Final Project: 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 and Initial Observations

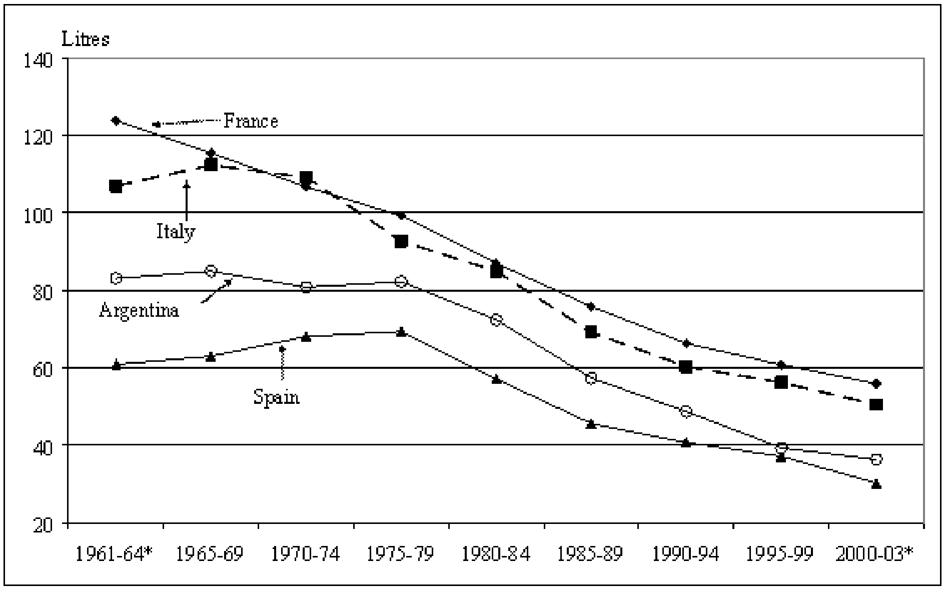
Nearly all major research, business, and financial decision making requires data mining. At the heart of data mining, mathematical principles are applied to structured data, tables, as well as unstructured data. This research paper entails conducting an exploratory data analysis (EDA) on one major data set and five nested data sets. Beyond the six different data sets, the relationship between all thirteen variables will be explored such that the most efficient algorithm is chosen. I will be utilizing the EDA paradigm and structure put forth by Bruce Ratner found in his book *Statistical and Machine-Learning Data Mining*:

> Objective: Explore the relationship between thirteen variables throughout 6 different data sets utilizing an array of different algorithms to recommend the best suited algorithm that best classifies and explains the data.
>
> Data: The data has been aggregated into six different sets and each set is made up of 178 instances, these instances are the same throughout each set. There are no unique instances to a particular data set.
>
> Analysis: Scatter plots and correlation coefficients will be used to study the nature of the relationships between the variables. I will conduct multiple analyses and briefly comment on the overall findings. Weka is the processing software that will be used to apply the different algorithms on the data.
>
> Algorithms/Models: After assessing the data, 7 different algorithms will be fit to the data. Listed below are the seven:
>
> 1. Naïve Bayes
> 2. J48
> 3. IBK
> 4. Artificial Neural Network
> 5. Logistic Regression
> 6. Simple Logistic
> 7. JRIP
>
> The algorithms listed above are truly expansive in that they span many different mathematical classifiers.
>
> Results/Interpretation: The models produced by each algorithm will be individually assessed and then compared to the other algorithms.

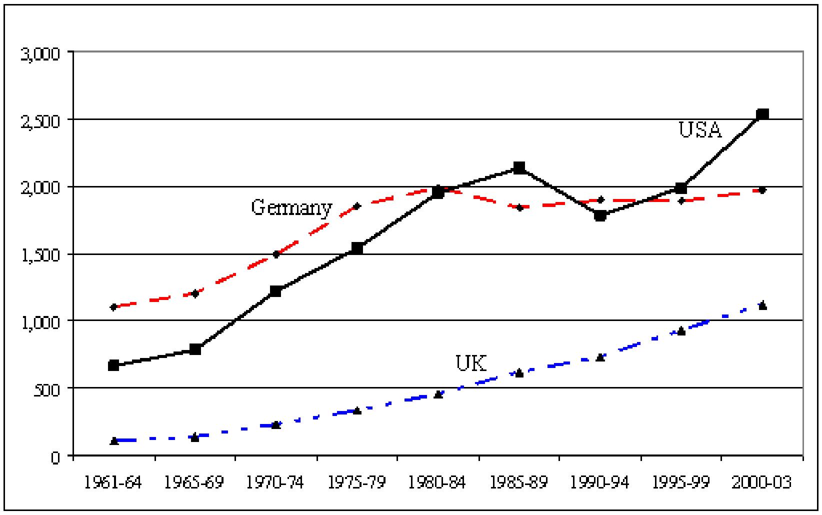
A properly executed EDA for management must reflect that the data was the driving force behind constructing the model. The steps outlined above are ordered such that the data drives which model is used, and the analyst's personal bias is mitigated.

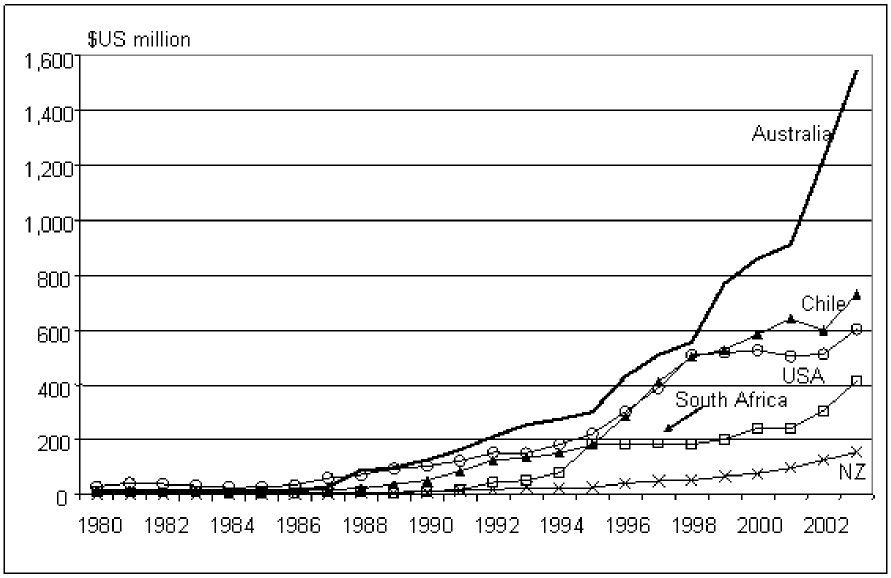
Wine Industry Analysis

Before delving into the algorithms and models, a brief analysis of the wine industry as a whole is helpful for background information and understanding the financially lucrative nature are cultivating grapes. Global consumption for wine has steadily increased over the past ten years. The world's largest wine producing countries have experienced an overall decline in wine consumption. These two factors have driven down the global price for grapes, as well as increased the exportation of wine from traditional wine consuming countries to new markets (Wittwer & Rothfield). The graph above shows the consumption rates of wine (liters) for the four largest wine producing countries. It can be seen that there has been a negative trend of wine consumption for the last 50 years in the traditional wine consumer countries. From the 1960's on, the world shifted to a more inclusive trading platform, which resulted in additional spirits in these countries. On the global stage, the USA, Germany, and the



UK all increased their consumption of wine and importation of wine during the same time period. Essentially the two graphs show that while wine consumption decreased in France, Italy, Spain, and Argentina, it increased in the US, UK and Germany. While new markets were expanded into over the past 40 years, it is vitally important to analyze which countries exported the most wine during this time period.





The graph to the left shows which countries exportation of wine grew the largest over the past 20years. It can clearly be seen that Australia, Chile, and the US have been winning the export war of wine. Analyzing production and exportation of wine is a helpful exercise in understanding successful export strategies.

