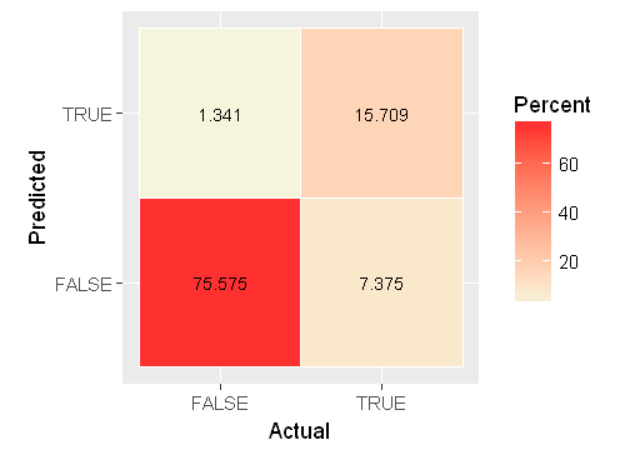


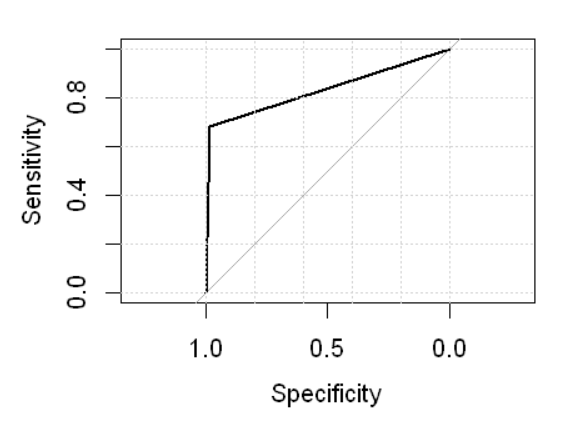


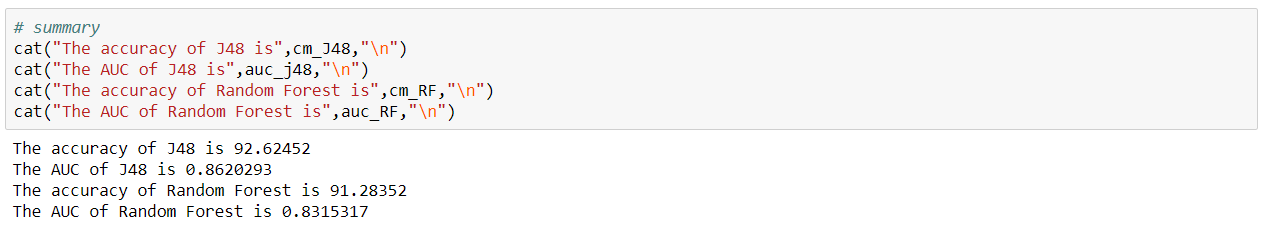












## 8. Discussion and conclusion

Through what we accomplished we conclude that with the techniques and tools we learned to use in CS699, we are able to classify a large dataset into classifications that people can easily look at to make well-informed decisions about a practical matter. If air quality is a serious concern of someone’s, more personalized research is needed. If for example someone had severe asthma or COPD, the person would need to do more research into which pollutants were most harmful to them and look at what the individual levels were of those pollutants in the area (information included in the original dataset). This is obviously beyond the scope of this course and project. Our results simply flag areas of concern.

From this project we learned how to complete the whole procedure of taking a dataset, creating class attributes, selecting sets of attributes that had the most influence on a tuple’s classification, running data mining classification algorithms, comparing classification performance measures and choosing a top classifier to use on new datasets. We went a step farther and actually applied our top model to a dataset from a different year. It was great to see our top classifier in use.

This might not be true for everyone, but we learned that once we had a solid understanding of the concepts behind data mining like attribute selection methodology, classification techniques, and the related performance measures, implementing them through the Weka software program to produce easily interpretable results was not too complicated.

With what we learned in the course and from doing this project, we can easily see how to use these tools in real-world applications to produce meaningful results.

## 9. Credits and references

## a. Credits

Dan Hoogasian:

3. Description of dataset

4. Description of data mining tools and algorithms used

5. Description of data mining procedure (part c)

6. Data mining results and evaluation

8. Discussion and conclusion

9. Credits and references

Sungmin Jung:

1. Cover page and table of contents

2. Statement of data mining goal

5. Description of data mining procedure (part a & b)

7. Application

## b. References

* [1] SourceForge. Slashdot Media, 2018, <http://weka.sourceforge.net/doc.dev> Accessed 25 March, 2018.
* [2] Gaganjot Kaur and Amit Chhabra. Article: Improved J48 Classification Algorithm for the Prediction of Diabetes. *International Journal of Computer Applications* 98(22):13-17, July 2014, <https://pdfs.semanticscholar.org/2456/a979fbe8eea47b90d625c1a064162be5382e.pdf>. Accessed 25 March, 2018.
* [3] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.
* Class slides:

Han, J., Kamber, M., Pei, J., “Data mining: concepts and

techniques,” 3rd Ed., Morgan Kaufmann, 2012.