Assignment 3: Wine Data

CIS 435

Section 56

Summer Quarter

School of Continuing Studies

Northwestern University

Daniel Prusinski

Business Intelligence Data Analyst

Target Corporation

Minneapolis, MN

In Compliance with Master of Science Predictive Analytics

Preliminary Data Analysis:

Before delving into the meat and potatoes of an analysis, it is important to conduct a preliminary exploratory data analysis. This gives an initial foundation that will prove to be the reference point moving forward into more complex analysis. The meta data states there are 14 attributes with 178 instances or rows. In addition, this data is pulled from 3 different cultivars. Cultivar A has the largest sample followed by A, and lastly C. An initial takeaway is that B will have more weight in the analysis simply based on the fact that it represents 40%, A is perfectly represented, and C is underrepresented.

Class

Attribute A B C

(0.33) (0.4) (0.27)

=====================================================

Alcohol

mean 13.7434 12.2782 13.1537

std. dev. 0.4587 0.5351 0.5252 🡨 Given the Avg, this distribution of values follows a rather normal distribution, which leads me to believe it is a veracious attribute. Also, between the different cultivars the deviation is not far, which shows balance as a variable.

weight sum 59 71 48 🡨 This sum validates what is shown in in the green circle above about B having more weight in the analysis.

precision 0.0304 0.0304 0.0304

Malic\_Acid

mean 2.0115 1.9329 3.3334

std. dev. 0.6824 1.0078 1.0749 🡨 The mean compared to the STDV (Standard Deviation) is somewhat similar, which points to a rather tumultuous distribution. In conclusion, the distribution might require a log transformation.

weight sum 59 71 48

precision 0.0383 0.0383 0.0383

Ash

mean 2.4555 2.2451 2.4354

std. dev. 0.2253 0.3139 0.1817

weight sum 59 71 48

precision 0.024 0.024 0.024

Ash\_Alcalinity

mean 17.0506 20.2594 21.4208

std. dev. 2.5279 3.3209 2.2327

weight sum 59 71 48

precision 0.3129 0.3129 0.3129

Magnesium

mean 106.3338 94.5915 99.2981

std. dev. 10.4831 16.6495 10.8441 🡨 This STDV is quite high for such a small mean. I am concerned about this distribution

weight sum 59 71 48

precision 1.7692 1.7692 1.7692

Total\_Phenols

mean 2.8396 2.2618 1.681

std. dev. 0.3357 0.5412 0.3553 🡨 This STDV is quite high for such a small mean. I am concerned about this distribution

weight sum 59 71 48

precision 0.0302 0.0302 0.0302

Flavanoids

mean 2.983 2.0793 0.7802

std. dev. 0.3944 0.7013 0.2896

weight sum 59 71 48

precision 0.0362 0.0362 0.0362

Nonflavanoid\_Phenols

mean 0.2908 0.3646 0.4478

std. dev. 0.0701 0.1229 0.123 🡨 The mean compared to the STDV (Standard Deviation) is somewhat similar, which points to a rather tumultuous distribution. In conclusion, the distribution might require a log transformation.

weight sum 59 71 48

precision 0.0139 0.0139 0.0139

Proanthocyanins

mean 1.8982 1.631 1.1518

std. dev. 0.4095 0.5992 0.4046

weight sum 59 71 48

precision 0.0317 0.0317 0.0317

Color\_Intensity

mean 5.5241 3.0796 7.3996

std. dev. 1.2265 0.9159 2.2849

weight sum 59 71 48

precision 0.0895 0.0895 0.0895

Hue

mean 1.0611 1.0559 0.6836

std. dev. 0.1151 0.2013 0.1129🡨 The mean compared to the STDV (Standard Deviation) is somewhat similar, which points to a rather tumultuous distribution. In conclusion, the distribution might require a log transformation.

weight sum 59 71 48

precision 0.016 0.016 0.016

OD280\_OD315

mean 3.1579 2.7843 1.6842

std. dev. 0.3543 0.4923 0.2688

weight sum 59 71 48

precision 0.0226 0.0226 0.0226

Proline

mean 1115.8573 519.8261 629.683

std. dev. 220.0034 154.7719 113.0791

weight sum 59 71 48

precision 11.6833 11.6833 11.6833

Many of the above variables do not have normal distributions, which raises data quality concerns. Additionally, Proline is significantly larger than any other variable, which has the potential to skew results.

Wine data in Weka using NaiveBayesSimple:

Scheme:weka.classifiers.bayes.NaiveBayes

Relation: Wine

Instances: 178

Attributes: 14

Test mode:10-fold cross-validation

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 172 96.6292 %

Incorrectly Classified Instances 6 3.3708 %

Kappa statistic 0.9489

Mean absolute error 0.0217

Root mean squared error 0.1294

Relative absolute error 4.9371 %

Root relative squared error 27.6176 %

Total Number of Instances 178

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class

0.949 0 1 0.949 0.974 0.998 A

0.958 0.028 0.958 0.958 0.958 0.997 B

1 0.023 0.941 1 0.97 1 C

Weighted Avg. 0.966 0.017 0.967 0.966 0.966 0.998

=== Confusion Matrix ===

a b c <-- classified as

56 3 0 | a = A

0 68 3 | b = B

1. 0 48 | c = C

Initial analysis of NaiveBayesSimple:

NaiveBayesSimple correctly classified 172 of the instances and from a practical perspective has great output. This approach has a better classification than J48.

Wine data in Weka using IBK (kNN algorithm):

Scheme:weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Relation: Wine

Instances: 178

Attributes: 14

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

IB1 instance-based classifier

using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 169 94.9438 %

Incorrectly Classified Instances 9 5.0562 %

Kappa statistic 0.9238

Mean absolute error 0.0413

Root mean squared error 0.1821

Relative absolute error 9.3973 %

Root relative squared error 38.8682 %

Total Number of Instances 178

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class

1 0.042 0.922 1 0.959 0.983 A

0.873 0 1 0.873 0.932 0.941 B

1 0.031 0.923 1 0.96 0.983 C

Weighted Avg. 0.949 0.022 0.953 0.949 0.949 0.966

=== Confusion Matrix ===

a b c <-- classified as

59 0 0 | a = A

5 62 4 | b = B

1. 0 48 | c = C

Initial analysis of IBK (kNN algorithm):

IBK does not classify as well as Naïve Bayes, and the Root Mean Squared error is larger. It is better than J48 and the root mean square error is not as high.