Analysis of Variables

Data set analyzed: Wine All Data

Instances: 178

Attributes: 14

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)
=======================================================
```

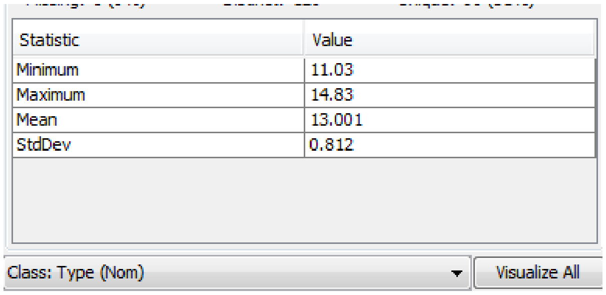
Alcohol

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.

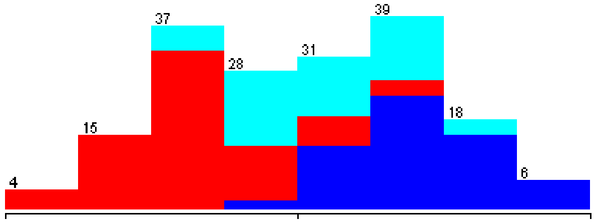
weight sum 59 71 48 ← This sum validates what is shown in in the green circle above about B having more weight in the analysis.

precision 0.0304 0.0304 0.0304



Statistic	Value
Minimum	11.03
Maximum	14.83
Mean	13.001
StdDev	0.812

The min and max can be seen to the left. The STDV is all three classes combined.

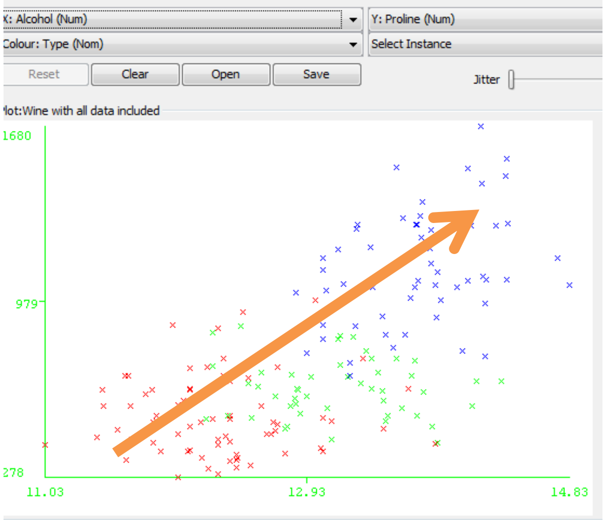


Listed below is the correlation of alcohol with the other variables:
M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pro= Proline

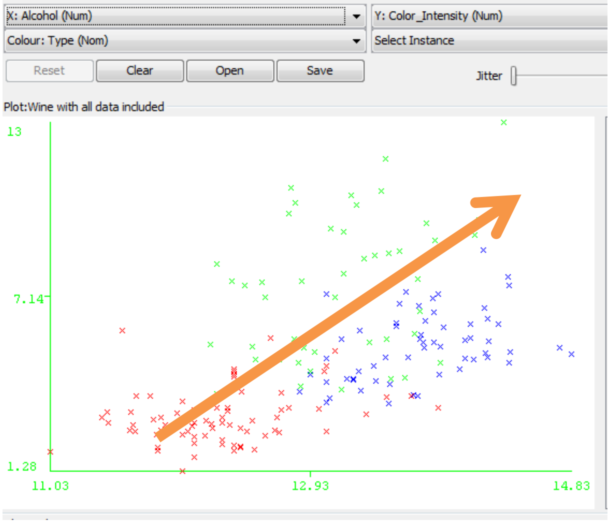
Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pro
1	0.09	0.21	-0.31	0.27	0.29	0.24	-0.16	0.14	0.55	-0.07	0.07	0.64

The only significant correlations are between Color Intensity and Proline.

Visually, the relationship is shown below:



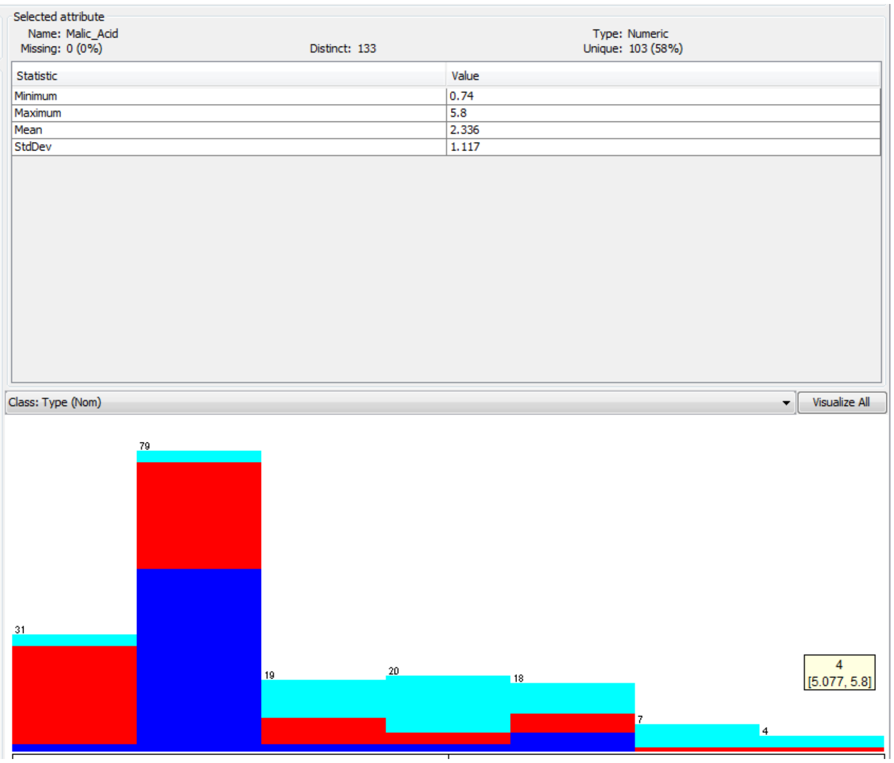
K: Alcohol (Num)
Y: Proline (Num)
Colour: Type (Nom)
lot:Wine with all data included



X: Alcohol (Num)
Y: Color_Intensity (Num)
Colour: Type (Nom)
Plot:Wine with all data included

It can be visually verified that the correlation is linear in nature, notice how Color Intensity is harder to discern a linear pattern. The conclusion I draw from the correlation is that these three variables are indicative of one another slightly, and may perform similarly with certain algorithms. Given that the correlative relationship has been identified here, I will not highlight how Proline and Color_Intensity are correlated with Alcohol.

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.



Selected attribute
Name: Malic_Acid
Missing: 0 (0%)
Distinct: 133
Type: Numeric
Unique: 103 (58%)

Statistic	Value
Minimum	0.74
Maximum	5.8
Mean	2.336
StdDev	1.117

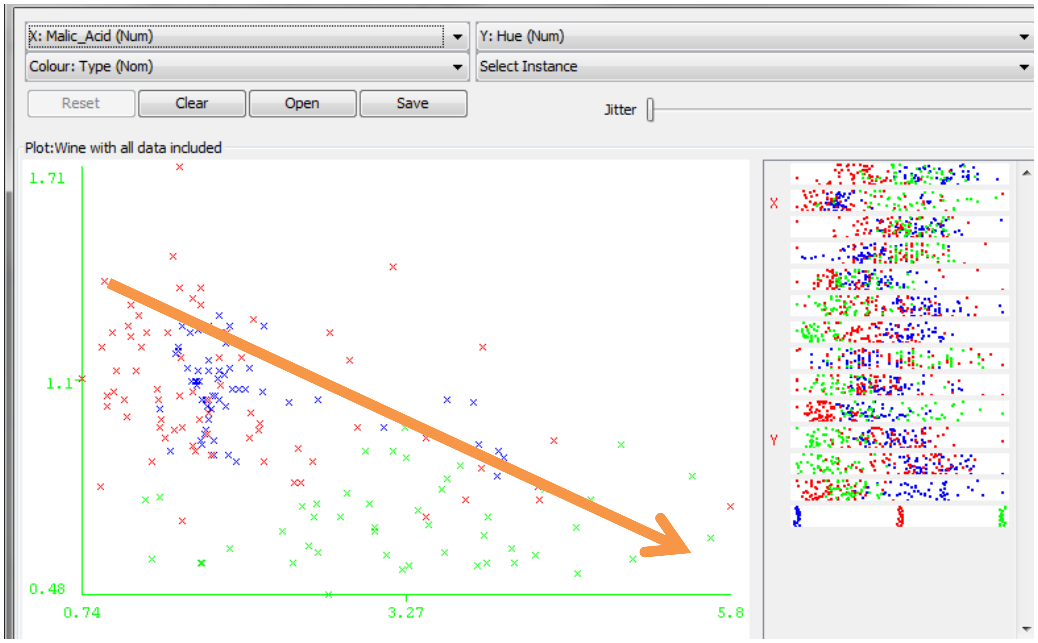
The distribution looks positively skewed, also there is quite a range from the Min and Max, which can be seen in the STDV.

weight sum   59   71   48
  precision   0.0383   0.0383   0.0383

Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.09	1	0.16	0.29	-0.05	-0.34	-0.41	0.29	-0.22	0.25	-0.56	-0.37	-0.19

The only significant correlation is between Hue.



X: Malic_Acid (Num)

Y: Hue (Num)

Colour: Type (Nom)

Select Instance

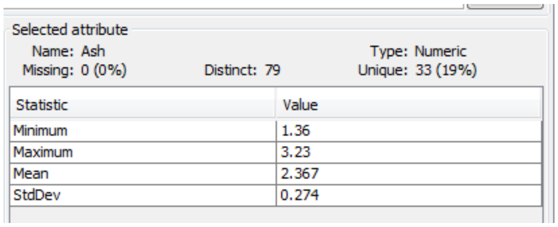
Plot:Wine with all data included

It can be seen that this is a negative correlation, but still a relationship. I would expect that variables to act in an inverse correlative nature throughout the models. On page XXX, one can see the complete scatter plot matrix. From this scatter plot, is can be seen that vast majority of the plots are positively skewed.

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

There is nothing out of the ordinary for this variable.



Selected attribute

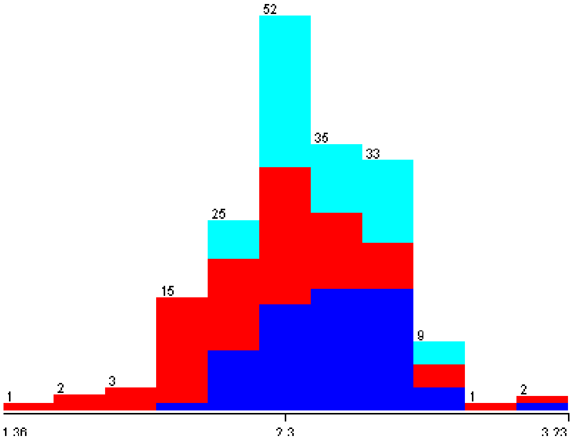
Name: Ash Type: Numeric

Missing: 0 (0%) Distinct: 79 Unique: 33 (19%)

Statistic	Value
Minimum	1.36
Maximum	3.23
Mean	2.367
StdDev	0.274

The min and max along with the STDV demonstrate a normal distribution. Visually, this is verified by a plot of the points.

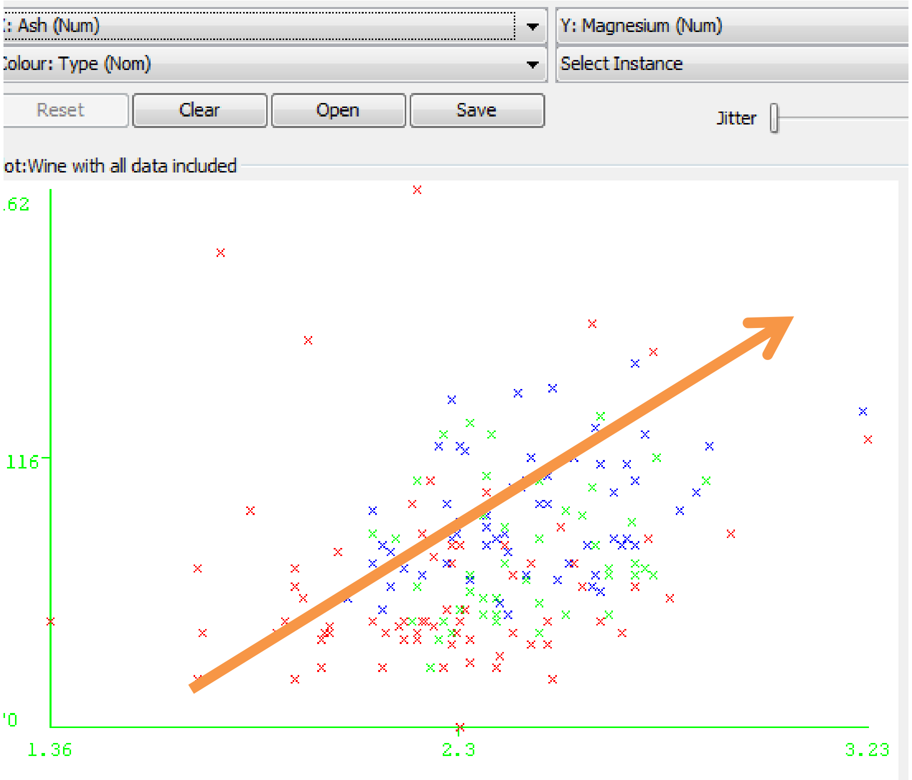
Notice how if there was a curve it would follow a normal distribution.



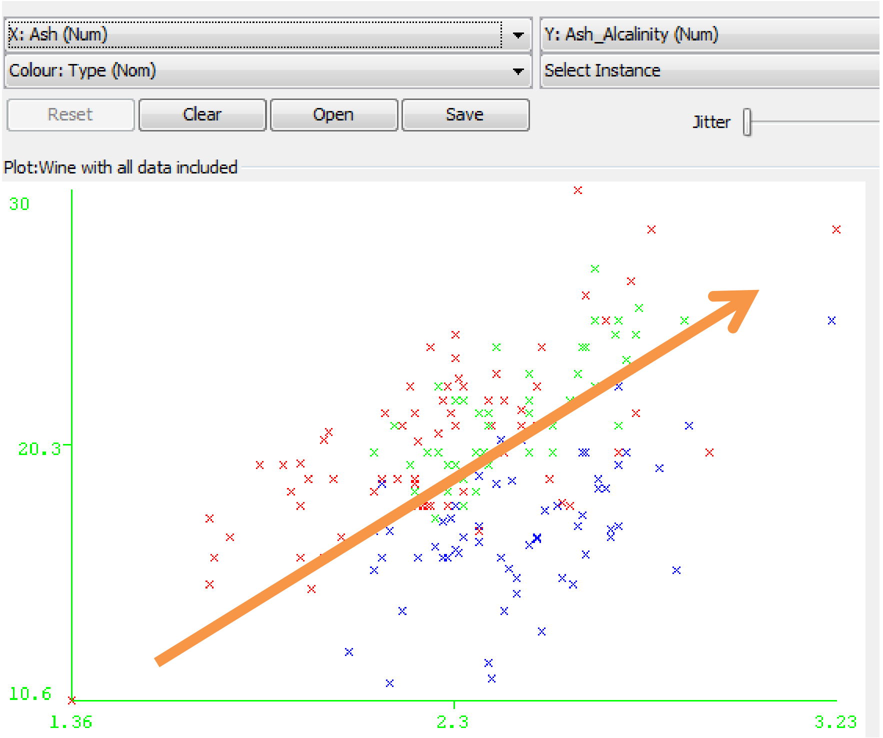
Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.21	0.16	1	0.44	0.29	0.13	0.12	0.19	0.01	0.26	-0.07	0	0.22

Numerically, there is not a strong correlation between and Ash and the other variables. Graphically, it appears that Ash and Magnesium as well as Ash_Alcalinity are linearly related.


: Ash (Num)
Y: Magnesium (Num)
:olour: Type (Nom)
ot:Wine with all data included

There appears to be a linear relationship. This is a discovery that may disrupt result further in the analysis.



X: Ash (Num)

Y: Ash_Alcalinity (Num)

Colour: Type (Nom)

Plot:Wine with all data included

There appears to be a linear relationship. This is a discovery that may disrupt result further in the analysis

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

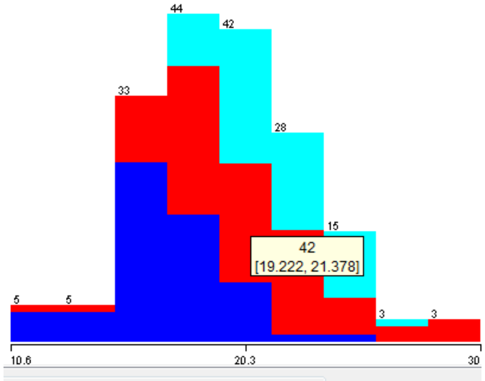
The Min = 10.6

The Max = 30

The Mean = 19.495

The STDV = 3.34

Overall these numbers reflect a fairly normal distribution. Graphically, the distribution is show below:



Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

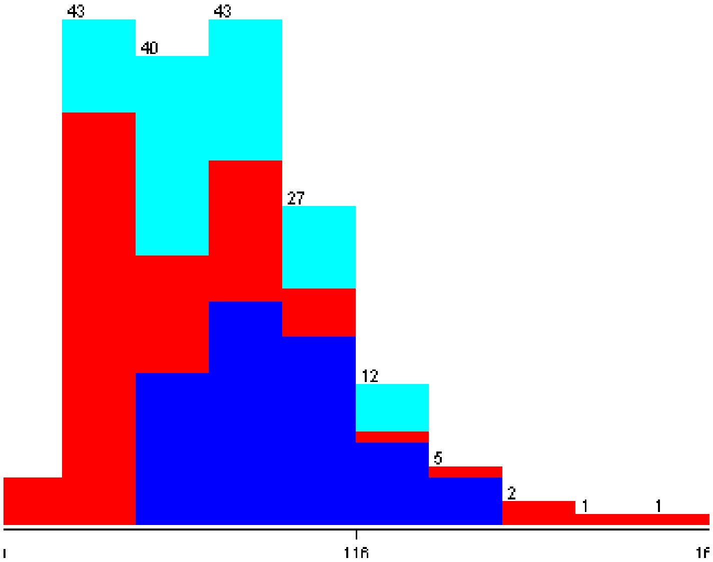
Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
-0.31	0.29	0.44	1	-0.08	-0.32	-0.35	0.36	-0.2	0.02	-0.27	-0.28	-0.44

These two relationships were discussed above.

<u>Magnesium</u>

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	

The Min = 70
The Max = 162
The Mean = 99
The STDV = 14.282

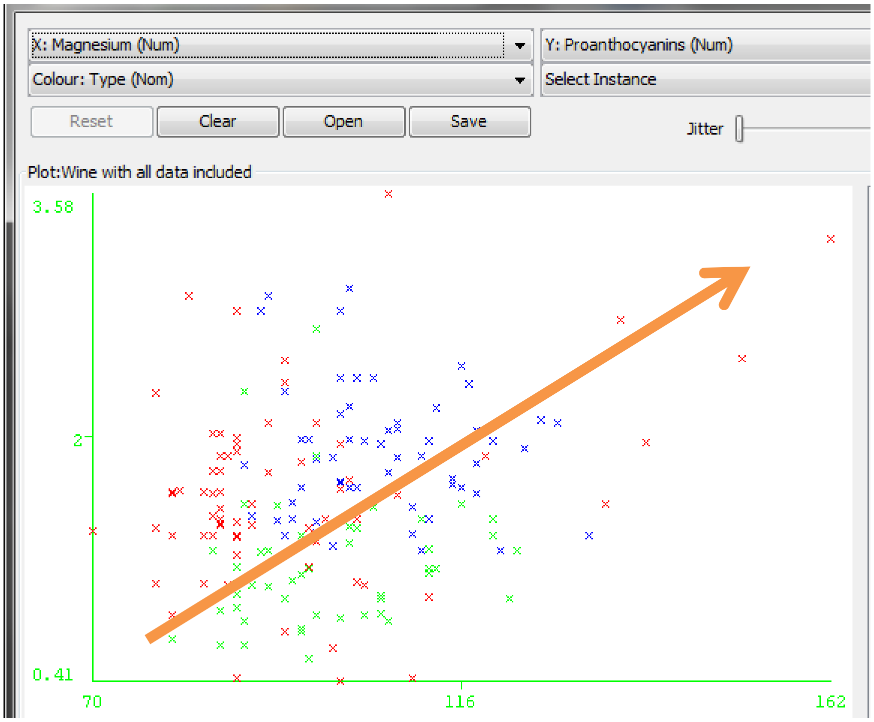


Notice the large positive skew in the distribution. When looking at the scatter plot matrix, one can see that the distribution for the first 5 variables are all highly skewed.

Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.27	-0.05	0.29	-0.08	1	0.21	0.2	-0.26	0.24	0.2	0.06	0.07	0.39

Numerically, there is not a strong correlation between and Ash and the other variables. Graphically, it appears that Mag and Proanthocyanins are linearly related.



X: Magnesium (Num)

Y: Proanthocyanins (Num)

Colour: Type (Nom)

Select Instance

Plot:Wine with all data included

<u>Total_Phenols</u>

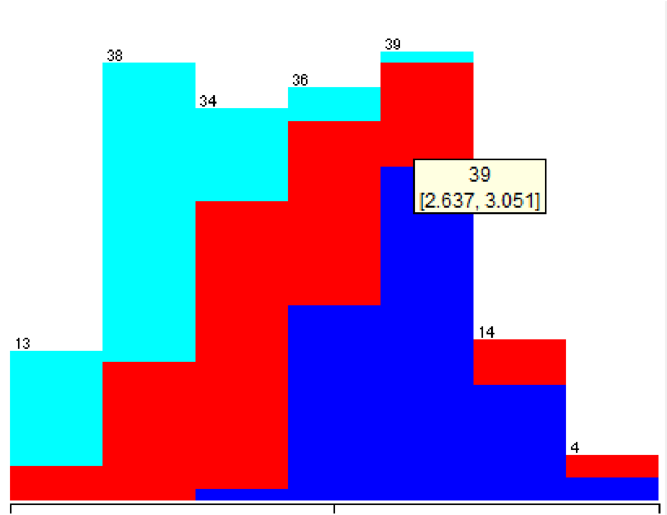
  mean   2.8396   2.2618   1.681

  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

The Min = .98



The Max = 3.88

The Mean = 2.295

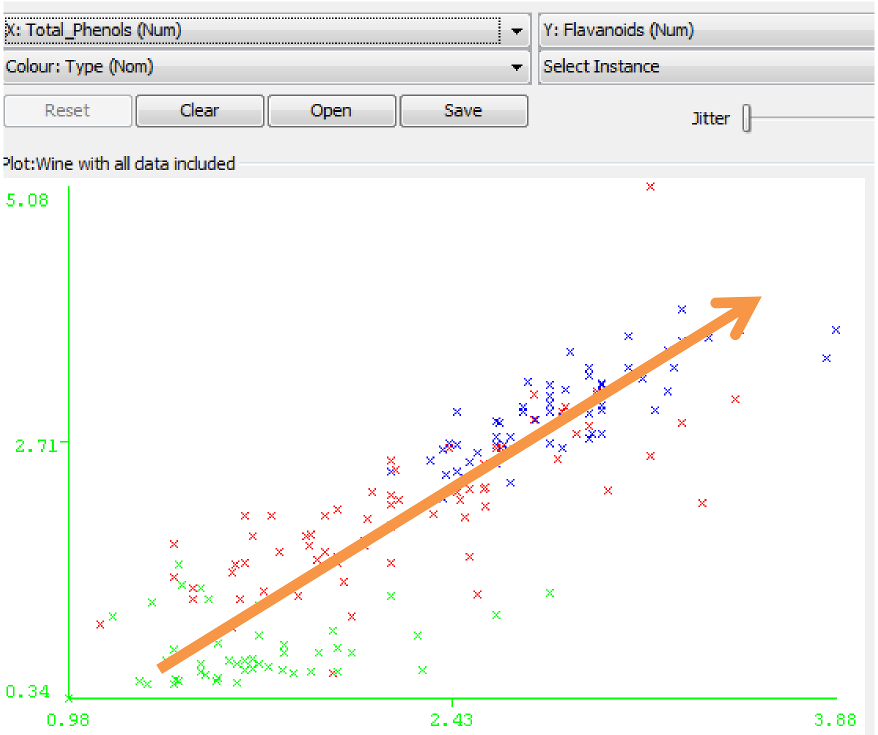
The STDV = .626

The ditribution is quite sloppy, which mean that this variable will be hard to forecast.

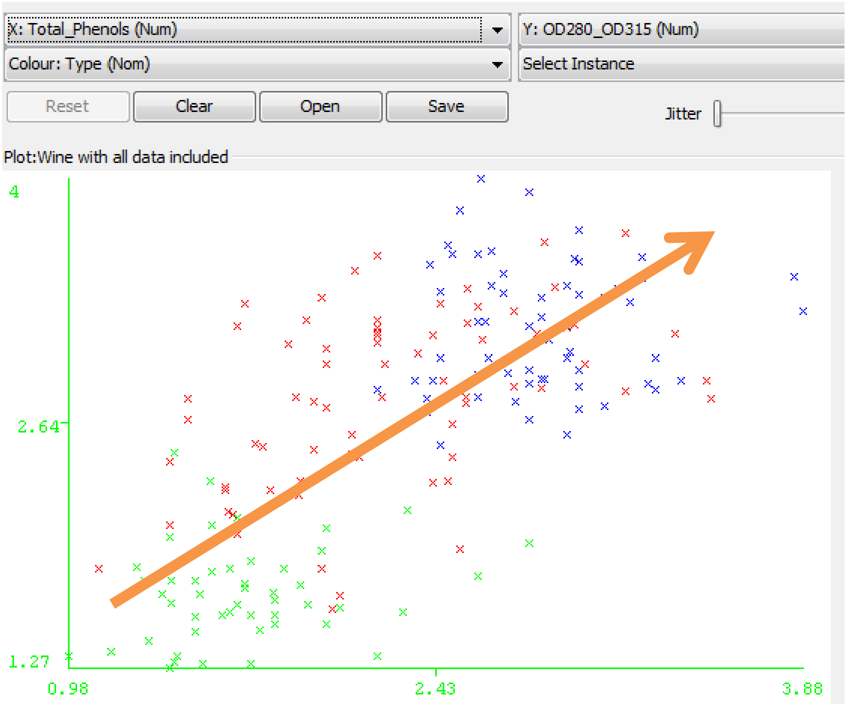
Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

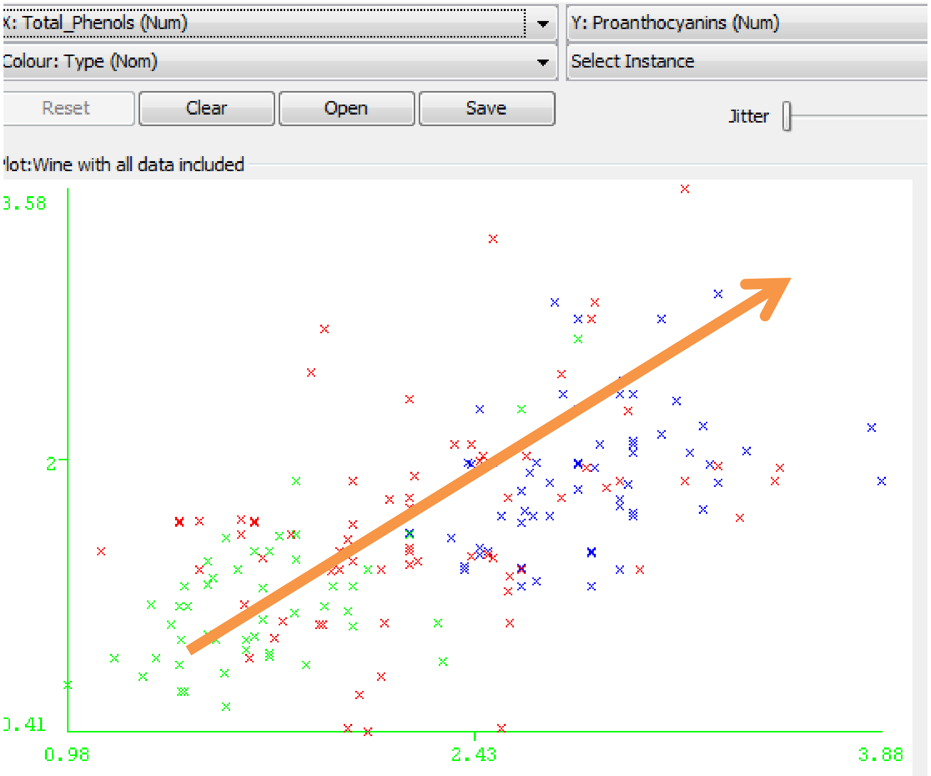
Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.29	-0.34	0.13	-0.32	0.21	1	0.86	-0.45	0.61	-0.06	0.43	0.7	0.5

Total Phenols is strongly correlated with Flavanoids, Proanthocyanins, and OD280_OD315.


X: Total_Phenols (Num)
Y: Flavanoids (Num)
Colour: Type (Nom)
Plot:Wine with all data included

Total Phenols and Flavanoids have the strongest correlation of variables that have been analyzed.


X: Total_Phenols (Num)
Y: OD280_OD315 (Num)
Colour: Type (Nom)
Plot:Wine with all data included


X: Total_Phenols (Num)

Y: Proanthocyanins (Num)

Colour: Type (Nom)

Plot:Wine with all data included

These three variables will all be relatively helpful predictors for one another moving forward in the analysis.

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

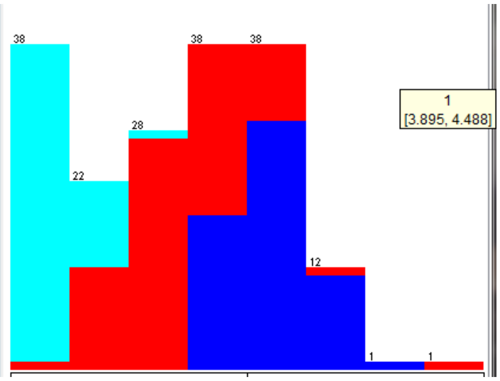
The Min = .34

The Max = 5.08

The Mean = 2.029

The STDV = .999

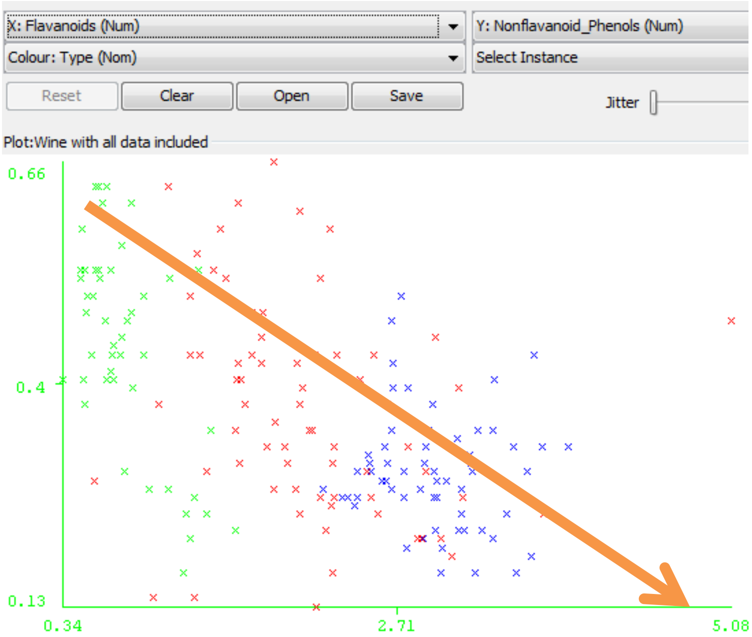
Overall there is a quite a gap between the min and max, which warrants concern for inferring this data set on a larger population.



Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

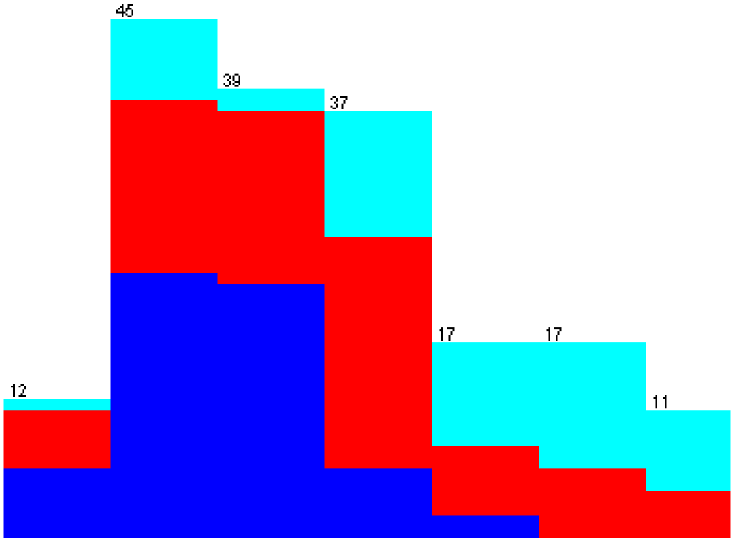
Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.24	-0.41	0.12	-0.35	0.2	0.86	1	-0.54	0.65	-0.17	0.54	0.79	0.49

The three high correlations have been highlighted above. Notice how there is a negative correlation with Nonflavanoid_Phenols.


X: Flavanoids (Num)
Y: Nonflavanoid_Phenols (Num)
Colour: Type (Nom)
Plot:Wine with all data included

This inverse relationship makes sense from the names of the variables, Nonflavanoid_Phenols and Flavanoids.

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
The Min = .13
The Max = .66
The Mean = .362
The STDV = .124



This is a fairly normal distribution.

Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

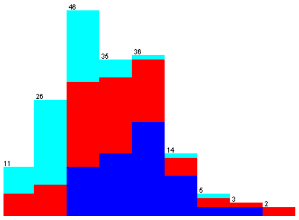
Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.16	0.29	0.19	0.36	-0.26	-0.45	-0.54	1	-0.37	0.14	-0.26	-0.5	-0.31

There are no numerical surprises here. This variable has an inverse relationship with Total_Phenols, Flavanoids, and OD280_OD315.

<u>Proanthocyanins</u>

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

The Min = .41
The Max = 3.58
The Mean = .572



This variable has a fairly normal distribution.

Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.14	-0.22	0.01	-0.2	0.24	0.61	0.65	-0.37	1	-0.03	0.3	0.52	0.33

This relationships highlighted above have been explained earlier.

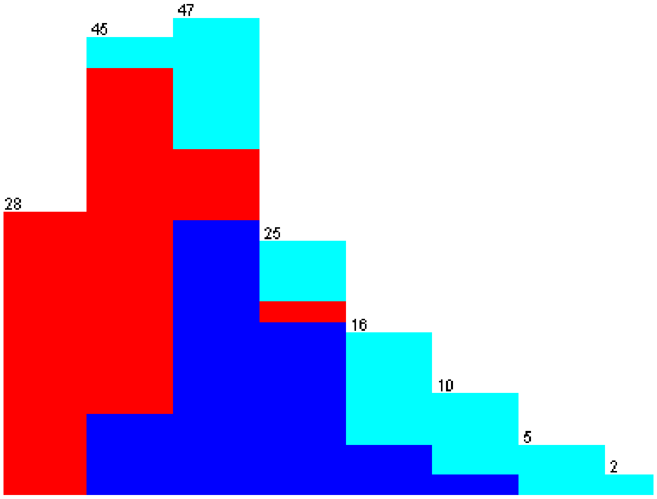
Color_Intensity

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

The Min = 1.28

The Max = 13

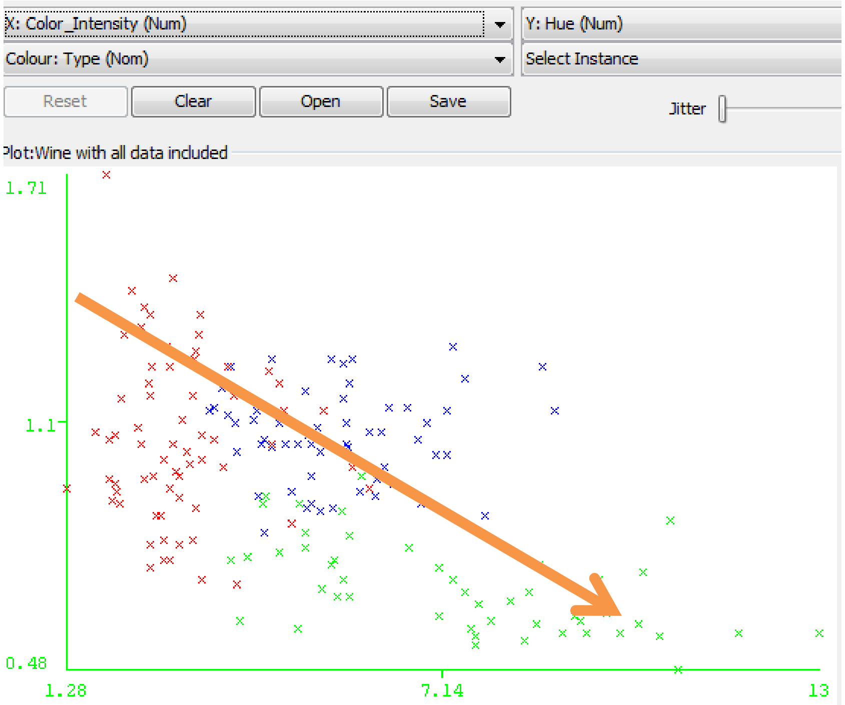
The Mean = 5.058



Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.55	0.25	0.26	0.02	0.2	-0.06	-0.17	0.14	-0.03	1	-0.52	-0.43	0.32

The relationship with Alcohol has already been called out, but the inverse relationship with color hue is logical.



X: Color_Intensity (Num)

Y: Hue (Num)

Colour: Type (Nom)

Plot:Wine with all data included

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.

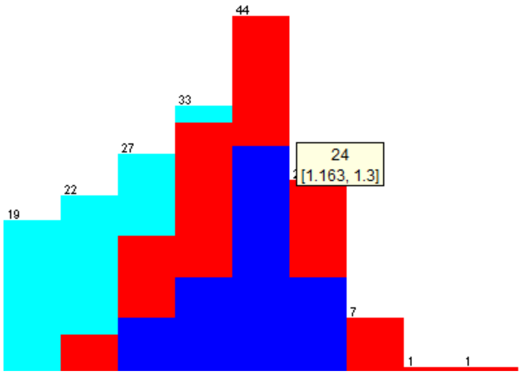
weight sum 59 71 48

precision 0.016 0.016 0.016

The Min = .48

The Max = 1.71

The Mean = .957



Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

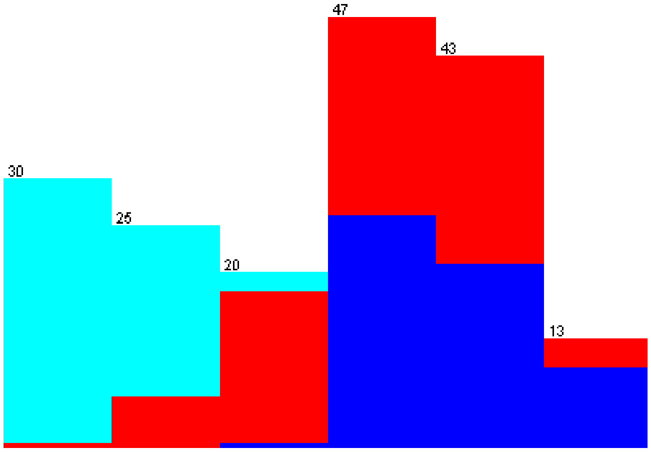
Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
-0.07	-0.56	-0.07	-0.27	0.06	0.43	0.54	-0.26	0.3	-0.52	1	0.57	0.24

All of these correlations have been addressed earlier.

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

The Min = 1.27
The Max = 4
The Mean = 2.612



Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.07	-0.37	0	-0.28	0.07	0.7	0.79	-0.5	0.52	-0.43	0.57	1	0.31

All of the major correlations have been addressed earlier.

<u>Proline</u>

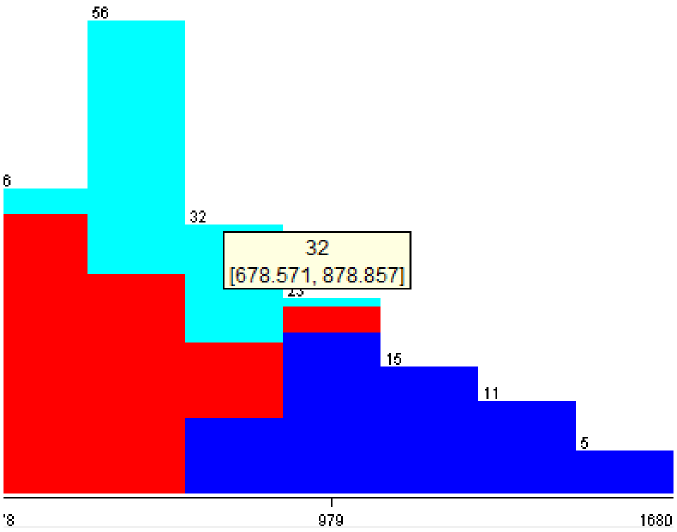
mean	1115.8573	519.8261	629.683
std. dev.	220.0034	154.7719	113.0791
weight sum	59	71	48
precision	11.6833	11.6833	11.6833

The Min = 278

The Max = 1680

The Mean = 746.893

Note how much larger this variable is compared to all the other variables. This has the potential to distort an analysis where size is a factor.



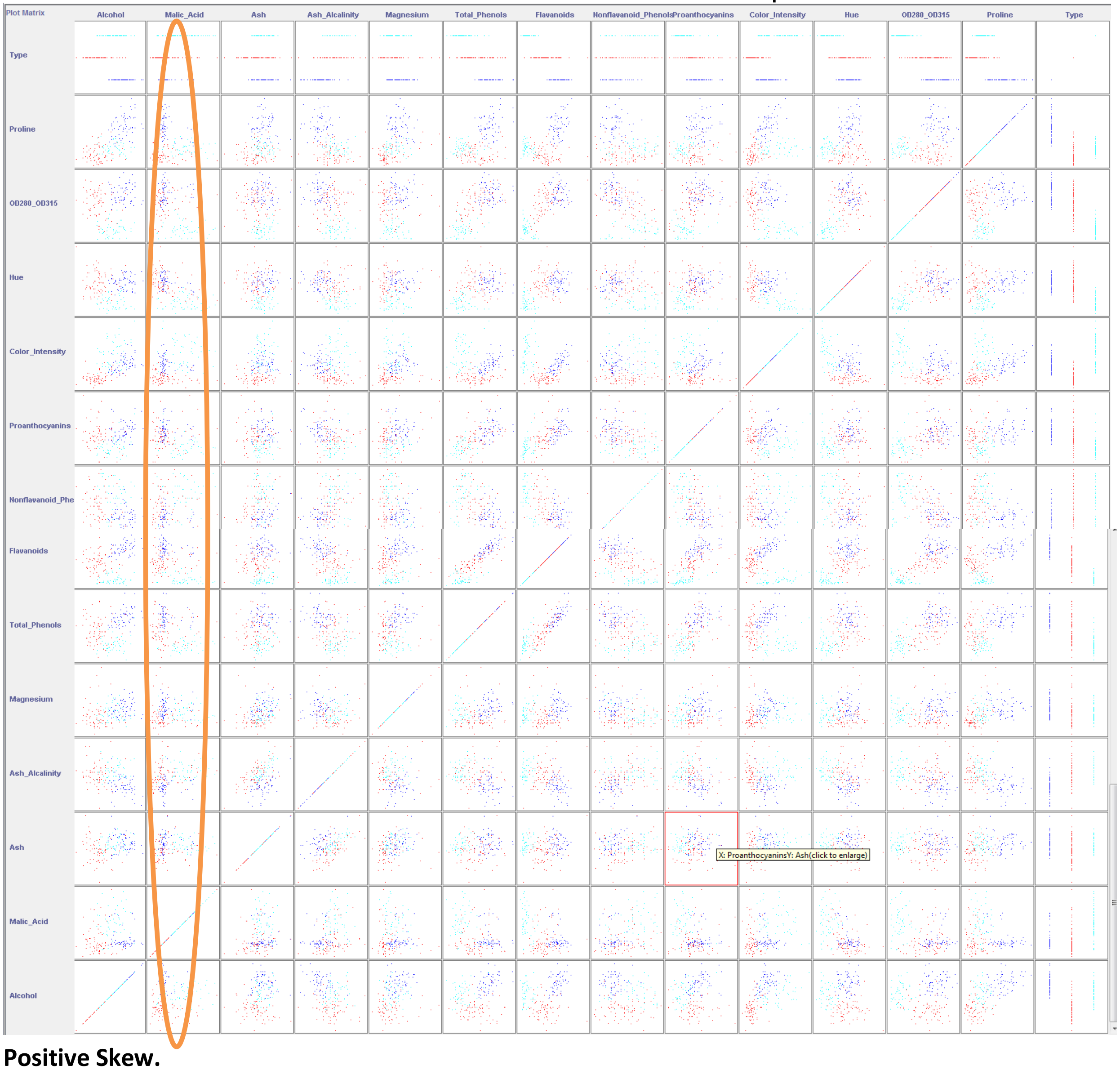
Listed below is the correlation with the variables:
Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.64	-0.19	0.22	-0.44	0.39	0.5	0.49	-0.31	0.33	0.32	0.24	0.31	1

These correlations have already been addressed.

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. The analysis above captures the type of data in the variables, the average, as well as variance.

This visual is a scatter matrix of all the variables compared to each other.



Positive Skew.

Methodology & Processes Implemented

Data Sets being Analyzed

Data set analyzed: Wine Dim1

Instances: 178

Attributes: 5

Total_Phenols
Flavanoids
Hue
OD280_OD315
Type

	Class		
Attribute	A	B	C
	(0.33)	(0.4)	(0.27)
Total_Phenols			
Flavanoids			
Hue			
OD280_OD315			

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.29	-0.34	0.13	-0.32	0.21	1	0.86	-0.45	0.61	-0.06	0.43	0.7	0.5

Total Phenols is strongly correlated with Flavanoids, Proanthocyanins, and OD280_OD315.

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.24	-0.41	0.12	-0.35	0.2	0.86	1	-0.54	0.65	-0.17	0.54	0.79	0.49

The three high correlations have been highlighted above. Notice how there is a negative correlation with Nonflavanoid_Phenols.

Hue:

Listed below is the correlation with the variables:

Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
-0.07	-0.56	-0.07	-0.27	0.06	0.43	0.54	-0.26	0.3	-0.52	1	0.57	0.24

OD280_OD315:

Listed below is the correlation with the variables:

Al=Alcohol M.A = Malic_Acid, Ash_A= Ash_Alcalinity, Mag= Magnesium, T.Ph = Total_Phenols, Flav= Flavanoids, Non_P= Nonflavanoid_Phenols, Pro= Proanthocyanins, Col_I= Color_Intensity, Od= OD280_OD315, Pr= Proline

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.07	-0.37	0	-0.28	0.07	0.7	0.79	-0.5	0.52	-0.43	0.57	1	0.31

The correlation matrix is:

Correlation matrix

1	0.86	0.43	0.7
0.86	1	0.54	0.79
0.43	0.54	1	0.57
0.7	0.79	0.57	1

Many of the variables are correlated with one another, and this may pose a problem called multi-collinearity.

Wine Dim2

Relation: Wine- Data of Second Dimension

Instances: 178

Attributes: 5

Alcohol

Magnesium

Color_Intensity

Proline

Correlation matrix

1	0.27	0.55	0.64
0.27	1	0.2	0.39
0.55	0.2	1	0.32
0.64	0.39	0.32	1

There is slight correlation amongst this data set, but not as bad as Wine Dim 1.

Relation: Wine- Data with all three dimension

Instances: 178

Attributes: 12

Alcohol

Malic_Acid

Ash

Ash_Alcalinity

Magnesium

Total_Phenols
Flavanoids
Color_Intensity
Hue
OD280_OD315
Proline
Type

Correlation matrix