Wine data in Weka using jRP:

=== Run information ===

Scheme:weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -S 1

Relation: Wine

Instances: 178

Attributes: 14

Alcohol

Malic\_Acid

Ash

Ash\_Alcalinity

Magnesium

Total\_Phenols

Flavanoids

Nonflavanoid\_Phenols

Proanthocyanins

Color\_Intensity

Hue

OD280\_OD315

Proline

Type

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

JRIP rules:

===========

(Flavanoids <= 1.39) and (Color\_Intensity >= 4) => Type=C (46.0/0.0)

(OD280\_OD315 <= 1.3) => Type=C (2.0/0.0)

(Proline >= 760) => Type=A (61.0/4.0)

=> Type=B (69.0/2.0)

Number of Rules : 4

Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 167 93.8202 %

Incorrectly Classified Instances 11 6.1798 %

Kappa statistic 0.9059

Mean absolute error 0.048

Root mean squared error 0.1991

Relative absolute error 10.9333 %

Root relative squared error 42.4986 %

Total Number of Instances 178

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class

0.932 0.034 0.932 0.932 0.932 0.973 A

0.944 0.056 0.918 0.944 0.931 0.97 B

0.938 0.008 0.978 0.938 0.957 0.97 C

Weighted Avg. 0.938 0.036 0.939 0.938 0.938 0.971

=== Confusion Matrix ===

a b c <-- classified as

55 4 0 | a = A

3 67 1 | b = B

1 2 45 | c = C

Initial analysis of JRIP):

JRIP and IBK have similar output, but JRIP is the least precise classifier.

Wine data in Weka using J48 classifier:

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: Wine

Instances: 178

Attributes: 14

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

------------------

Flavanoids <= 1.57

| Color\_Intensity <= 3.8: B (13.0)

| Color\_Intensity > 3.8: C (49.0/1.0)

Flavanoids > 1.57

| Proline <= 720: B (54.0/1.0)

| Proline > 720

| | Color\_Intensity <= 3.4: B (4.0)

| | Color\_Intensity > 3.4: A (58.0)

Number of Leaves : 5

Size of the tree : 9

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 167 93.8202 %

Incorrectly Classified Instances 11 6.1798 %

Kappa statistic 0.9058

Mean absolute error 0.0486

Root mean squared error 0.2019

Relative absolute error 11.0723 %

Root relative squared error 43.0865 %

Total Number of Instances 178

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class

0.983 0.034 0.935 0.983 0.959 0.977 A

0.944 0.056 0.918 0.944 0.931 0.937 B

0.875 0.008 0.977 0.875 0.923 0.946 C

Weighted Avg. 0.938 0.036 0.94 0.938 0.938 0.953

=== Confusion Matrix ===

a b c <-- classified as

58 1 0 | a = A

3 67 1 | b = B

1 5 42 | c = C

Initial analysis of J48:

J48 correctly classified 167 of the instances and from a practical perspective has great output. This tree is rather small and only has 5 leaves. In accordance with the principle of Parsimony, I feel capable to explain these results to clients. The root mean squared error is often compared as an inverses calculation to the correlation coefficient. This data is from wine, and I am not as concerned with False Negatives as compared with medical diagnostic data. As my analysis furthers, I am curious to analyze the relationship between Proline and color intensity given that they led to such an apt decision tree.

1.A Rank the top three methods according to the rate of correctly classified instances:

1. NaiveBayesSimple: C.C. 172 96.6292 %
2. IBK (kNN algorithm): 169 94.9438 %
3. J48 and JRIP have the same % value: 93.8202 %\*
   1. In my opinion, JRIP is better because it has a small root mean square error.

1.B Compare the difference between jRIP and j48. Are the two methods concluding the same classification rules? If not, what are the differences?

The output for the two different methods can be seen above.

The two methods are not concluding the same classification rules.

jRIP has the following rules 🡪 (Flavanoids <= 1.39) and (Color\_Intensity >= 4) => Type=C (46.0/0.0)

(OD280\_OD315 <= 1.3) => Type=C (2.0/0.0)

(Proline >= 760) => Type=A (61.0/4.0)

=> Type=B (69.0/2.0)

J48 has the following rules🡪 Flavanoids <= 1.57

| Color\_Intensity <= 3.8: B (13.0)

| Color\_Intensity > 3.8: C (49.0/1.0)

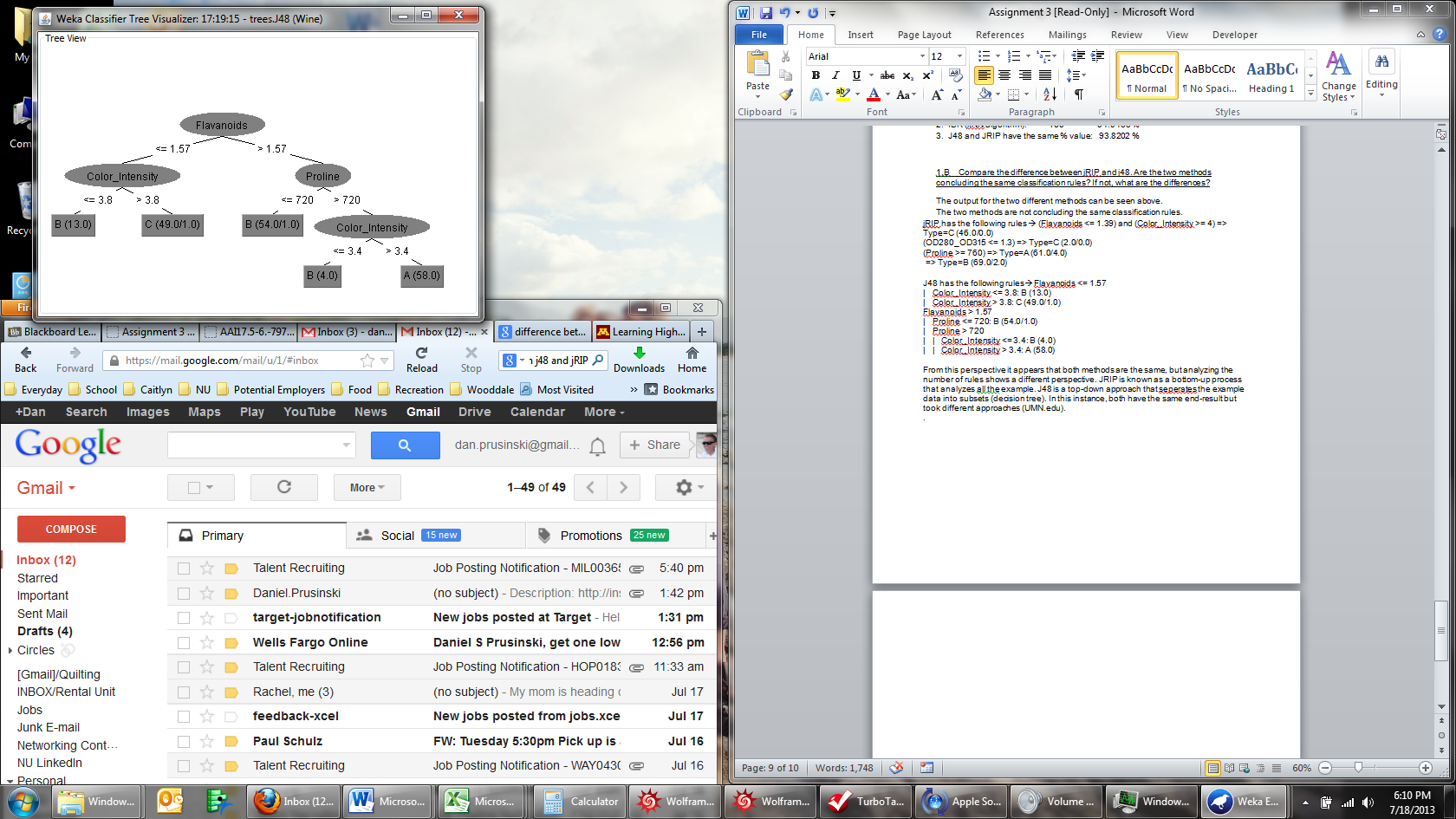
Flavanoids > 1.57

| Proline <= 720: B (54.0/1.0)

| Proline > 720

| | Color\_Intensity <= 3.4: B (4.0)

| | Color\_Intensity > 3.4: A (58.0)

From this perspective it appears that both methods are the same, but analyzing the number of rules shows a different perspective. JRIP is known as a bottom-up process that analyzes all the examplee. J48 is a top-down approach that separates the example data into subsets (decision tree). In this instance, both have the same end-result but took different approaches (UMN.edu). The values and order of the classification rules are different.

The decision tree demonstrataes the top down approach from J48.

.

Part 2: Changing the data and algorithms:

The next step in this exploratory data analysis (EDA) is to make a few changes to the data as well as the algorithms in an effort to learn how they respond in a different environment.

Changes to the data: In my preliminary analysis of the variables, Proline has the following summary statistics:

Proline

mean 1115.8573 519.8261 629.683

std. dev. 220.0034 154.7719 113.0791

weight sum 59 71 48

precision 11.6833 11.6833 11.6833

The mean is much higher than any other variable by quite a bit, in fact no other mean for a variable is larger than 21. While the standard deviation suggests a normal distribution, I am still going to remove this variable from the overall data.

Changes to Algorithms:

At first I wanted to use the “Use training set” option in Weka, but upon further research I found this approach uses the whole data set provided which is pointless if one cannot test it on other data. Therefore, I will not be using this option to explore algorithmic changes.

I want to know the output of J48, JRIP Naïve Bayes, and IBK if the number of folds are changed from 10 to 5. The folds have the following effect on analyzing the data:

**Use 10 fold CV (Rushdi Shams, stackoverflow.com)**

* Weka takes 100 labeled data
* it produces 10 equal sized sets. Each set is divided into two groups: 90 labeled data are used for training and 10 labeled data are used for testing.
* it produces a classifier with an algorithm from 90 labeled data and applies that on the 10 testing data for set 1.
* It does the same thing for set 2 to 10 and produces 9 more classifiers
* it averages the performance of the 10 classifiers produced from 10 equal sized (90 training and 10 testing) set

Given that our data is not massive, by lessening the folds it gives the algorithm less opportunity to learn and is based more on the actual initial production of the algorithm. In addition, I want to simulate a “real-life” situation where I would be using one of these algorithms on millions/billions of rows of data and I will utilize a smaller percentage of the data. The default is 66%, and I will be changing that to 33% in an effort to simulate a larger data set than what is actually given in this instance.