```
  1      0.09   0.21  -0.31   0.27   0.29   0.24   0.55  -0.07   0.07   0.64
  0.09   1      0.16   0.29  -0.05  -0.34  -0.41   0.25  -0.56  -0.37  -0.19
  0.21   0.16   1      0.44   0.29   0.13   0.12   0.26  -0.07   0      0.22
 -0.31   0.29   0.44   1     -0.08  -0.32  -0.35   0.02  -0.27  -0.28  -0.44
  0.27  -0.05   0.29  -0.08   1      0.21   0.2    0.2    0.06   0.07   0.39
  0.29  -0.34   0.13  -0.32   0.21   1      0.86  -0.06   0.43   0.7    0.5
  0.24  -0.41   0.12  -0.35   0.2    0.86   1     -0.17   0.54   0.79   0.49
  0.55   0.25   0.26   0.02   0.2   -0.06  -0.17   1     -0.52  -0.43   0.32
 -0.07  -0.56  -0.07  -0.27   0.06   0.43   0.54  -0.52   1      0.57   0.24
  0.07  -0.37   0     -0.28   0.07   0.7    0.79  -0.43   0.57   1      0.31
  0.64  -0.19   0.22  -0.44   0.39   0.5    0.49   0.32   0.24   0.31   1
```

There are a few correlated variables; I might remove Proline from this data set to make it less correlated.

Relation: Wine-NoCorrelation-Data
Instances: 178
Attributes: 8

Malic_Acid
Ash
Ash_Alcalinity
Magnesium
Nonflavanoid_Phenols
Color_Intensity
Hue
Type

Correlation matrix

```
  1      0.16   0.29  -0.05   0.29   0.25  -0.56
  0.16   1      0.44   0.29   0.19   0.26  -0.07
  0.29   0.44   1     -0.08   0.36   0.02  -0.27
 -0.05   0.29  -0.08   1     -0.26   0.2    0.06
```

0.29	0.19	0.36	-0.26	1	0.14	-0.26
0.25	0.26	0.02	0.2	0.14	1	-0.52
-0.56	-0.07	-0.27	0.06	-0.26	-0.52	1

This data set is the cleanest for no multicollinearity.

Relation: Wine With Correlated Data
Instances: 178
Attributes: 6
Alcohol
Total_Phenols
Flavanoids
OD280_OD315
Proline
Type
Correlation matrix

1	0.29	0.24	0.07	0.64
0.29	1	0.86	0.7	0.5
0.24	0.86	1	0.79	0.49
0.07	0.7	0.79	1	0.31
0.64	0.5	0.49	0.31	1

This data set has a concentration of highly correlated variables.
The total data set has been analyzed from the initial variable analysis.

Cross Tabulation of Variables used						
	W_Dim_1	W_Dim_2	W_Dim_3	No_Corr	Corr	Total
Alcohol		■	🟦		🟩	🟪
Malic_Acid			🟦	🟥		🟪
Ash			🟦	🟥		🟪
Ash_Alcalinity			🟦	🟥		🟪
Magnesium		■	🟦	🟥		🟪
Total_Phenols	▨		🟦		🟩	🟪
Flavanoids	▨		🟦		🟩	🟪
Nonid_Phenols				🟥		🟪
Proanthocyanins						🟪
Color_Intensity		■	🟦	🟥		🟪
Hue	▨		🟦	🟥		🟪
OD280_OD315	▨		🟦		🟩	🟪
Proline		■	🟦		🟩	🟪

Proanthocyanins as a variable is only used in the total data set, its correlation matrix is shown below:

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.14	-0.22	0.01	-0.2	0.24	0.61	0.65	-0.37	1	-0.03	0.3	0.52	0.33

It has a mild correlation, but I wonder if this variable is necessary for the analysis.

Nonflavanoid_Phenols is only used in the non-correlated data set as well as the total.

Al	M.A	Ash	Ash_A	Mag	T.Phe	Flav	Non_P	Pro	Col_I	Hue	Od	Pr
0.16	0.29	0.19	0.36	-0.26	-0.45	-0.54	1	-0.37	0.14	-0.26	-0.5	-0.31

This variable only has slight inverse relationships with the other data. I will be watching this variable to analyze whether it is contributing to the overall EDA.

Explanation & Description of Algorithms

The 7 algorithms used on the 6 different data sets are the following:

1. Naïve Bayes
2. J48
3. IBK
4. Artificial Neural Network
5. Logistic Regression
6. Simple Regression
7. Linear Regression

In this section, I will describe each algorithm listed above and its approach.

Naïve Bayes: This algorithm is a classifier that utilizes probability based on an assumption of orthogonal variables. In other words, if the data suffers from multicollinearity this algorithm will not perform well. Naïve Bayes works well in a supervised learning environment, and utilizes maximum likelihood as the method of association. While Naïve Bayes appears to be over simplistic, it has a solid track record of creating solid models. Preliminarily, I see this model having an issue with the correlated data sets, but doing well with the overall data set as well as the uncorrelated data set.

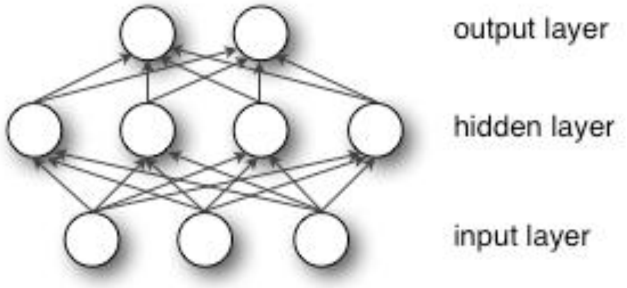
J48: This is a top-down approach that separates the example data into subsets (decision tree). J48 is a powerful tool for showing different patterns in data that might not have been understood from a traditional analysis. One caveat of decision tree algorithms like J48 is the pruning function, which serves to simplify the overall tree. One advantage of pruning is the indirect benefit of inhibiting over fitting. This is an attempt to strike the balance between accuracy and specificity. The pruning methods can best be described by Illinois.edu

> “J48 employs two pruning methods. The first is known as subtree replacement. This means that nodes in a decision tree may be replaced with a leaf -- basically reducing the number of tests along a certain path. This process starts from the leaves of the fully formed tree, and works backwards toward the root. The second type of pruning used in J48 is termed subtree raising. In this case, a node may be moved upwards towards the root of the tree, replacing other nodes along the way. Subtree raising often has a negligible effect on decision tree models. There is often no clear way to predict the utility of the option, though it may be advisable to try turning it off if the induction process is taking a long time. This is due to the fact that subtree raising can be somewhat computationally complex.”

As I fit this model, I will look to utilze the pruning options to create an iteration that maximizes the model, while bearing in mind over fitting.

IBK: This algorithm utilizes K-nearest neighbor classification. An object or data point is classified in comparison to its nearest points, neighbors, and the calculations are conducted when all objects have been classified. This is why this is called a ‘lazy’ approach. K is often assigned by the user and utilizes Euclidean distance. The drawback with this algorithm is the distribution of the data. From the initial analysis, there are many variables in this data set that have negative or positive skews associated with their distributions. Preliminarily, I see this being a major issue with the overall efficacy of this algorithm and the data sets. Perhaps, I will look at transforming the variables such that they follow a more normal distribution.

Artificial Neural Network (ANN): Logistic Regression is used as a non-linear transformer in the MLP process. A ANN mimics an actual physiological neural network in the learning process. Computationally, the ANN is heavy. It considers all the different options and associates different weights to specific nodes such that the best pathway is chosen. The goal of using logistic regression is to linearly separate data that initially is not linear. Shown below is an example of a Neural Network that has one hidden layer:



Logistic Regression: This algorithm utilizes maximum likelihood. Logistic Regression is used as a non-linear transformer in the MLP process. The goal of using logistic regression is to linearly analyze data that initially is not linear. This process is done through maximum likelihood estimation. This approach is specifically helpful for binary outcome variables.

Simple Logistic: This approach is very similar to Logistic Regression, but with the caveat that regression functions are used as the base for learning and fitting the model. I would expect this model to do well with the correlated data set, given that linearity has been observed and commented.

JRIP: This algorithm is known as a bottom-up process that analyzes all the examples. It is a rule based algorithm similar to J48, with the exception of how the branches are optimized. Pentaho.com best illustrates the building process described below:

> “Initialize RS = {}, and for each class from the less prevalent one to the more frequent one, DO:
>
> 1. Building stage:
> Repeat 1.1 and 1.2 until the descrition length (DL) of the ruleset and examples is 64 bits greater than the smallest DL met so far, or there are no positive examples, or the error rate >= 50%.
>
> 1.1. Grow phase:
> Grow one rule by greedily adding antecedents (or conditions) to the rule until the rule is perfect (i.e. 100% accurate). The procedure tries every

possible value of each attribute and selects the condition with highest information gain: p(log(p/t)-log(P/T)).

1.2. Prune phase:

Incrementally prune each rule and allow the pruning of any final sequences of the antecedents;The pruning metric is (p-n)/(p+n) – but it's actually 2p/(p+n) -1, so in this implementation we simply use p/(p+n) (actually (p+1)/(p+n+2), thus if p+n is 0, it's 0.5).

2. Optimization stage:

after generating the initial ruleset {Ri}, generate and prune two variants of each rule Ri from randomized data using procedure 1.1 and 1.2. But one variant is generated from an empty rule while the other is generated by greedily adding antecedents to the original rule. Moreover, the pruning metric used here is (TP+TN)/(P+N).Then the smallest possible DL for each variant and the original rule is computed. The variant with the minimal DL is selected as the final representative of Ri in the ruleset.After all the rules in {Ri} have been examined and if there are still residual positives, more rules are generated based on the residual positives using Building Stage again.

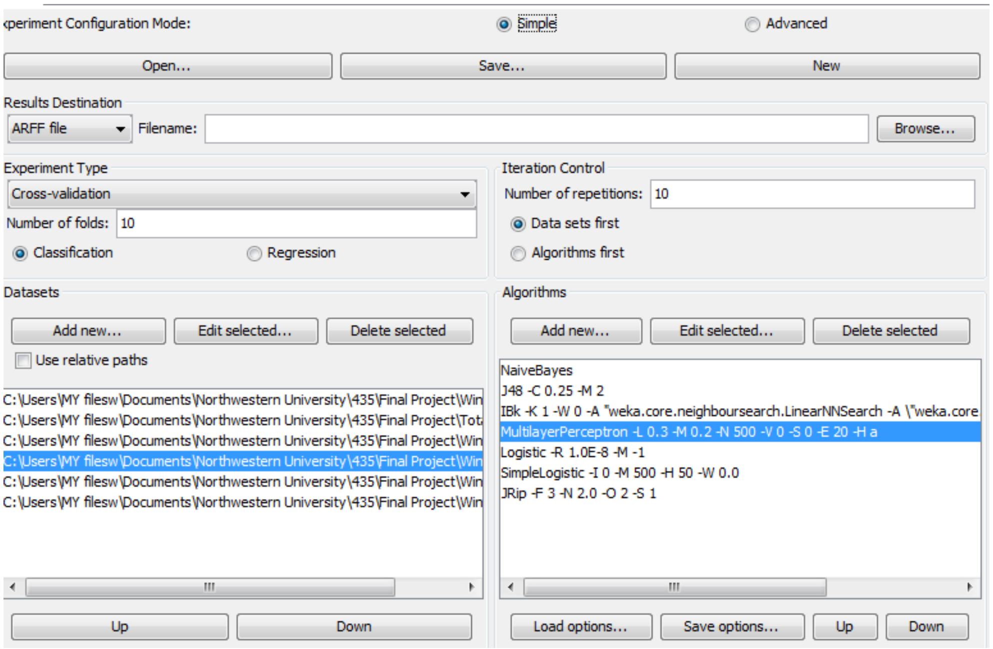
3. Delete the rules from the ruleset that would increase the DL of the whole ruleset if it were in it. and add resultant ruleset to RS.”

Described above are the seven different algorithms I will be using to analyze the six different data sets. The next step is to assign key metrics that will be used to assess and compare the different algorithms.

Key Performance Metrics

Each algorithm will produce a model, and within this model key metrics are calculated. The key performance indicators (KPIs) will be used to assess how well the algorithm performs compared to the other algorithm. Correctly classified percentage is important this explains the accuracy of the model. The mean square error highlights the precision of the model. Finally, the false negatives are normally an important indicator, but given that this data is not medical in nature it does not carry as much weight.

Loading the Data into Weka Experiment: The six datasets and seven algorithms were loaded into the Weka Experiment Environment.


xperiment Configuration Mode: ● Simple ○ Advanced

Results Destination: ARFF file; Filename:

Experiment Type: Cross-validation
Number of folds: 10
● Classification ○ Regression

Iteration Control
Number of repetitions: 10
● Data sets first
○ Algorithms first

Datasets
□ Use relative paths

```
C:\Users\MY filesw\Documents\Northwestern University\435\Final Project\Win
C:\Users\MY filesw\Documents\Northwestern University\435\Final Project\Tot
C:\Users\MY filesw\Documents\Northwestern University\435\Final Project\Win
C:\Users\MY filesw\Documents\Northwestern University\435\Final Project\Win
C:\Users\MY filesw\Documents\Northwestern University\435\Final Project\Win
C:\Users\MY filesw\Documents\Northwestern University\435\Final Project\Win
```

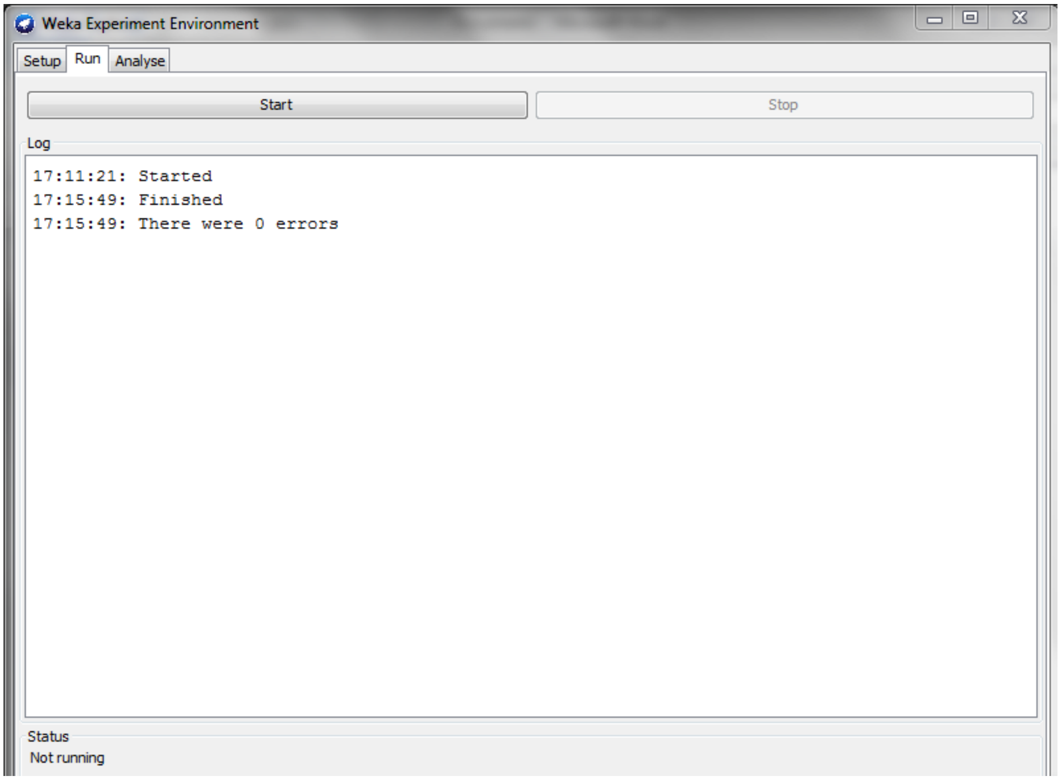
Algorithms

```
NaiveBayes
J48 -C 0.25 -M 2
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Logistic -R 1.0E-8 -M -1
SimpleLogistic -I 0 -M 500 -H 50 -W 0.0
JRip -F 3 -N 2.0 -O 2 -S 1
```

Notice how for this first analysis cross-validation is being used with 10 folds.

There are ten repetitions for iteration control.

The datasets and algorithms ran with 0 errors.



```
17:11:21: Started
17:15:49: Finished
17:15:49: There were 0 errors
```

Not running

Output:

Analysing: Percent_correct

Datasets: 6

Resultsets: 7

Confidence: 0.05 (two tailed)

Sorted by: -

Date: 8/20/13 6:01 PM

Analysis of Initial Benchmark

The initial output generated from the above parameters will serve as the benchmark to improve the algorithms from the data and algorithmic attribute side. This model output below will serve as the benchmark to improve future models. After the data and algorithms are modified, the comparative analysis will take place. The top three algorithms will be classified with Yellow (1), Green (2), and Blue (3).

```
Dataset                                   (1) bayes.Na | (2) trees (3) lazy. (4) funct (5) funct (6) funct (7) rules
----------------------------------------------------------------------------------------------------
               'Wine With Correlated Dat(100)   94.42 |   88.60 *   91.10     93.76     92.51     93.70     90.76
                'Wine with all data inclu(100)   97.46 |   93.20     95.12     98.02     97.23     97.92     92.97 *
                'Wine- Data with all thre(100)   97.41 |   93.14     96.13     98.99     96.63     97.87     92.47 *
                  'Wine- Variables for firs(100)   86.37 |   84.84     80.00 *   89.87     83.89     83.44     83.71
              'Wine- Data of Second Dim(100)   91.89 |   87.25     91.61     91.12     90.71     90.94     85.41 *
              Wine-NoCorrelation-Data  (100)   92.29 |   89.55     86.30 *   90.60     89.27     89.15     86.01
----------------------------------------------------------------------------------------------------
                              (v/ /*) |   (0/5/1)   (0/4/2)   (0/6/0)   (0/6/0)   (0/6/0)   (0/3/3)
```

Key:

(1) bayes.NaiveBayes '' 5995231201785697655

(2) trees.J48 '-C 0.25 -M 2' -217733168393644444

(3) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.EuclideanDistance -R first-last\\\"\"' -3080186098777067172

(4) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -5990607817048210779

(5) functions.Logistic '-R 1.0E-8 -M -1' 3932117032546553727

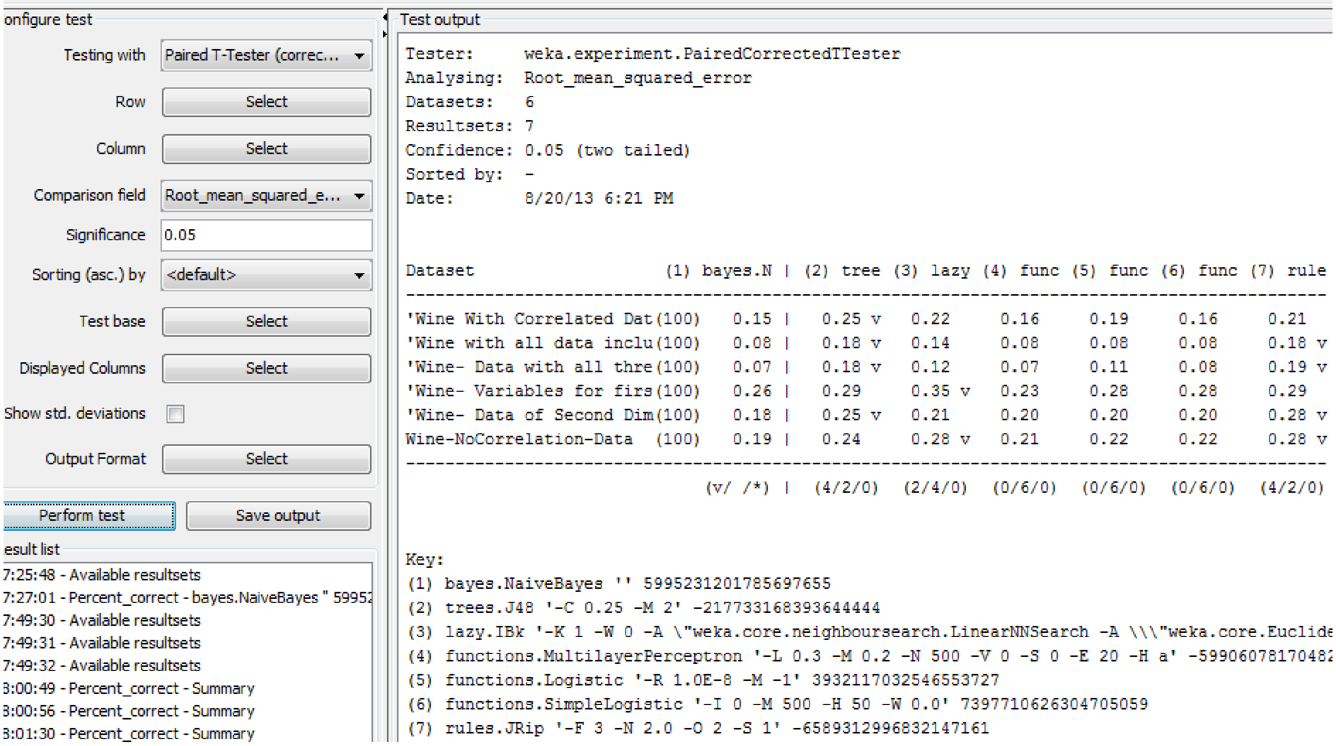
(6) functions.SimpleLogistic '-I 0 -M 500 -H 50 -W 0.0' 7397710626304705059

(7) rules.JRip '-F 3 -N 2.0 -O 2 -S 1' -6589312996832147161

IBK and Naïve Bayes both had 3 of the top correctly classifying percentages for the data sets. JRIP was not in any of the three top spots for any data set. This classifier is rule based, which leads me to believe other rule based algorithms will have similar output.

Root mean square output:

Naïve Bayes consistently has the lowest root mean square error, and the ANN consistently ranks second. The more precise a model the lower the root mean square error.



Testing with: Paired T-Tester (correc…

Comparison field: Root_mean_squared_e…

Significance: 0.05

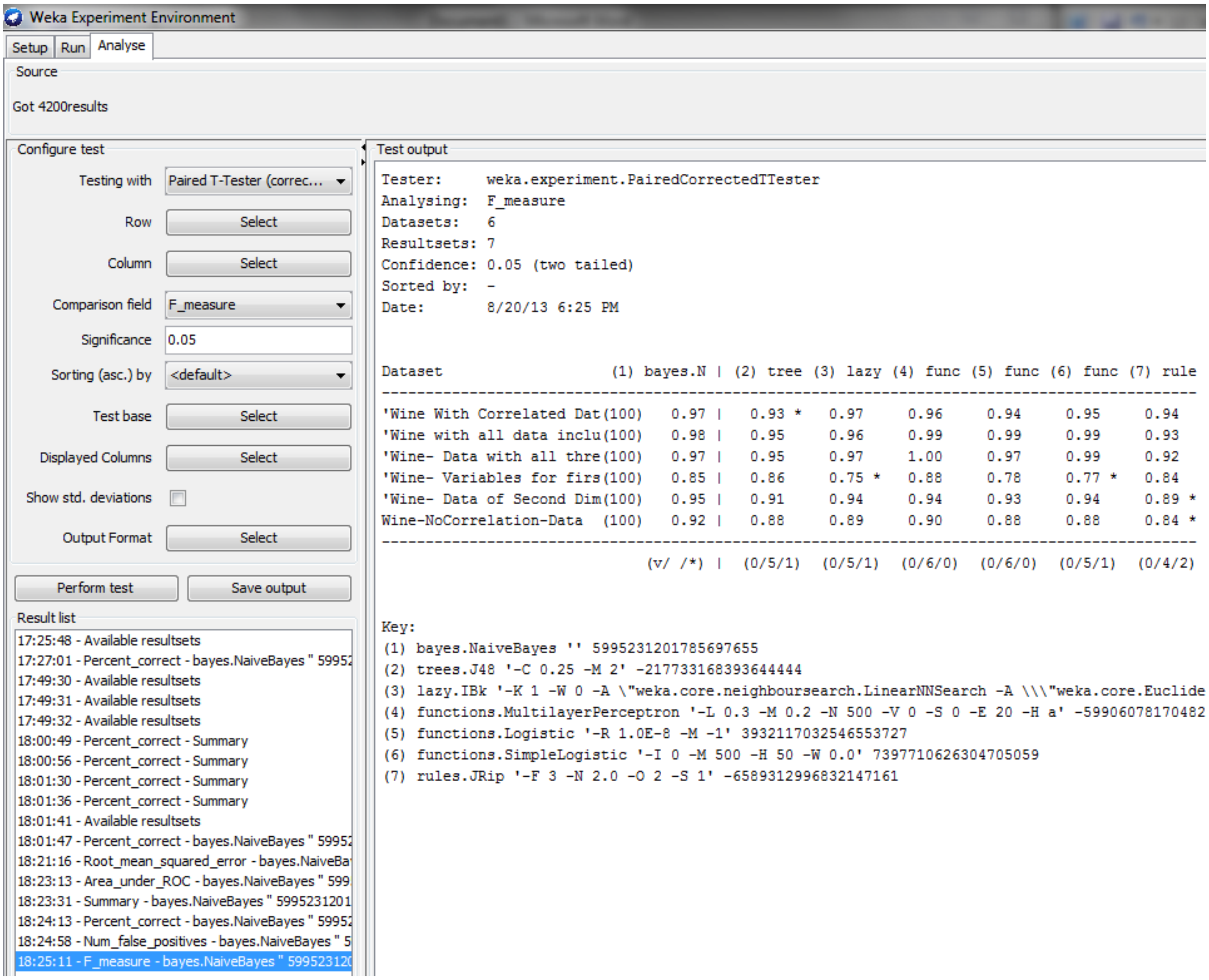
Sorting (asc.) by: <default>

Show std. deviations ☐