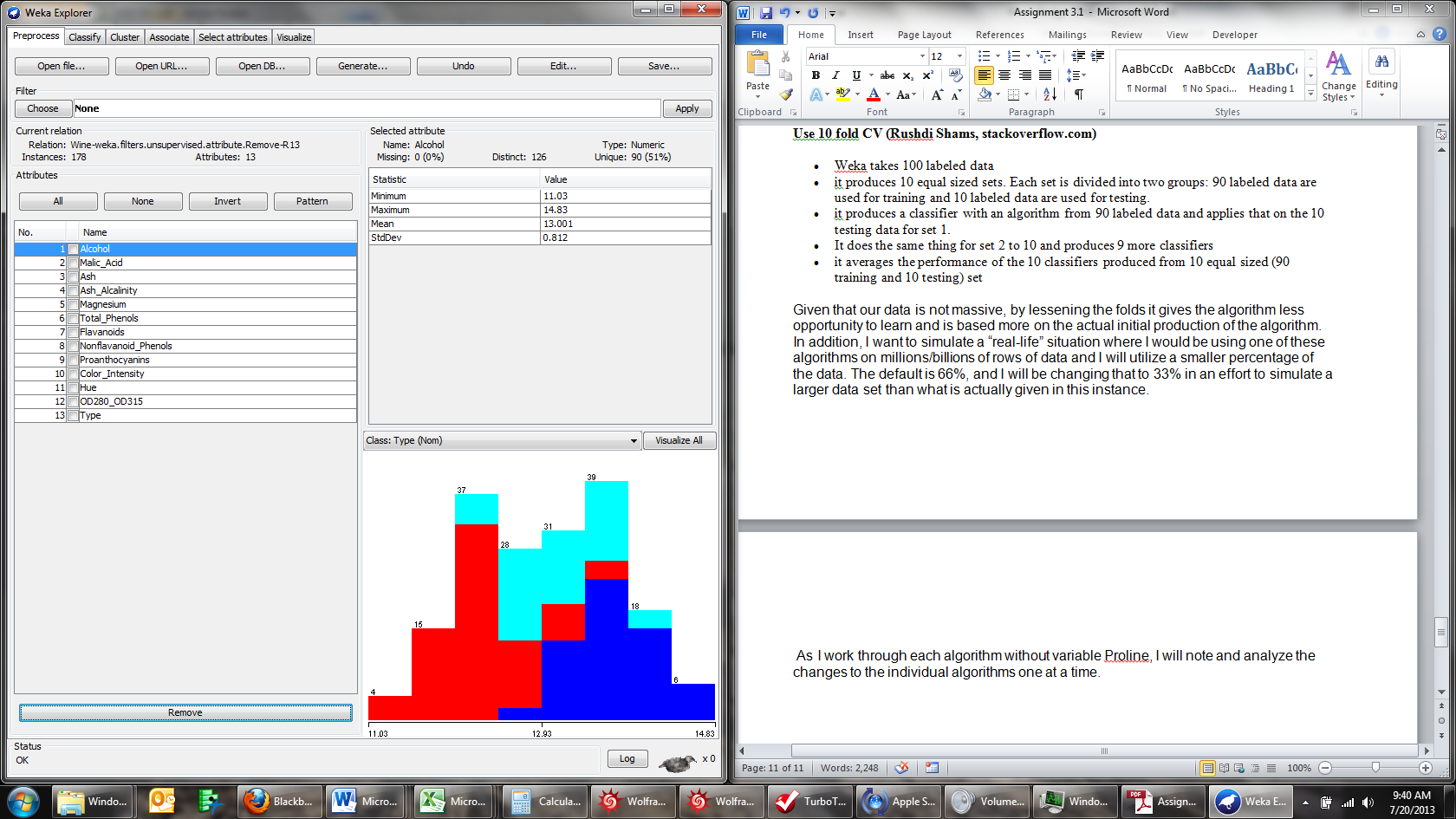
As I work through each algorithm without variable Proline, I will note and analyze the changes to the individual algorithms one at a time.

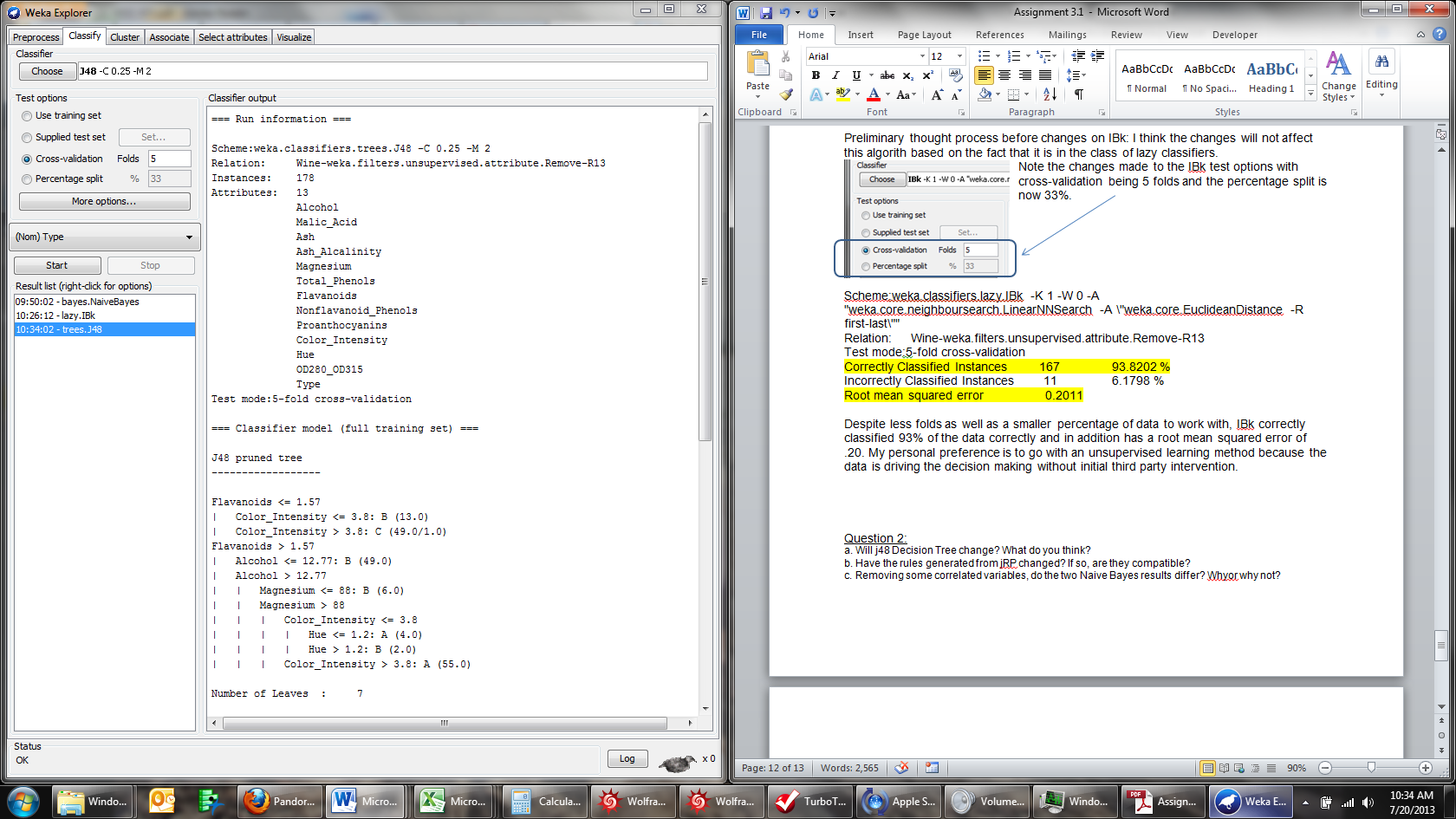
**Question 2:**

**a. Will j48 Decision Tree change? What do you think?**

The j48 Decision Tree will change because the Proline variable has been removed. Given that I am only using 5 folds, I think that will negatively affect the process as well.

Documented Changes:

Proline has been removed from the list of attributes being analyzed.

Note the changes made to the J48 test options with cross-validation being 5 folds and the percentage split is now 33%.

After change analysis:

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: Wine-weka.filters.unsupervised.attribute.Remove-R13

Instances: 178

Attributes: 13

Test mode:5-fold cross-validation

Number of Leaves : 7

Size of the tree : 13

Flavanoids <= 1.57

| Color\_Intensity <= 3.8: B (13.0)

| Color\_Intensity > 3.8: C (49.0/1.0)

Flavanoids > 1.57

| Alcohol <= 12.77: B (49.0)

| Alcohol > 12.77

| | Magnesium <= 88: B (6.0)

| | Magnesium > 88

| | | Color\_Intensity <= 3.8

| | | | Hue <= 1.2: A (4.0)

| | | | Hue > 1.2: B (2.0)

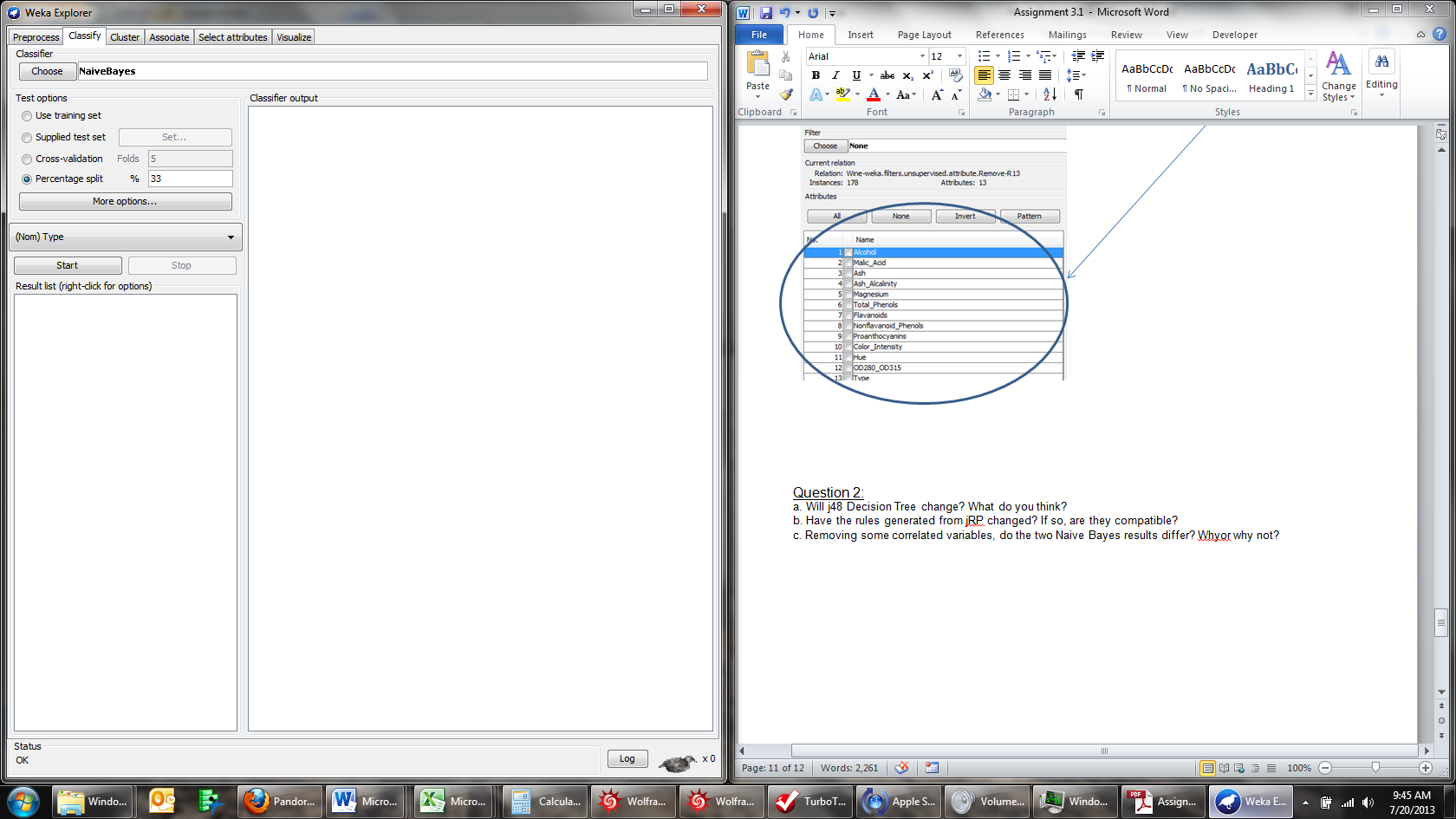
| | | Color\_Intensity > 3.8: A (55.0)

Correctly Classified Instances 159 89.3258 %

Incorrectly Classified Instances 19 10.6742 %

Root mean squared error 0.2576

Despite less folds as well as a smaller percentage of data to work with, J48 correctly classified 89% of the data correctly and in addition has a root mean squared error of .25. In my opinion, removing Proline has the greatest effect on J48 based on how the algorithm is a top down approach and since Proline was used in node that was not the bottom leaf I automatically assumed it would take an accuracy hit.



Note the changes made to the Naïve Bayes test options with cross-validation being 5 folds and the percentage split is now 33%.