```
7:25:48 - Available resultsets
7:27:01 - Percent_correct - bayes.NaiveBayes " 59952
7:49:30 - Available resultsets
7:49:31 - Available resultsets
7:49:32 - Available resultsets
3:00:49 - Percent_correct - Summary
3:00:56 - Percent_correct - Summary
3:01:30 - Percent_correct - Summary
```

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Root_mean_squared_error
Datasets:   6
Resultsets: 7
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       8/20/13 6:21 PM


Dataset                   (1) bayes.N | (2) tree (3) lazy (4) func (5) func (6) func (7) rule
---------------------------------------------------------------------------------------------
'Wine With Correlated Dat(100)   0.15 |   0.25 v   0.22     0.16     0.19     0.16     0.21  
'Wine with all data inclu(100)   0.08 |   0.18 v   0.14     0.08     0.08     0.08     0.18 v
'Wine- Data with all thre(100)   0.07 |   0.18 v   0.12     0.07     0.11     0.08     0.19 v
'Wine- Variables for firs(100)   0.26 |   0.29     0.35 v   0.23     0.28     0.28     0.29  
'Wine- Data of Second Dim(100)   0.18 |   0.25 v   0.21     0.20     0.20     0.20     0.28 v
Wine-NoCorrelation-Data  (100)   0.19 |   0.24     0.28 v   0.21     0.22     0.22     0.28 v
---------------------------------------------------------------------------------------------
                              (v/ /*) |  (4/2/0)  (2/4/0)  (0/6/0)  (0/6/0)  (0/6/0)  (4/2/0)


Key:
(1) bayes.NaiveBayes '' 5995231201785697655
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.Euclide
(4) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -59906078170482
(5) functions.Logistic '-R 1.0E-8 -M -1' 3932117032546553727
(6) functions.SimpleLogistic '-I 0 -M 500 -H 50 -W 0.0' 7397710626304705059
(7) rules.JRip '-F 3 -N 2.0 -O 2 -S 1' -6589312996832147161
```

The F-test, which is similar to the root mean square error, assesses the overall model fit. The greater the value per algorithm the better the model.



Got 4200results

Testing with: Paired T-Tester (correc...
Comparison field: F_measure
Significance: 0.05
Sorting (asc.) by: <default>
Show std. deviations: □

```
17:25:48 - Available resultsets
17:27:01 - Percent_correct - bayes.NaiveBayes '' 59952
17:49:30 - Available resultsets
17:49:31 - Available resultsets
17:49:32 - Available resultsets
18:00:49 - Percent_correct - Summary
18:00:56 - Percent_correct - Summary
18:01:30 - Percent_correct - Summary
18:01:36 - Percent_correct - Summary
18:01:41 - Available resultsets
18:01:47 - Percent_correct - bayes.NaiveBayes '' 59952
18:21:16 - Root_mean_squared_error - bayes.NaiveBa
18:23:13 - Area_under_ROC - bayes.NaiveBayes '' 599
18:23:31 - Summary - bayes.NaiveBayes '' 5995231201
18:24:13 - Percent_correct - bayes.NaiveBayes '' 59952
18:24:58 - Num_false_positives - bayes.NaiveBayes '' 5
18:25:11 - F_measure - bayes.NaiveBayes '' 59952312
```

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  F_measure
Datasets:   6
Resultsets: 7
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       8/20/13 6:25 PM


Dataset                   (1) bayes.N | (2) tree (3) lazy (4) func (5) func (6) func (7) rule
---------------------------------------------------------------------------------------------
'Wine With Correlated Dat(100)   0.97 |   0.93 *   0.97     0.96     0.94     0.95     0.94
'Wine with all data inclu(100)   0.98 |   0.95     0.96     0.99     0.99     0.99     0.93
'Wine- Data with all thre(100)   0.97 |   0.95     0.97     1.00     0.97     0.99     0.92
'Wine- Variables for firs(100)   0.85 |   0.86     0.75 *   0.88     0.78     0.77 *   0.84
'Wine- Data of Second Dim(100)   0.95 |   0.91     0.94     0.94     0.93     0.94     0.89 *
Wine-NoCorrelation-Data  (100)   0.92 |   0.88     0.89     0.90     0.88     0.88     0.84 *
---------------------------------------------------------------------------------------------
                              (v/ /*) |  (0/5/1)  (0/5/1)  (0/6/0)  (0/6/0)  (0/5/1)  (0/4/2)


Key:
(1) bayes.NaiveBayes '' 5995231201785697655
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.Euclide
(4) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -59906078170482
(5) functions.Logistic '-R 1.0E-8 -M -1' 3932117032546553727
(6) functions.SimpleLogistic '-I 0 -M 500 -H 50 -W 0.0' 7397710626304705059
(7) rules.JRip '-F 3 -N 2.0 -O 2 -S 1' -6589312996832147161
```

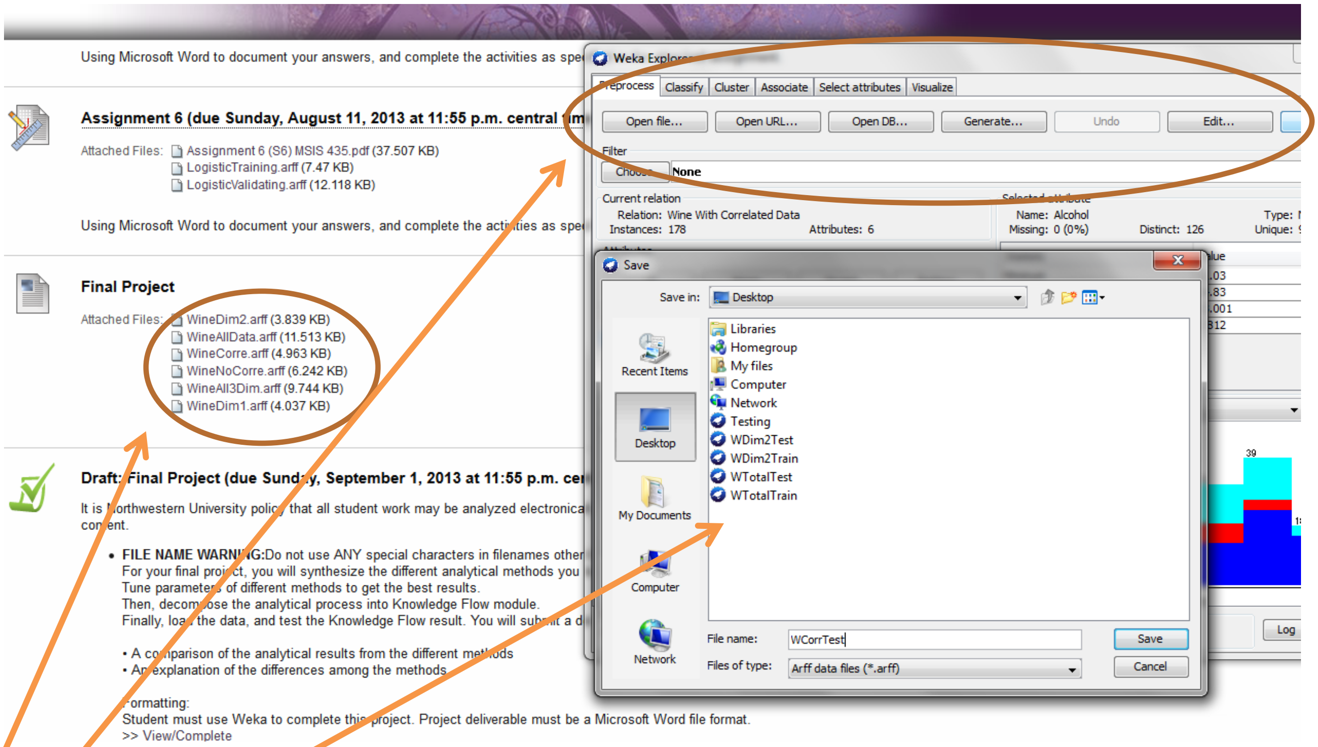
The F-test follows similar suit in that Naïve Bayes, ANN, and Logistic Regression are consistently the top models. Moving forward, I will assess both the data and algorithm parameters to maximize the three KPIs listed above for each algorithm. After the iterations have been carried out, I will partition the data such that there is a learning set and a testing set. This will provide another metric to critique the models. Algorithms that perform well on both data sets demonstrate a robust nature.

Creating testing and training data sets:

Utilizing the Data Edit function in Weka Experiment, I am going to create a training data set that represents 33% (58 instances) of the data, and a testing data set that represents 67% (120 instances) of the data for each data set. I will then test the algorithms on both sets and compare the results.

Below are the screen shots for performing this task:

Step 1: Download each file twice, saving one file as training and the other as testing.



Using Microsoft Word to document your answers, and complete the activities as spe

Assignment 6 (due Sunday, August 11, 2013 at 11:55 p.m. central tim

Attached Files: Assignment 6 (S6) MSIS 435.pdf (37.507 KB)
LogisticTraining.arff (7.47 KB)
LogisticValidating.arff (12.118 KB)

Using Microsoft Word to document your answers, and complete the activities as spe

Final Project

Attached Files: WineDim2.arff (3.839 KB)
WineAllData.arff (11.513 KB)
WineCorre.arff (4.963 KB)
WineNoCorre.arff (6.242 KB)
WineAll3Dim.arff (9.744 KB)
WineDim1.arff (4.037 KB)

Draft: Final Project (due Sunday, September 1, 2013 at 11:55 p.m. cen

It is Northwestern University policy that all student work may be analyzed electronica content.

- **FILE NAME WARNING:**Do not use ANY special characters in filenames other
For your final project, you will synthesize the different analytical methods you
Tune parameters of different methods to get the best results.
Then, decompose the analytical process into Knowledge Flow module.
Finally, load the data, and test the Knowledge Flow result. You will submit a d

• A comparison of the analytical results from the different methods
• An explanation of the differences among the methods

Formatting:
Student must use Weka to complete this project. Project deliverable must be a Microsoft Word file format.
>> View/Complete

Current relation
Relation: Wine With Correlated Data
Instances: 178
Attributes: 6

Selected attribute
Name: Alcohol
Missing: 0 (0%)
Distinct: 126

Save in: Desktop

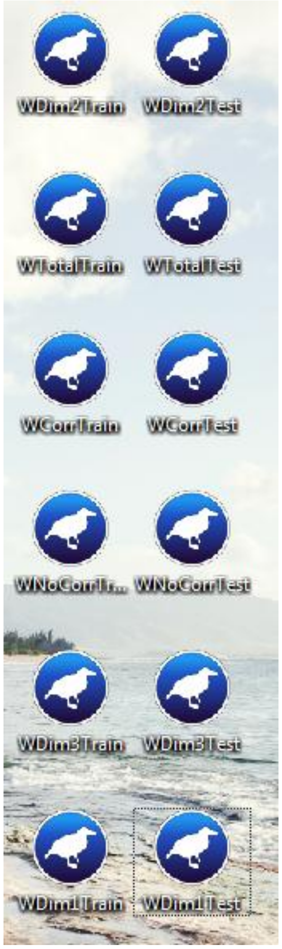
File name: WCorrTest
Files of type: Arff data files (*.arff)

From this screen shot, I am downloading the Arff files from the Blackoard website.

Each data set is then opened.

It is saved twice, the first it to be modified by 33%, the second is the remaining 67%.

Looking below, one can see that all 6 data sets have been saved as testing and training sets.



The data sets will now be opened in Weka Experiment to refine the amount of instances.

Within the bounds for this experiment, it has been communicated that this type of analysis is not needed. In my opinion, this is a very realistic experience that happens quite frequently in the working environment. Communication with management is key for staying on track and not wasting unnecessary time on efforts that do not add value to the bottom line. In conclusion, the last two pages of documentation are not within the scope of this particular project, and I found it necessary to document the process for project flow efficacy.

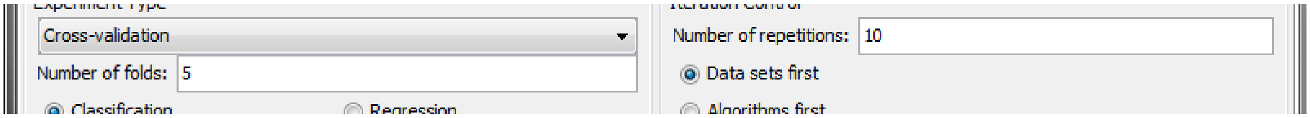
Preprocessing & Attribute Selection

Data is often modified to better suit particular algorithms. Throughout this section, I will modify the data in an effort to increase the three KPI's for each algorithm.

Naïve Bayes: Benchmark Model

```
Dataset                                Naïve Bayes
 'Wine With Correlated Dat(100)          94.42 |
 'Wine with all data inclu(100)          97.46 |
 'Wine- Data with all thre(100)          97.41 |
 'Wine- Variables for firs(100)          86.37 |
 'Wine- Data of Second Dim(100)          91.89 |
 Wine-NoCorrelation-Data  (100)          92.29 |
```

The numbers to the left represent the correctly classified instances with ten folds and ten repetitions.



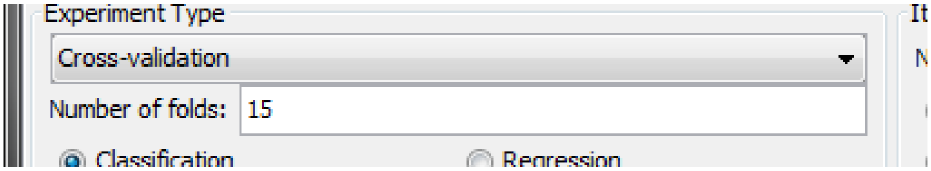
Experiment Type
Cross-validation
Number of folds: 5
● Classification ○ Regression

Iteration Control
Number of repetitions: 10
● Data sets first
○ Algorithms first

```
Dataset                  (1) bayes.Naive

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

'Wine with all data inclu (50)    97.53 |

'Wine- Data with all thre (50)    97.47 |

'Wine With Correlated Dat (50)   100.00 |

'Wine- Variables for firs (50)    85.80 |

'Wine- Data of Second Dim (50)    91.91 |

Wine-NoCorrelation-Data   (50)    91.97 |
```

I changed the folds from ten to five. The folds are an option to cross-validate the model. The more folds the more the model is going to learn. Overfitting can result from having too many folds. The inverse of more folds is less folds. The less the folds the greater the chance that the model is not over-fit. With less folds the correctly classified increased, except for “'Wine- Variables for firs (50)   85.80”.

I then increased the folds from ten to 15. This also produced better output in all categories.



Experiment Type

Cross-validation

Number of folds: 15

● Classification ○ Regression

.

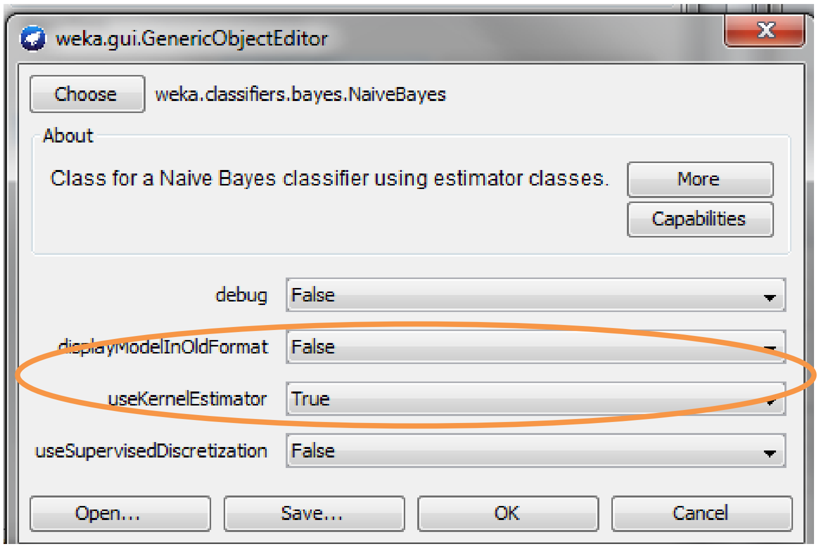
```
Dataset                   (1) bayes.Naive

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

'Wine with all data inclu(150)    97.30 |

'Wine- Data with all thre(150)    97.25 |

'Wine With Correlated Dat(150)   100.00 |

'Wine- Variables for firs(150)    86.73 |

'Wine- Data of Second Dim(150)    92.13 |

Wine-NoCorrelation-Data  (150)    91.99 |
```

Naive Bayes consistently classified the three highlighted data sets to the left exceptionally well. Moving forward five folds produces the best classification results.