After change analysis:

Scheme:weka.classifiers.bayes.NaiveBayes

Relation: Wine-weka.filters.unsupervised.attribute.Remove-R13

Instances: 178

Attributes: 13

Test mode:5-fold cross-validation

Correctly Classified Instances 170 95.5056 %

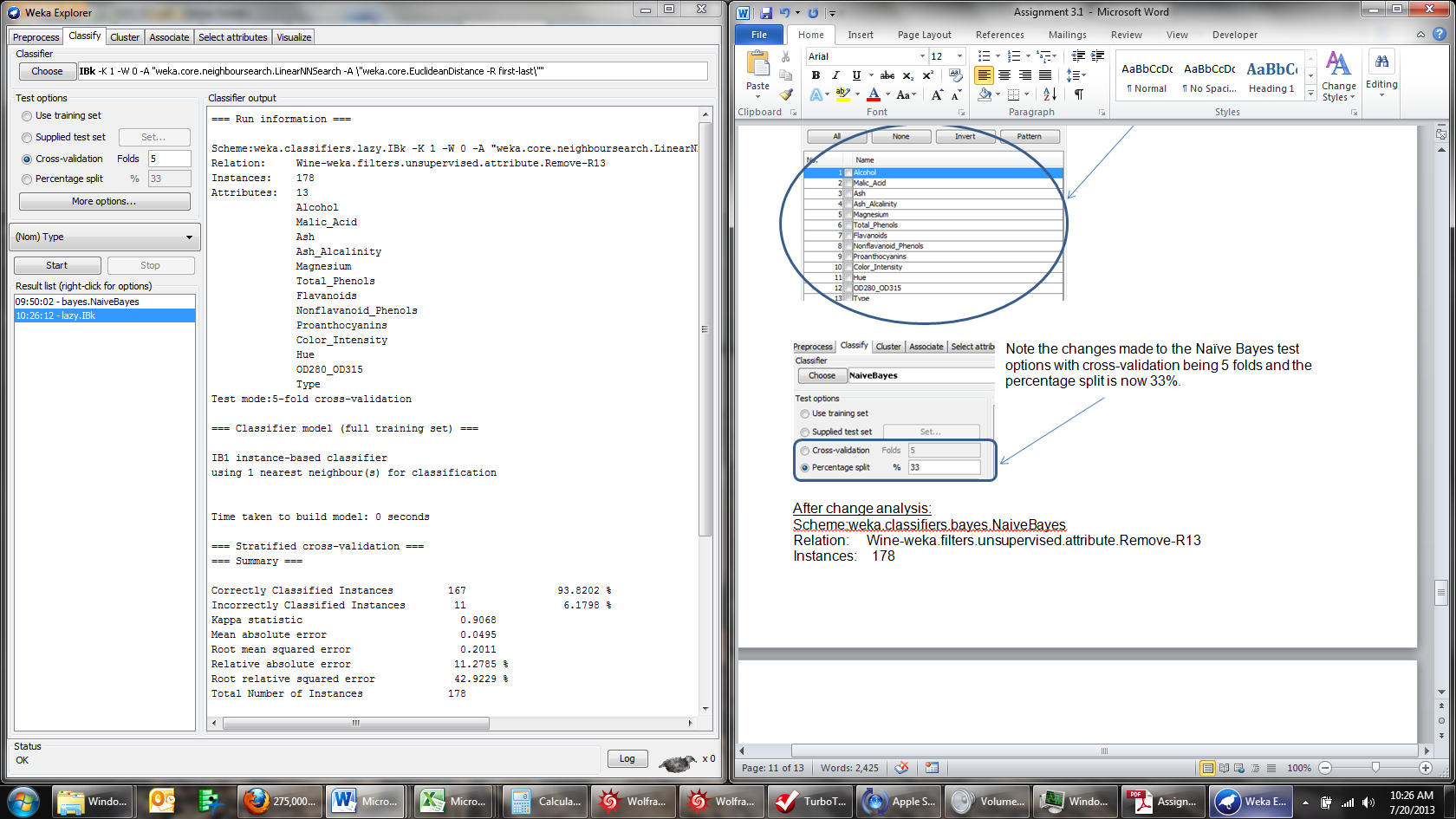
Incorrectly Classified Instances 8 4.4944 %

Total Number of Instances 178

Root mean squared error 0.1555

Despite less folds as well as a smaller percentage of data to work with, Naïve Bayes correctly classified 95% of the data correctly and in addition has a root mean squared error of .1555. There are quite a few variables in this data set that have a strong correlation coefficient, which is one reason that the Naïve Bayes is doing so well at classification.

Preliminary thought process before changes on IBk: I think the changes will not affect this algorith based on the fact that it is in the class of lazy classifiers.

Note the changes made to the IBk test options with cross-validation being 5 folds and the percentage split is now 33%.

After change analysis:

Scheme:weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Relation: Wine-weka.filters.unsupervised.attribute.Remove-R13