```
Dataset                   (1) bayes.Nai

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

'Wine with all data inclu (50)   0.10 |

'Wine- Data with all thre (50)   0.10 |

'Wine With Correlated Dat (50)   0.00 |

'Wine- Variables for firs (50)   0.27 |

'Wine- Data of Second Dim (50)   0.19 |

Wine-NoCorrelation-Data   (50)   0.20 |
```

This output is the Root Mean squared for five folds. This output also reflects multiple models where the goodness of fit is acceptable.

Now that I have settled on five folds, I will move on to modifying the Naive Bayes algorithmic parameters.



weka.classifiers.bayes.NaiveBayes

Class for a Naive Bayes classifier using estimator classes.

debug False

displayModelInOldFormat False

useKernelEstimator True

useSupervisedDiscretization False

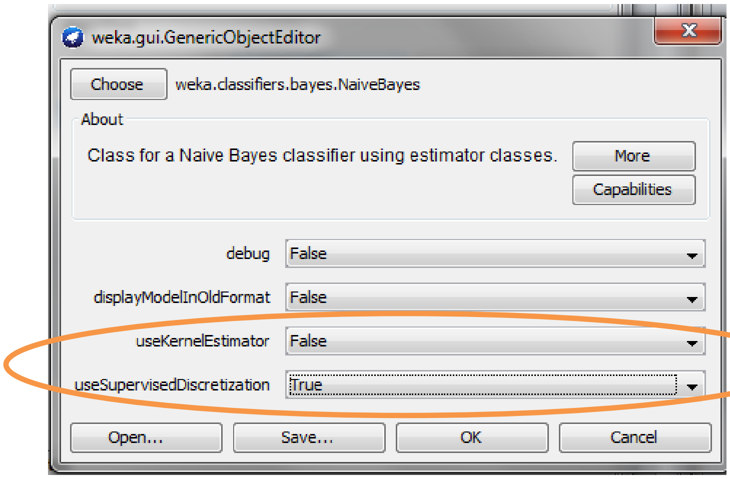
I am going to focus on the two highlighted options for Naive Bayes. The goal by modifying these two different options it to improve the model fit. I am looking to improve the output listed below.

```
Dataset                    (1) bayes.Naive

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

'Wine with all data inclu (50)    97.53 |

'Wine- Data with all thre (50)    97.47 |

'Wine With Correlated Dat (50)   100.00 |

'Wine- Variables for firs (50)    85.80 |

'Wine- Data of Second Dim (50)    91.91 |

Wine-NoCorrelation-Data   (50)    91.97 |
```

This is the new benchmark.

```
Dataset                    (1) bayes.Naive

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

'Wine with all data inclu (50)    97.41 |

'Wine- Data with all thre (50)    97.86 |

'Wine With Correlated Dat (50)   100.00 |

'Wine- Variables for firs (50)    85.40 |

'Wine- Data of Second Dim (50)    92.14 |

Wine-NoCorrelation-Data   (50)    93.42 |
```

The Kernel Estimator increased "Wine- Data with all three" and slightly lowered 'all data'. I am going to stick with the Kernel Estimator. The increase is worth the slight decrease.



weka.classifiers.bayes.NaiveBayes

Class for a Naive Bayes classifier using estimator classes.

debug: False
displayModelInOldFormat: False
useKernelEstimator: False
useSupervisedDiscretization: True

I am now going to run the same experiment with useSupervisedDiscretization.

```
Dataset                    (1) bayes.Naive

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

'Wine with all data inclu (50)    98.37 |

'Wine- Data with all thre (50)    98.37 |

'Wine With Correlated Dat (50)   100.00 |

'Wine- Variables for firs (50)    83.89 |

'Wine- Data of Second Dim (50)    89.32 |

Wine-NoCorrelation-Data   (50)    93.65 |------
```

The top 3 datasets have continued to get better. I need to look at my Root Mean Square Error as well as the F-Statistic.

```
Dataset                    (1) bayes.Nai

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

'Wine with all data inclu (50)   0.07 |

'Wine- Data with all thre (50)   0.07 |

'Wine With Correlated Dat (50)   0.03 |

'Wine- Variables for firs (50)   0.28 |

'Wine- Data of Second Dim (50)   0.23 |

Wine-NoCorrelation-Data   (50)   0.18 |
```

Not surprisingly, the root mean square error has lowered which is a positive sign that the model is a better fit.

```
Dataset                    (1) bayes.Nai

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

'Wine with all data inclu (50)   0.99 |

'Wine- Data with all thre (50)   0.99 |

'Wine With Correlated Dat (50)   1.00 |

'Wine- Variables for firs (50)   0.85 |

'Wine- Data of Second Dim (50)   0.93 |

Wine-NoCorrelation-Data   (50)   0.94 |
```

The F-measure also reflects a good fit for the first three data sets.

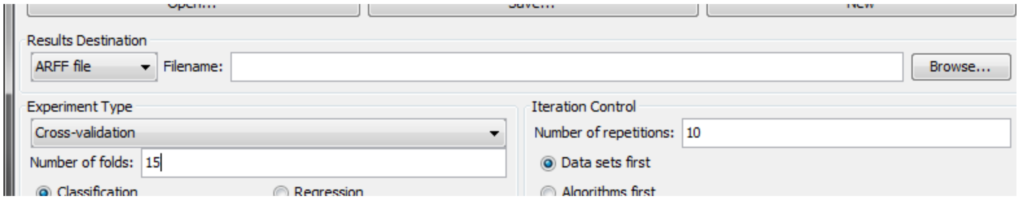
Now that the top three data sets have been successfully iterated, addressing the three other datasets is necessary. I do not have a need to refine these three datasets because from the initial analysis there are different algorithms that better classify these datasets. I do not want to spend time refining the overall algorithm for certain datasets where the algorithm is not suited well. To recap, I will be using Naive Bayes, five folds, and Supervised Discretization.

J48

J48's initial benchmark is displayed below:

```
Wine-NoCorrelation-Data  (100)  |   89.55

'Wine- Data of Second Dim(100)   87.25

'Wine- Variables for firs(100)   |   84.84

'Wine- Data with all thre(100)   93.14

'Wine with all data inclu(100)   93.20

'Wine With Correlated Dat(100)   88.60 *
```

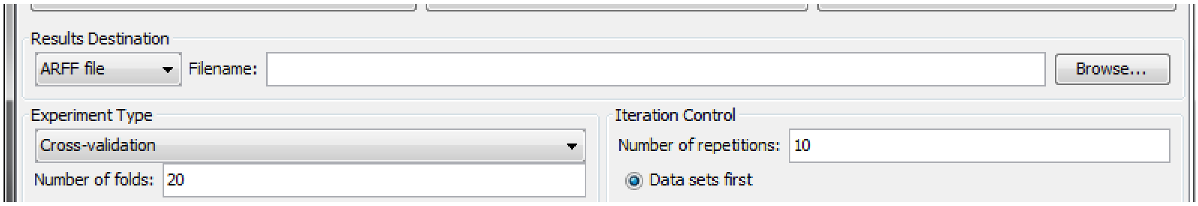
The highlighted data reflects the data sets that are still open for an algorithm. Naive Bayes has a classification rate of 98.37% and 100% for the non-highlighted data sets. I will focus pre-processing and attribute selection on the remaining data sets. I will start with analyzing the folds, starting with 5, 15, and 20.



Results Destination
ARFF file ▾ Filename:

Experiment Type
Cross-validation ▾
Number of folds: 15
● Classification ○ Regression

Iteration Control
Number of repetitions: 10
● Data sets first
○ Algorithms first



Results Destination

ARFF file Filename: Browse...

Experiment Type

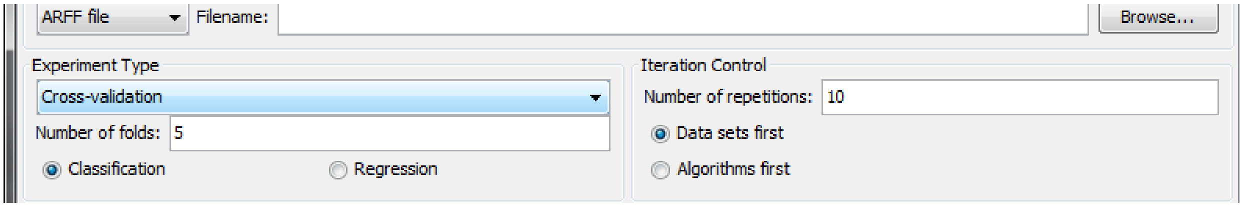
Cross-validation

Number of folds: 20

Iteration Control

Number of repetitions: 10

● Data sets first



ARFF file Filename: Browse...

Experiment Type

Cross-validation

Number of folds: 5

● Classification ○ Regression

Iteration Control

Number of repetitions: 10

● Data sets first

○ Algorithms first

Of the three fold options, I found that 15 folds had the most desirable output in regard to correctly classified instances.

'Wine with all data inclu(150)   93.21 |

'Wine- Data with all thre(150)   93.15 |

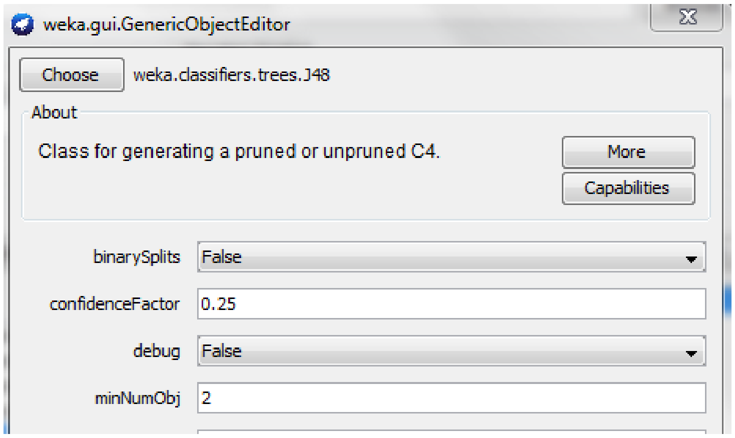
'Wine With Correlated Dat(150)   100.00 |

'Wine- Variables for firs(150)   85.35 |

'Wine- Data of Second Dim(150)   86.92 |

Wine-NoCorrelation-Data (150)   90.21 |

Dataset 'Wine-NoCorrelation' improved while the other two focus datasets declined slightly. While the goal is to focus on all three datasets at this point, the output did not improve with any of three for the fold options.



weka.gui.GenericObjectEditor

Choose weka.classifiers.trees.J48

About

Class for generating a pruned or unpruned C4.

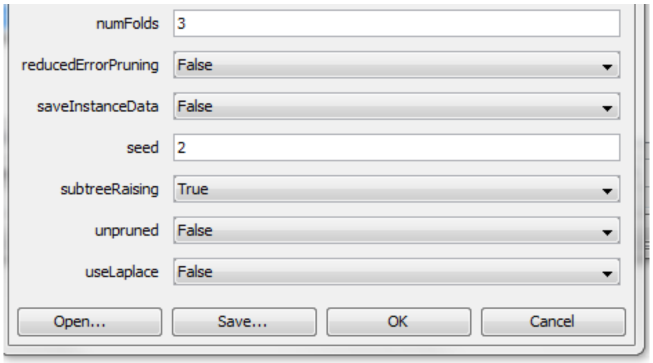
binarySplits False

confidenceFactor 0.25

debug False

minNumObj 2

J48 has quite a few opportunities for iterations. The first four are listed to the left. I conducted multiple iterations for each option, but the model did not improve for these options.



numFolds	3
reducedErrorPruning	False
saveInstanceData	False
seed	2
subtreeRaising	True
unpruned	False
useLaplace	False

Listed to the left are the other options for customization. After modifiying all the options, I was not able to find a sequence that improved the overall accuracy. Moving forward I will use the base line settings with five folds.

```
'Wine with all data inclu(150)   0.16 |
'Wine- Data with all thre(150)   0.16 |
'Wine With Correlated Dat(150)   0.00 |
'Wine- Variables for firs(150)   0.28 |
'Wine- Data of Second Dim(150)   0.24 |
Wine-NoCorrelation-Data  (150)   0.21 |
```

The only dataset that I see J48 competing for in regard to overall classification is the non-correlated data set. The rootmean squared error to the left is reasonble, but not great.

```
'Wine with all data inclu(150)   0.95 |
'Wine- Data with all thre(150)   0.95 |
'Wine With Correlated Dat(150)   1.00 |
'Wine- Variables for firs(150)   0.87 |
'Wine- Data of Second Dim(150)   0.90 |
Wine-NoCorrelation-Data  (150)   0.90 |
```

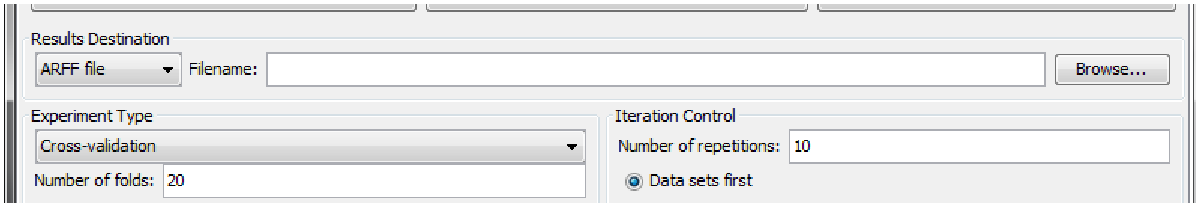
The F-test results to the left reinforce the idea that ‘Wine-NoCorrelation-Data’ is the best dataset for J48.

After conducting the iterations, I do not see this algorithm making it into the final rounds of evaluation. Even after the iteration, J48 did not classify ‘Wine-NoCorrelation-Data’ better than Naive Bayes.

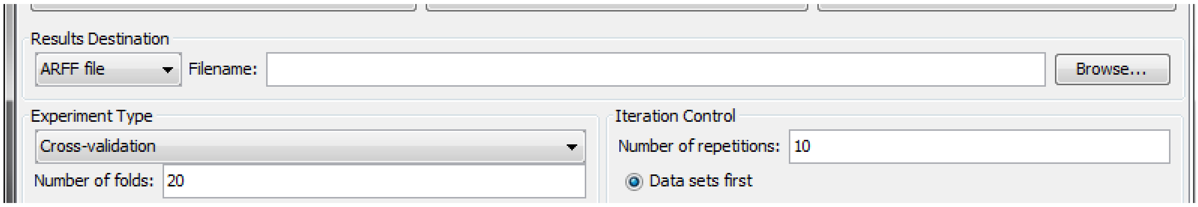
IBK

When refining a particular algorithm, it can be helpful to see where it lies in comparison to the other algorithms it is being assessed against. Listed below is the update algorithmic classification output.

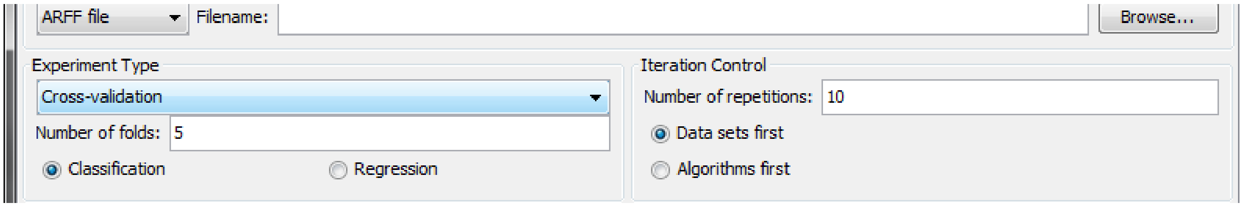
IBK may have an opportunity to take the lead with the ‘Wine-Second Dim’ dataset. I will start with analyzing the folds, starting with 5, 15, and 20.



Results Destination
ARFF file ▾ Filename: [Browse...]
Experiment Type
Cross-validation ▾
Number of folds: 20
Iteration Control
Number of repetitions: 10
● Data sets first



Results Destination
ARFF file ▾ Filename: [Browse...]
Experiment Type
Cross-validation ▾
Number of folds: 20
Iteration Control
Number of repetitions: 10
● Data sets first

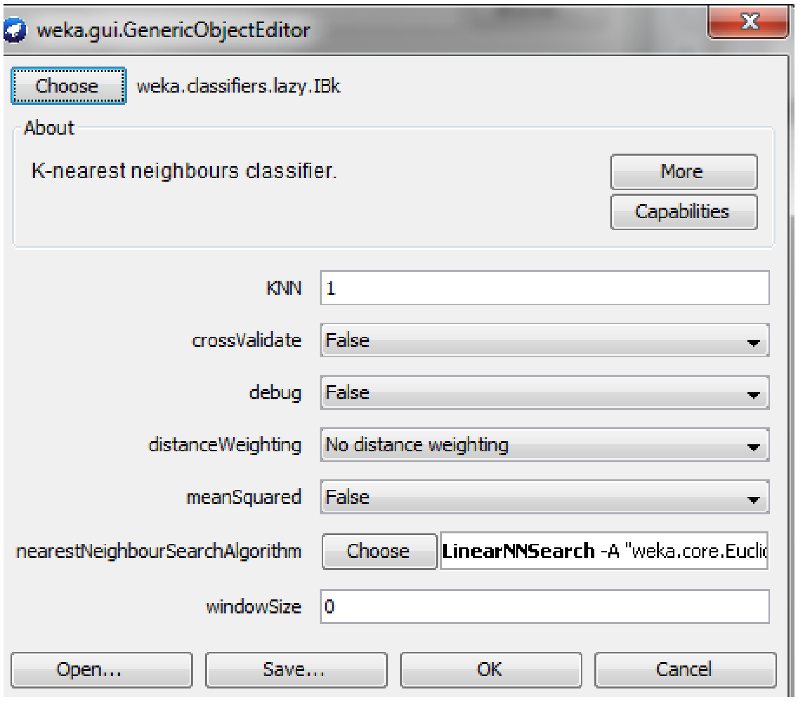


ARFF file ▾ Filename: [Browse...]
Experiment Type
Cross-validation ▾
Number of folds: 5
● Classification ○ Regression
Iteration Control
Number of repetitions: 10
● Data sets first
○ Algorithms first

At 20 folds the data set looks like the following:

'Wine with all data inclu(200)	95.01 |
'Wine- Data with all thre(200)	96.06 |
'Wine With Correlated Dat(200)	100.00 |
'Wine- Variables for firs(200)	80.03 |
'Wine- Data of Second Dim(200)	91.76 |
Wine-NoCorrelation-Data (200)	86.76 |

I have a hard time justifying 10 extra folds for such a small increase. Moving forward I will stick with the standard 10 folds for this algorithm based on the fact I could not justify increasing or decreasing folds to increase accuracy.


weka.gui.GenericObjectEditor

Choose weka.classifiers.lazy.IBk

About

K-nearest neighbours classifier.

KNN 1

crossValidate False

debug False

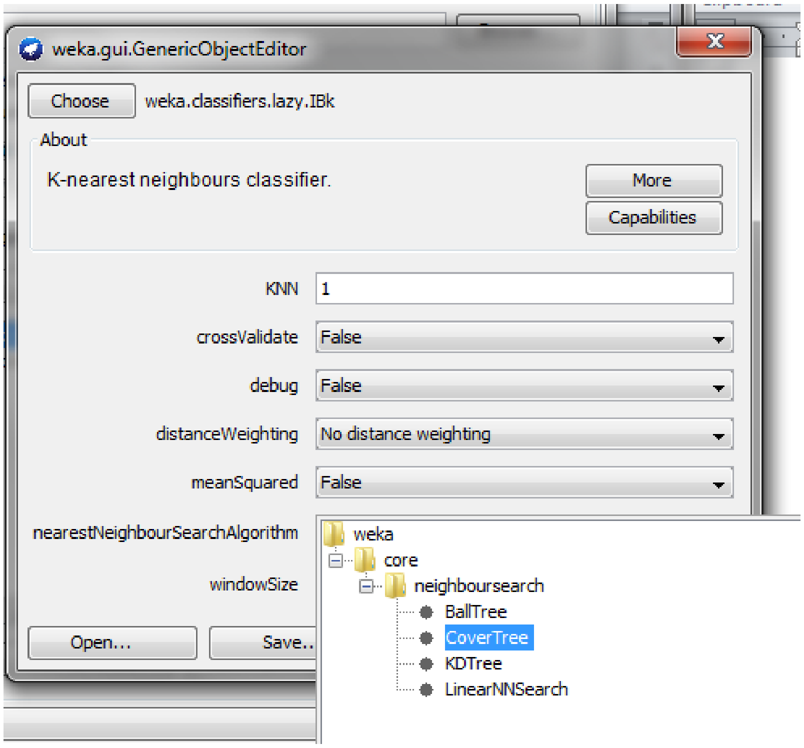
distanceWeighting No distance weighting

meanSquared False

nearestNeighbourSearchAlgorithm Choose **LinearNNSearch** -A "weka.core.Euclid

windowSize 0

IBK has the following options to improve the algorithm based on the data. I will work through each option and where there is opportunity I will document it. The end goal is to iterate the model such that it can increase its classification percentage.


weka.gui.GenericObjectEditor

Choose weka.classifiers.lazy.IBk

About

K-nearest neighbours classifier.

KNN 1

crossValidate False

debug False

distanceWeighting No distance weighting

meanSquared False

nearestNeighbourSearchAlgorithm

windowSize

Within the options, different algorithms could be used to assess the datasets. This is a new option outside of the 5 other iterative options in the options menu. After fitting each algorithm, I could not beat the original benchmark options. This is quite disappointing because time has been invested trying to modify the algorithm.

```
'Wine with all data inclu(100)    95.12 |
'Wine- Data with all thre(100)    96.13 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    80.00 |
'Wine- Data of Second Dim(100)    91.61 |
Wine-NoCorrelation-Data  (100)    86.30 |
```

As seen in the first iteration, the most value IBK offers to the overall datasets is its classification of the highlighted dataset to the left.

```
'Wine with all data inclu(100)   0.14 |
'Wine- Data with all thre(100)   0.12 |
'Wine With Correlated Dat(100)   0.03 |
'Wine- Variables for firs(100)   0.35 |
'Wine- Data of Second Dim(100)   0.21 |
Wine-NoCorrelation-Data  (100)   0.28 |
```

The root mean square error is not that impressive to lead me to the conclusion that this algorithm fits the data well.

```
'Wine with all data inclu(100)   0.96 |
'Wine- Data with all thre(100)   0.97 |
'Wine With Correlated Dat(100)   1.00 |
'Wine- Variables for firs(100)   0.75 |
'Wine- Data of Second Dim(100)   0.94 |
Wine-NoCorrelation-Data  (100)   0.89 |
```

The best KPI is the F-Test. At .94, I see this metric making a case for including it in the final model evaluation.

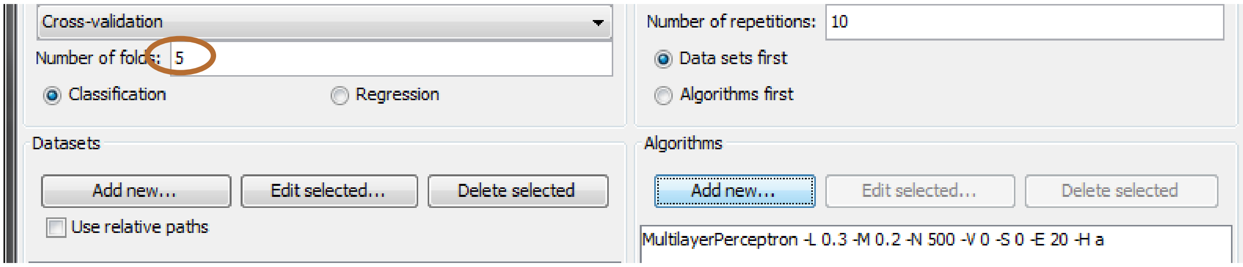
After conducting the iterations, I do not see this overall algorithm making it into the final rounds of evaluation, except for the F-Measure. Even after the iterations, JRIP did not classify "Wine- Data of Second Dim" better than Naive Bayes.

Artificial Neural Network

This EDA has seven different algorithms, and I expect the ANN known as Multilayer Perceptron to be one of the best classifiers. From past use, Multilayer Perceptron does well with classification as a result of its robust modeling technique.

	bayes.Na	(2) trees	(3) lazy.	(4) funct	(5) funct	(6) funct	(7) rules
'Wine With Correlated Dat(100)	100	88.60 *	100	93.76	92.51	93.70	90.76
'Wine with all data inclu(100)	97.46	93.20	95.12	98.02	97.23	97.92	92.97 *
'Wine- Data with all thre(100)	93	93.14	96.13	98.99	96.63	97.87	92.47 *
'Wine- Variables for firs(100)	86.37	84.84	80.00 *	89.87	83.89	83.44	83.71
'Wine- Data of Second Dim(100)	91.89	87.25	91.61	91.12	90.71	90.94	85.41 *
Wine-NoCorrelation-Data (100)	92.29	89.55	86.30 *	90.60	89.27	89.15	86.01

As seen above in the preliminary analysis, Multilayer Perceptron is consistently in the top three rankings for every dataset. Moving forward, I will be looking to improve this algorithm. As done with the other algorithms, I first start with analyzing the fold option in an effort to improve overall performance. The benchmark is 10 folds.



Cross-validation

Number of folds: 5

● Classification ○ Regression

Number of repetitions: 10

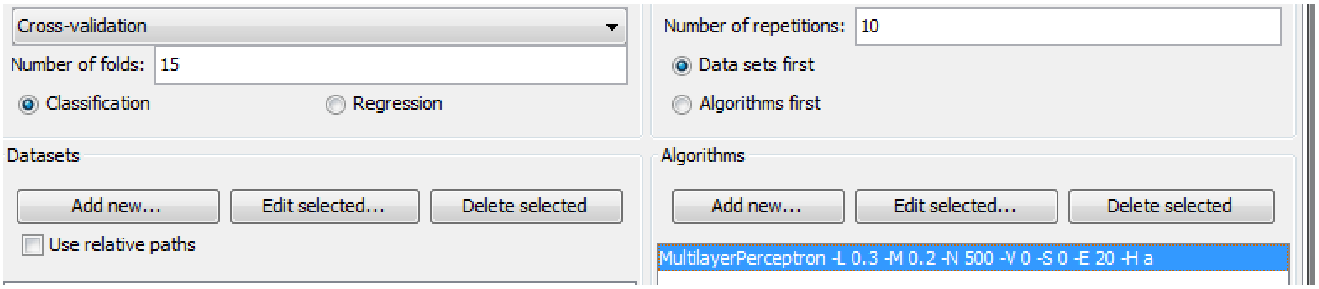
● Data sets first

○ Algorithms first

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

```
'Wine with all data inclu (50)    98.03 |
'Wine- Data with all thre (50)    98.71 |
'Wine With Correlated Dat (50)   100.00 |
'Wine- Variables for firs (50)    88.44 |
'Wine- Data of Second Dim (50)    90.90 |
Wine-NoCorrelation-Data   (50)    90.44 |
```

At 5 folds, the algorithm was not as accurate as 10 folds. Given the computational complexity of Multilayer Perceptron, this algorithm took the longest amount of time to run per iteration.



Cross-validation

Number of folds: 15

● Classification ○ Regression

Number of repetitions: 10

● Data sets first

○ Algorithms first

Datasets

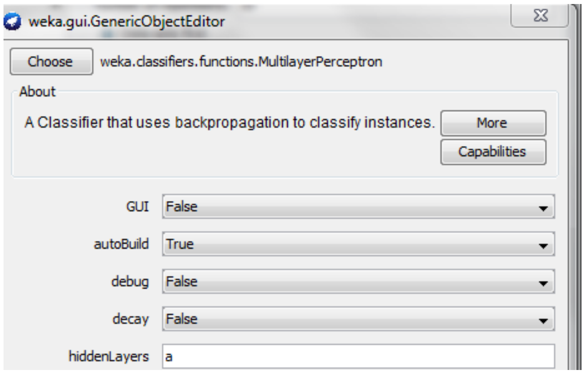
□ Use relative paths

Algorithms

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

```
'Wine with all data inclu(150)    98.25 |
'Wine- Data with all thre(150)    98.71 |
'Wine With Correlated Dat(150)   100.00 |
'Wine- Variables for firs(150)    88.43 |
'Wine- Data of Second Dim(150)    89.98 |
Wine-NoCorrelation-Data  (150)    90.62 |
```

With 15 folds, Multilayer Perceptron has the following output to the left. The overall accuracy percent increased for the vast majority of the datasets. I am going to stick with 15 folds for this algorithm.



weka.gui.GenericObjectEditor

weka.classifiers.functions.MultilayerPerceptron

About

A Classifier that uses backpropagation to classify instances.

GUI: False

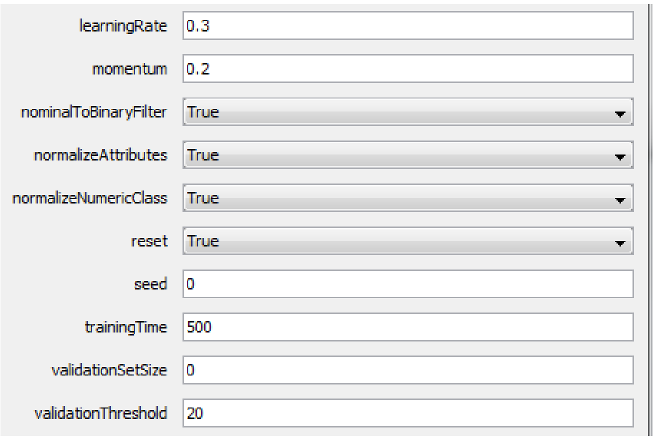
autoBuild: True

debug: False

decay: False

hiddenLayers: a

As seen to the left, there are many different modifications that can be done for this algorithm.



learningRate	0.3
momentum	0.2
nominalToBinaryFilter	True
normalizeAttributes	True
normalizeNumericClass	True
reset	True
seed	0
trainingTime	500
validationSetSize	0
validationThreshold	20

'Wine with all data inclu(150)   98.25 |

'Wine- Data with all thre(150)   98.71 |

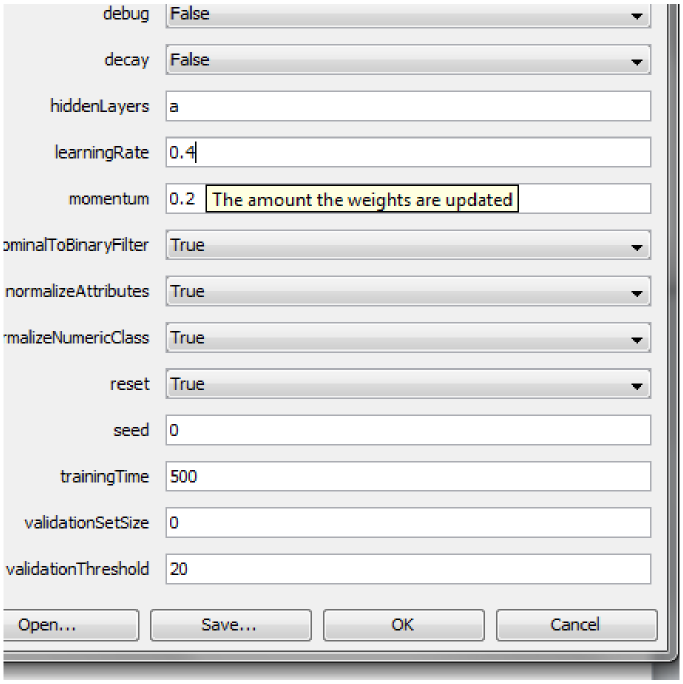
'Wine With Correlated Dat(150)   100.00 |

'Wine- Variables for firs(150)   88.43 |

'Wine- Data of Second Dim(150)   89.98 |

Wine-NoCorrelation-Data (150)   90.62 |

I changed the validation threshold from 20 to 10. This change, generated the output to the left. After taking the time to wait, this change had not effect on the classification accuracy.



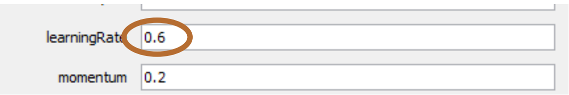
debug	False
decay	False
hiddenLayers	a
learningRate	0.4
momentum	0.2
ominalToBinaryFilter	True
normalizeAttributes	True
rmalizeNumericClass	True
reset	True
seed	0
trainingTime	500
validationSetSize	0
validationThreshold	20

The amount the weights are updated

The next iteration I did was change the learning rate from .3 to .4. My initial thinking is this will increase the accuracy based on the increase in learning rate.

```
'Wine with all data inclu(150)    98.25 |
'Wine- Data with all thre(150)    98.71 |
'Wine With Correlated Dat(150)   100.00 |
'Wine- Variables for firs(150)    88.33 |
'Wine- Data of Second Dim(150)    90.22 |
Wine-NoCorrelation-Data  (150)    91.01 |
```

After changing the learning rate, the 'Wine-NoCorrealtion-Data' dataset had a major improvement, while the other datasets increased or decreased slightly. I will increase the learning rate again and see if the model does not increase.

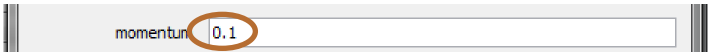


learningRate 0.6

momentum 0.2

```
'Wine with all data inclu(150)    98.20 |
'Wine- Data with all thre(150)    98.60 |
'Wine With Correlated Dat(150)   100.00 |
'Wine- Variables for firs(150)    88.99 |
'Wine- Data of Second Dim(150)    89.94 |
Wine-NoCorrelation-Data  (150)    90.23 |
```

After changing the learning rate to .6, the 'Wine-NoCorrealtion-Data' dataset did not improve, and the other datasets increased or decreased slightly. I will leave the learning rate at .4.



momentum 0.1

```
'Wine with all data inclu(150)    98.25 |
'Wine- Data with all thre(150)    98.71 |
'Wine With Correlated Dat(150)   100.00 |
'Wine- Variables for firs(150)    88.49 |
'Wine- Data of Second Dim(150)    89.88 |
Wine-NoCorrelation-Data  (150)    91.02 |
```

The momentum has been changed from .2 to .1. This feature does the following: 'momentum -- Momentum applied to the weights during updating –Weka.' With this new change, accuracy for classification can be seen to the left. I am going to increase it in the next iteration.

```
'Wine with all data inclu(150)    98.20 |
'Wine- Data with all thre(150)    98.71 |
'Wine With Correlated Dat(150)   100.00 |
'Wine- Variables for firs(150)    88.37 |
'Wine- Data of Second Dim(150)    90.11 |
Wine-NoCorrelation-Data  (150)    90.95
```

With the momentum at .3, the accuracy for the datasets is to the left. This is a mild decrease, I will increase it to .6 for the next iteration.

momentum 0.6

```
'Wine with all data inclu(150)    98.20 |
'Wine- Data with all thre(150)    98.65 |
'Wine With Correlated Dat(150)   100.00 |
'Wine- Variables for firs(150)    88.65 |
'Wine- Data of Second Dim(150)    89.94 |
Wine-NoCorrelation-Data  (150)    90.26
```

As seen to the left, the accuracy has decreased as well at .6. In conclusion, I will keep the momentum at the original setting of .2. The other settings for Multilayer Perceptron are in the optimal setting for the datasets. The final output for this algorithm can be seen to the left. The root mean square error and F-test values reflect that this is one of the strongest models if not the strongest model of the datasets.

```
'Wine with all data inclu(150)   0.07 |
'Wine- Data with all thre(150)   0.06 |
'Wine With Correlated Dat(150)   0.00 |
'Wine- Variables for firs(150)   0.24 |
'Wine- Data of Second Dim(150)   0.20 |
Wine-NoCorrelation-Data  (150)   0.20 |
'Wine with all data inclu(150)   0.07 |
'Wine- Data with all thre(150)   0.06 |
'Wine With Correlated Dat(150)   0.00 |
'Wine- Variables for firs(150)   0.24 |
'Wine- Data of Second Dim(150)   0.20 |
Wine-NoCorrelation-Data  (150)   0.20 |
```

The root mean square error (RMSE) reflects a good fit for the top three datasets, but the bottom three datasets has a rather large RMSE.

```
'Wine with all data inclu(150)   0.99 |
'Wine- Data with all thre(150)   1.00 |
'Wine With Correlated Dat(150)   1.00 |
'Wine- Variables for firs(150)   0.85 |
'Wine- Data of Second Dim(150)   0.93 |
Wine-NoCorrelation-Data  (150)   0.91 |
```

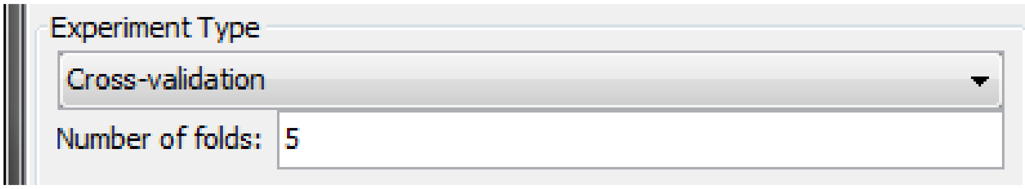
The F-test is reflective of the other KPIs. Although Multilayer Perceptron classified well, I see the low F-measures as an indicator that perhaps other algorithms are a better fit to the data.

Logistic Regression

The classification variables are in a discretized categorization, and from the initial output this algorithm did well.