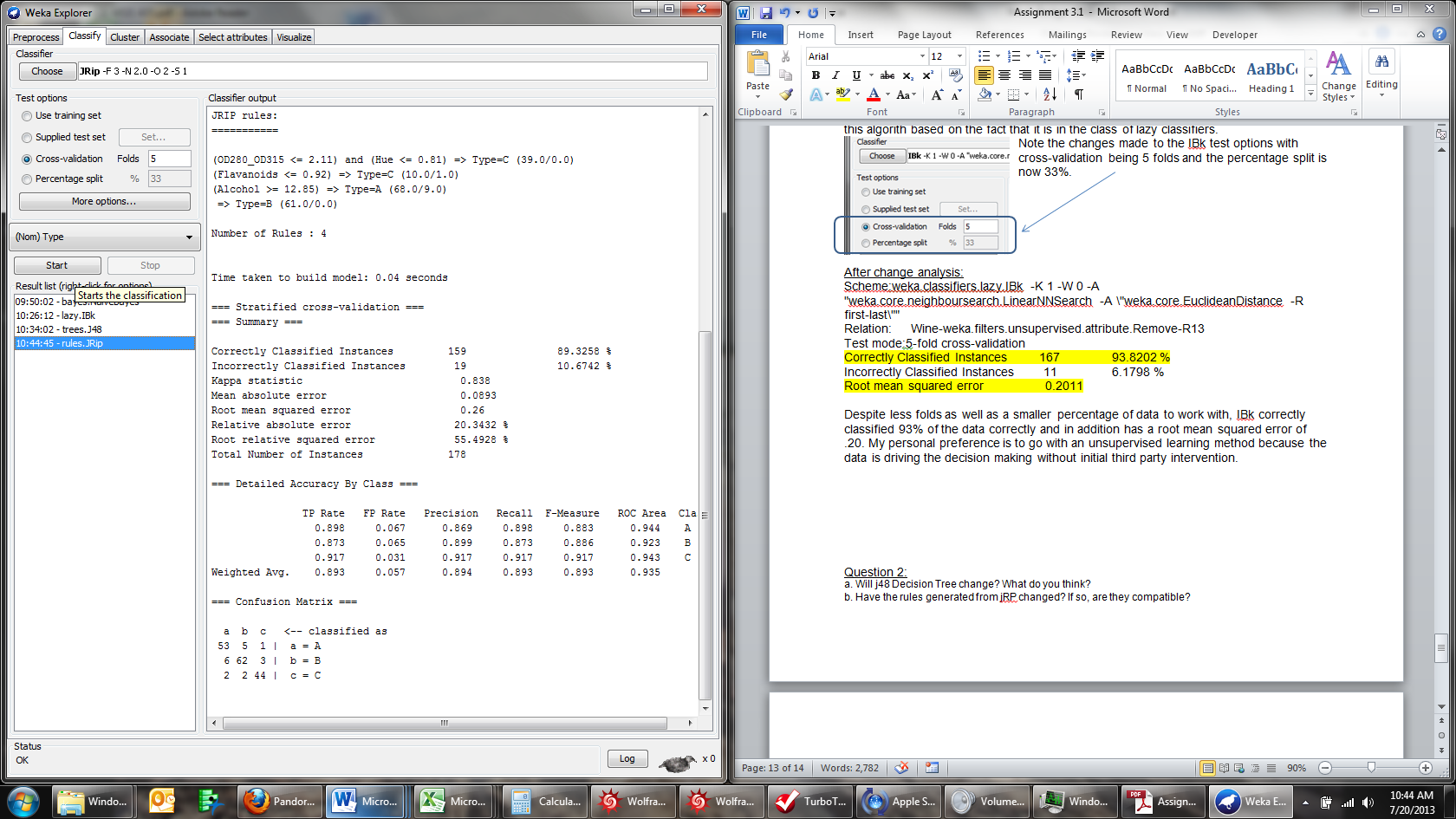
Test mode:5-fold cross-validation

Correctly Classified Instances 167 93.8202 %

Incorrectly Classified Instances 11 6.1798 %

Root mean squared error 0.2011

Despite less folds as well as a smaller percentage of data to work with, IBk correctly classified 93% of the data correctly and in addition has a root mean squared error of .20. My personal preference is to go with an unsupervised learning method because the data is driving the decision making without initial third party intervention.

Note the changes made to the JRIP test options with cross-validation being 5 folds and the percentage split is now 33%.

After change analysis:

Scheme:weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -S 1

Relation: Wine-weka.filters.unsupervised.attribute.Remove-R13

Test mode:5-fold cross-validation

(OD280\_OD315 <= 2.11) and (Hue <= 0.81) => Type=C (39.0/0.0)

(Flavanoids <= 0.92) => Type=C (10.0/1.0)

(Alcohol >= 12.85) => Type=A (68.0/9.0)

=> Type=B (61.0/0.0)

Number of Rules : 4

Correctly Classified Instances 159 89.3258 %

Incorrectly Classified Instances 19 10.6742 %

Root mean squared error 0.26

Given the changes, JRIP and J48 have very similar output.

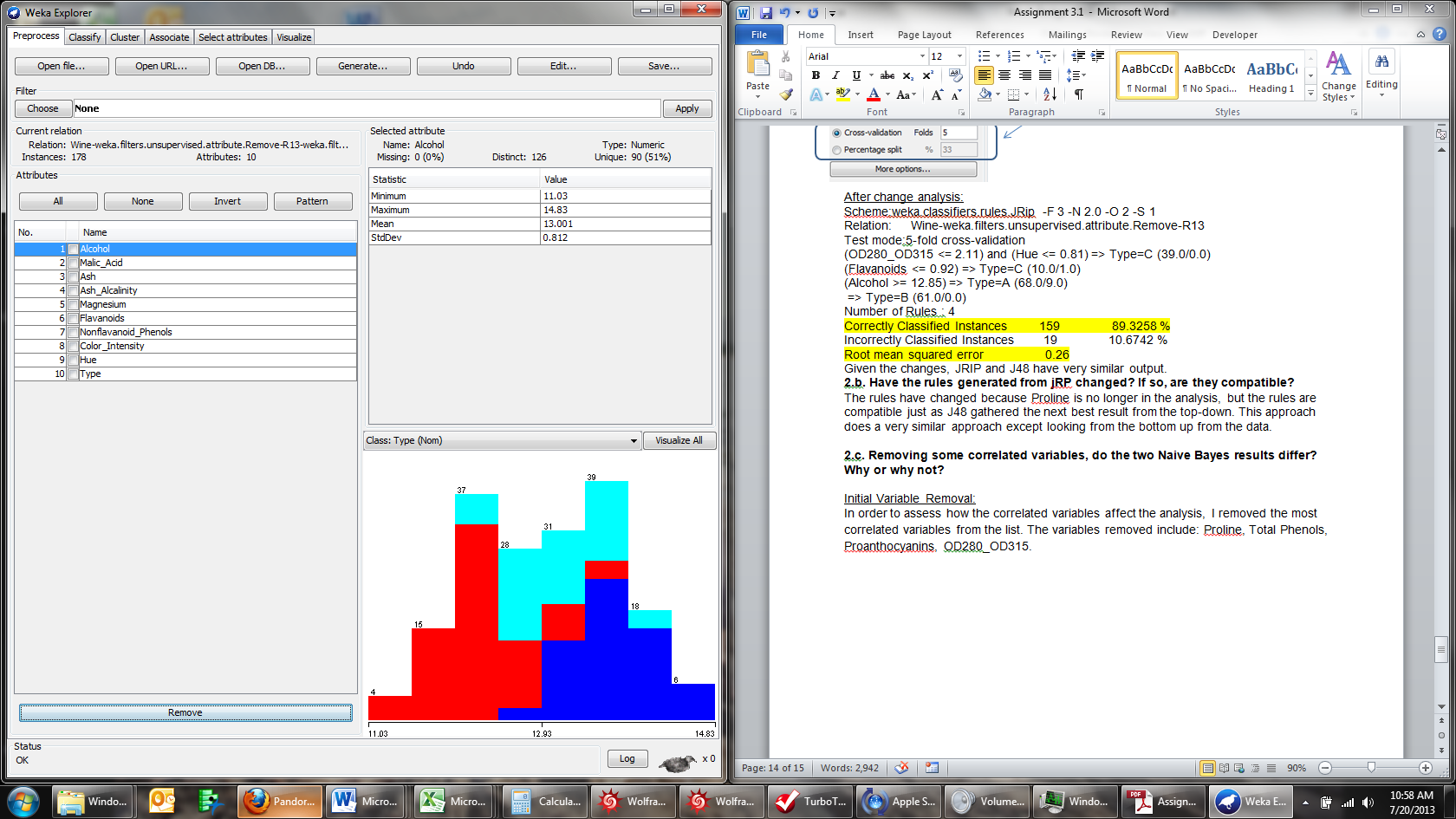
**2.b. Have the rules generated from jRP changed? If so, are they compatible?**

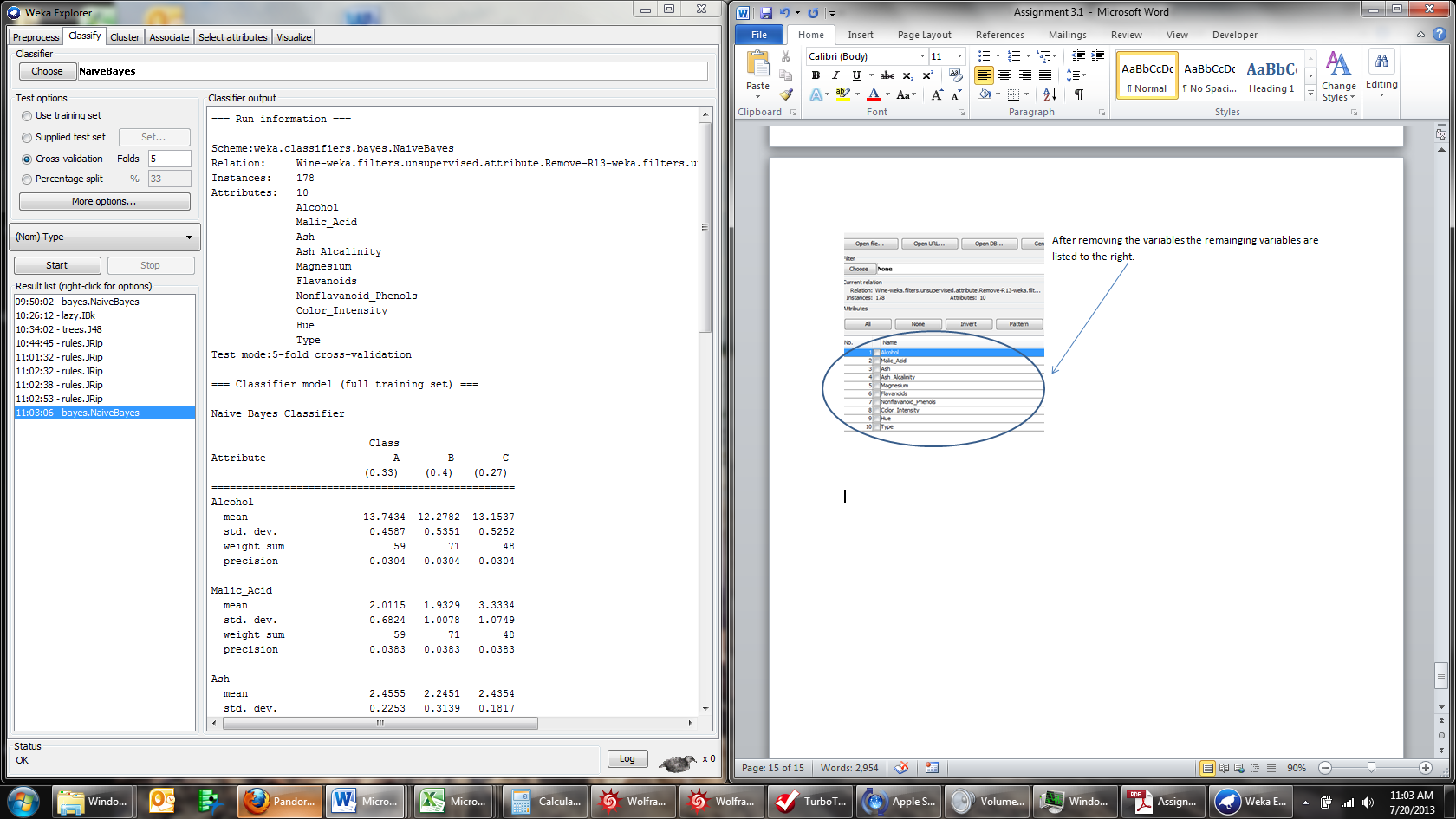
The rules have changed because Proline is no longer in the analysis, but the rules are compatible just as J48 gathered the next best result from the top-down. This approach does a very similar approach except looking from the bottom up from the data.

**2.c. Removing some correlated variables, do the two Naive Bayes results differ? Why or why not?**

Initial Variable Removal:

In order to assess how the correlated variables affect the analysis, I removed the most correlated variables from the list. The variables removed include: Proline, Total Phenols, Proanthocyanins, OD280\_OD315.

After removing the variables the remainging variables are listed to the right.



Note the changes made to the JRIP test options with cross-validation being 5 folds and the percentage split is now 33%.

After change analysis:

Scheme:weka.classifiers.bayes.NaiveBayes

Relation: Wine-weka.filters.unsupervised.attribute.Remove-R13-weka.filters.unsupervised.attribute.Remove-R6,9,12

Test mode: 5-fold cross-validation

Correctly Classified Instances 171 96.0674 %

Incorrectly Classified Instances 7 3.9326 %

Root mean squared error 0.1423

Surprisingly, removing the correlated variables boosted the correctly classified percentage and shrunk the root mean squared error. Initially, I thought the correlation was helping the Naïve Bayes classifier, but now I wonder if the data suffers from a high value of multicollinearity given the removing the correlated variables had a positive effect on the overall classification output.

Concluding Thoughts:

Through this exercise, I have learned how to use 4 different classification algorithms on a relatively small data set. The next step would be to apply this to a rather large data set in an effort to analyze what happens as more data is brought into the decision making process. On a personal note, all four examples had a rather high percentage of correct classification. If I was tasked with picking one, I would resort to an algorithm that correctly classifies above 93% and best adheres to the principle of parsimony.