```
   'Wine With Correlated Dat(100)   100 |  88.60 *   100     93.76    92.51    93.70    90.76
 'Wine with all data inclu(100)   97.46 |  93.20    95.12    98.02    97.23    97.92    92.97 *
    'Wine- Data with all thre(100)   93 |  93.14    96.13    98.99    96.63    97.87    92.47 *
  'Wine- Variables for firs(100)  86.37 |  84.84    80.00 *  89.87    83.89    83.44    83.71
'Wine- Data of Second Dim(100)   91.89 |  87.25    91.61    91.12    90.71    90.94    85.41 *
  Wine-NoCorrelation-Data  (100)  92.29 |  89.55    86.30 *  90.60    89.27    89.15    86.01
```

I will start with the folds option to analyze potential improvements on the overall model.



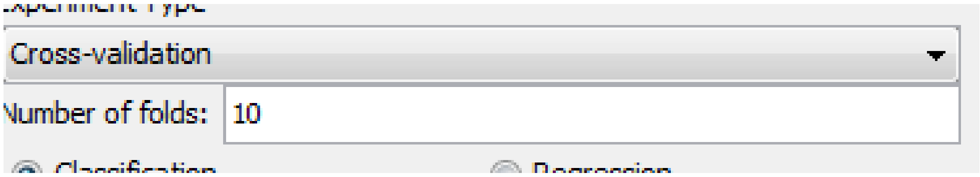
Experiment Type

Cross-validation

Number of folds: 5

```
'Wine with all data inclu (50)    97.14 |
'Wine- Data with all thre (50)    96.63 |
'Wine With Correlated Dat (50)   100.00 |
'Wine- Variables for firs (50)    83.88 |
'Wine- Data of Second Dim (50)    90.17 |
Wine-NoCorrelation-Data   (50)    89.27 |
```

5 folds had an inverse relationship with overall accuracy throughout the datasets.



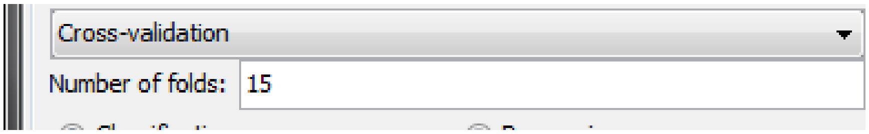
Experiment Type

Cross-validation

Number of folds: 10

```
'Wine with all data inclu(100)    97.23 |
'Wine- Data with all thre(100)    96.63 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    83.89 |
'Wine- Data of Second Dim(100)    90.71 |
Wine-NoCorrelation-Data  (100)    89.27
```

As the folds increase, I am seeing an increase in overall accuracy.



Cross-validation

Number of folds: 15

'Wine with all data inclu(150)   96.91 |

'Wine- Data with all thre(150)   96.68 |

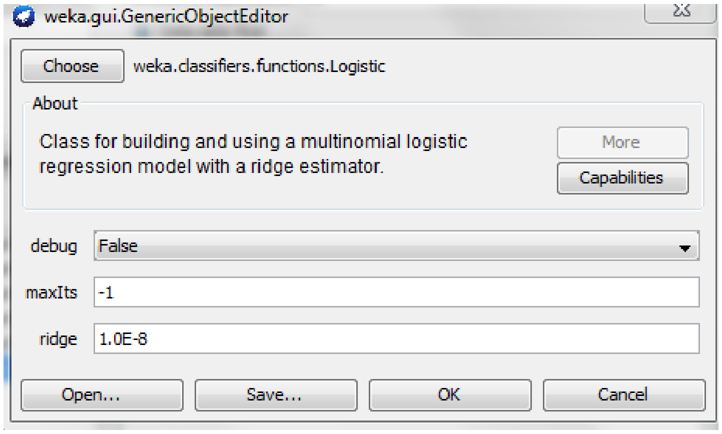
'Wine With Correlated Dat(150)   100.00 |

'Wine- Variables for firs(150)   84.37 |

'Wine- Data of Second Dim(150)   90.56 |

Wine-NoCorrelation-Data (150)   89.40

At 15 folds, accuracy is less than at 10. I am going to keep the folds to 10 based on the accuracy parameter.



weka.gui.GenericObjectEditor

Choose weka.classifiers.functions.Logistic

About

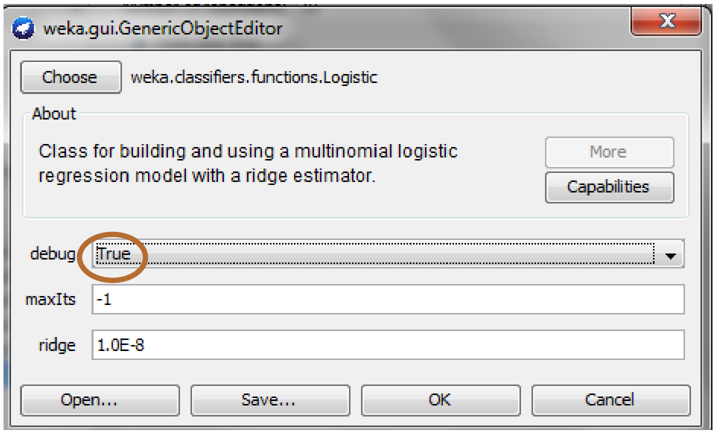
Class for building and using a multinomial logistic regression model with a ridge estimator.

debug False

maxIts -1

ridge 1.0E-8

These are the following iterations that can be made to the Logistic Regression algorithm. Moving forward I will work through each option to find the optimal fit to the data.



weka.gui.GenericObjectEditor

Choose weka.classifiers.functions.Logistic

About

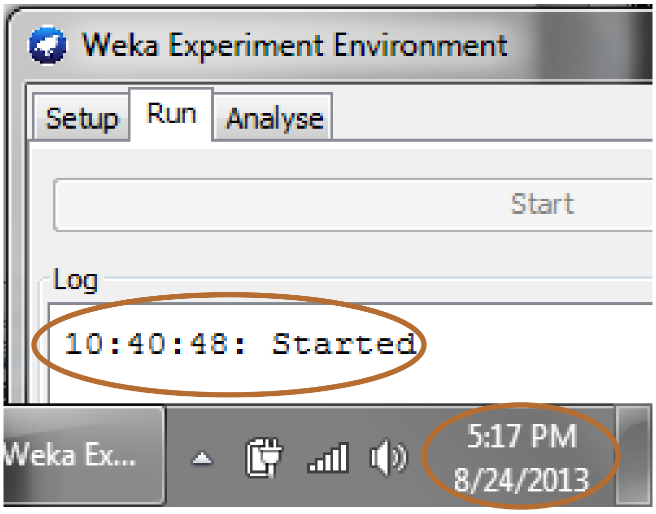
Class for building and using a multinomial logistic regression model with a ridge estimator.

debug True

maxIts -1

ridge 1.0E-8

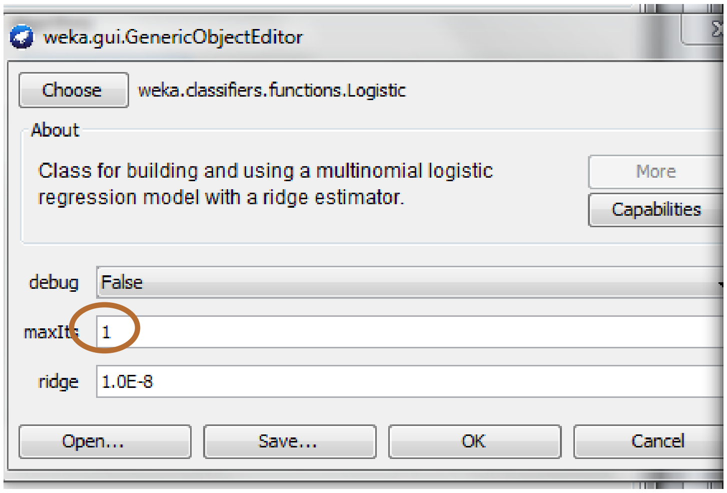
My first iteration is to change the debug feature.



10:40:48: Started

5:17 PM
8/24/2013

I ran this modification for almost seven hours before I decided to stop it. This modification in Weka is too timely for me to an analysis worth extrapolating.



weka.classifiers.functions.Logistic

Class for building and using a multinomial logistic regression model with a ridge estimator.

debug False

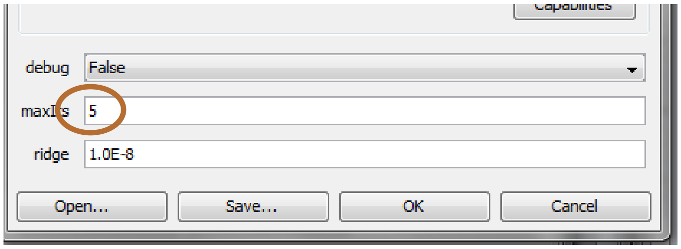
maxIts 1

ridge 1.0E-8

The maxIts function pertains to the iterations for the overall algorithm. In this iteration, I have changed it to 1 from -1.

```
'Wine with all data inclu(100)    93.07 |

'Wine- Data with all thre(100)    95.83 |

'Wine With Correlated Dat(100)   100.00 |

'Wine- Variables for firs(100)    74.76 |

'Wine- Data of Second Dim(100)    73.89 |

Wine-NoCorrelation-Data  (100)    87.65 |
```

Changing to 1 had an inverse effect on accuracy. My next iteration will be at 5.



debug False

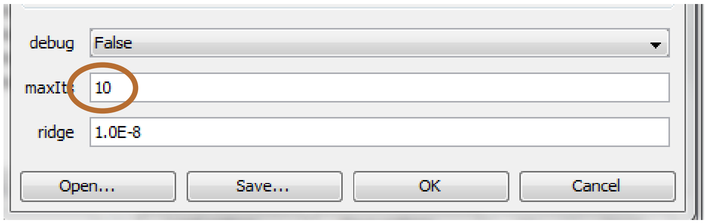
maxIts 5

ridge 1.0E-8

Notice the 5 maxIts.

```
'Wine with all data inclu(100)    97.23 |
'Wine- Data with all thre(100)    97.11 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    80.50 |
'Wine- Data of Second Dim(100)    88.63 |
Wine-NoCorrelation-Data  (100)    90.45 |
```

Five iterations had a positive effect on a few of the datasets. Now, I will look add 10 in the maxIts column.



debug False

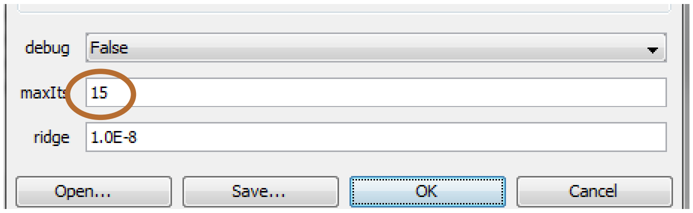
maxIts 10

ridge 1.0E-8

Notice the 10 maxIts.

```
'Wine with all data inclu(100)    97.06 |
'Wine- Data with all thre(100)    97.23 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    84.45 |
'Wine- Data of Second Dim(100)    89.70 |
Wine-NoCorrelation-Data  (100)    89.67 |
```

The 10 iterations have increased three of the six datasets. I will move on to 15 iterations next.



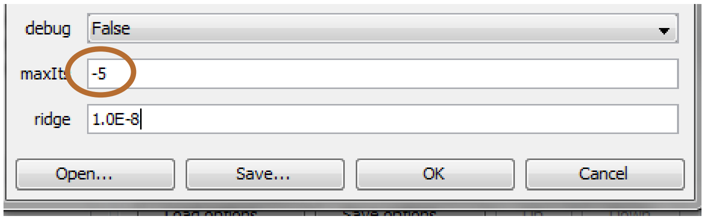
debug: False

maxIts: 15

ridge: 1.0E-8

```
'Wine with all data inclu(100)    96.83 |
'Wine- Data with all thre(100)    97.00 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    84.56 |
'Wine- Data of Second Dim(100)    90.90 |
Wine-NoCorrelation-Data  (100)    89.56 |
```

At 15 iterations, ''Wine- Data of Second Dim' is the only dataset that has continued to increase in accuracy. In my opinion, 15 iterations is too many, and I will settle on 5.



debug: False

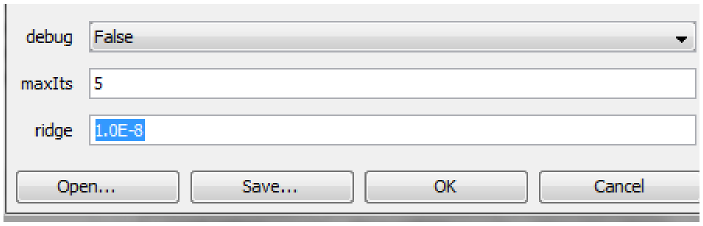
maxIts: -5

ridge: 1.0E-8

Given that the first iteration for this model was -1 I want to make one last iteration at -5.

```
'Wine with all data inclu(100)    39.91 |
'Wine- Data with all thre(100)    39.91 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    39.91 |
'Wine- Data of Second Dim(100)    39.91 |
Wine-NoCorrelation-Data  (100)    39.91 |
```

At -5 iterations, it can clearly be seen that the overall model regresses in its accuracy for all the datasets except one.

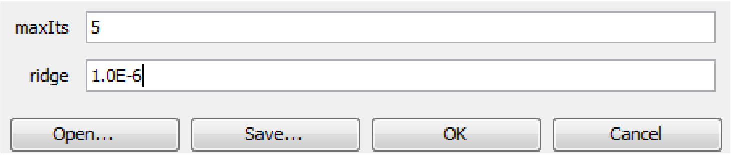


debug: False

maxIts: 5

ridge: 1.0E-8

The last iteration option is the ridge function, and it represents the log-likelihood.



maxIts: 5

ridge: 1.0E-6

I changed the log-likelihood to -6 from the standard -8.

'Wine with all data inclu(100)   97.23 |

'Wine- Data with all thre(100)   97.11 |

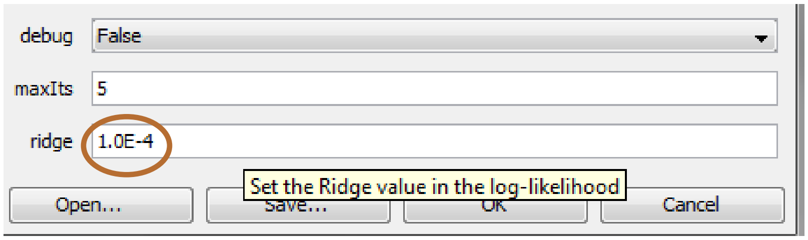
'Wine With Correlated Dat(100)   100.00 |

'Wine- Variables for firs(100)   80.50 |

'Wine- Data of Second Dim(100)   88.63 |

Wine-NoCorrelation-Data (100)   90.45 |

This had not effect on the model accuracy and I will change the ridge to -4.



debug: False

maxIts: 5

ridge: 1.0E-4

Set the Ridge value in the log-likelihood

Notice the -4.

'Wine with all data inclu(100)   97.23 |

'Wine- Data with all thre(100)   97.11 |

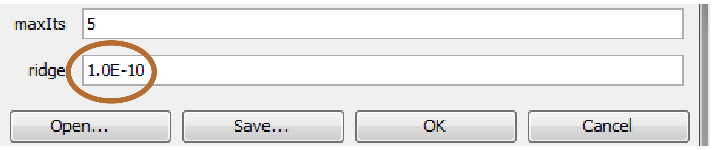
'Wine With Correlated Dat(100)   100.00 |

'Wine- Variables for firs(100)   80.50 |

'Wine- Data of Second Dim(100)   88.63 |

Wine-NoCorrelation-Data (100)   90.45 |

Notice again how none of the values changed. I will now increase the log-likelihood.

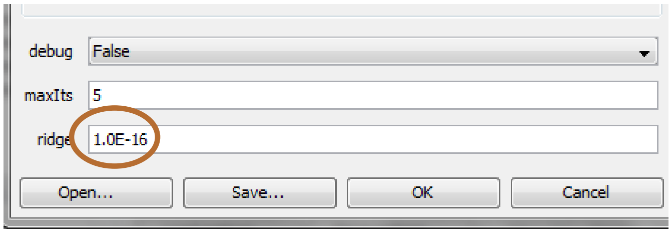


maxIts 5

ridge 1.0E-10

```
'Wine with all data inclu(100)    97.23 |
'Wine- Data with all thre(100)    97.11 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    80.50 |
'Wine- Data of Second Dim(100)    88.63 |
Wine-NoCorrelation-Data  (100)    90.45 |
```

Increasing the ridge appears to have no effect on accuracy. I will increase the ridge once more.



debug False

maxIts 5

ridge 1.0E-16

```
'Wine with all data inclu(100)    97.23 |
'Wine- Data with all thre(100)    97.11 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    80.50 |
'Wine- Data of Second Dim(100)    88.63 |
Wine-NoCorrelation-Data  (100)    90.45 |
```

Increasing the negative value of the ridge value had not effect on the output. I will keep the value at the original number of -8.

```
'Wine with all data inclu(100)   0.08 |
'Wine- Data with all thre(100)   0.09 |
'Wine With Correlated Dat(100)   0.00 |
'Wine- Variables for firs(100)   0.30 |
'Wine- Data of Second Dim(100)   0.23 |
Wine-NoCorrelation-Data  (100)   0.21 |
```

The root mean square error is okay. I would have expected the last three data-sets to have a better RMSE.

```
'Wine with all data inclu(100)   0.98 |
'Wine- Data with all thre(100)   0.98 |
'Wine With Correlated Dat(100)   1.00 |
'Wine- Variables for firs(100)   0.74 |
'Wine- Data of Second Dim(100)   0.91 |
Wine-NoCorrelation-Data  (100)   0.91 |
```

The F-Measure reflects a model that is well fit except for the ‘‘Wine- Variables for firs’.

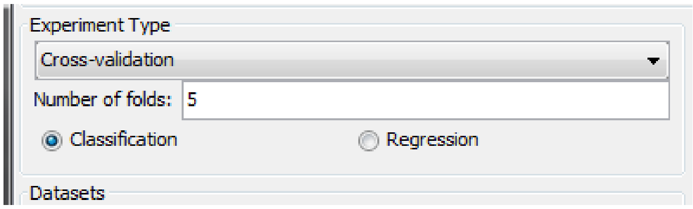
I spent quite a bit of time trying to find the best iterations for the Logistic model, this was due in part to the fact that the goal of this data is to predict classification. Classification is a ternary response for the datasets, and I would have expected the models to have improved better.

Simple Logistic

Given the ternary classification, I want to try fitting the data with another logistic model.

```
    'Wine With Correlated Dat(100)   100 |   88.60 *   100      93.76     92.51     93.70     90.76
   'Wine with all data inclu(100)  97.46 |   93.20     95.12     98.02     97.23     97.92     92.97 *
    'Wine- Data with all thre(100)   93 |   93.14     96.13     98.99     96.63     97.87     92.47 *
   'Wine- Variables for firs(100)   86.37 |   84.84     80.00 *   89.87     83.89     83.44     83.71
'Wine- Data of Second Dim(100)   91.89 |   87.25     91.61     91.12     90.71     90.94     85.41 *
  Wine-NoCorrelation-Data  (100)   92.29 |   89.55     86.30 *   90.60     89.27     89.15     86.01
```

I will start with the folds option to analyze potential improvements on the overall model.


Experiment Type

Cross-validation

Number of folds: 5

● Classification ○ Regression

Datasets

'Wine with all data inclu (50)   97.63 |

'Wine- Data with all thre (50)   97.75 |

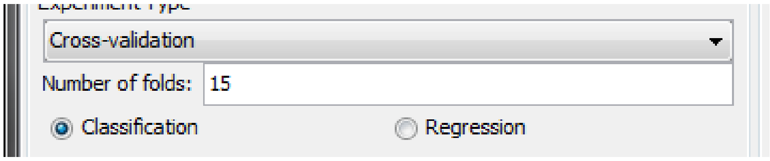
'Wine With Correlated Dat (50)   100.00 |

'Wine- Variables for firs (50)   83.72 |

'Wine- Data of Second Dim (50)   90.28 |

Wine-NoCorrelation-Data   (50)   89.32 |

As seen with the Logistic model, the smaller the number of folds the less accurate the classification.


Cross-validation

Number of folds: 15

● Classification ○ Regression

'Wine with all data inclu (50)   97.63 |

'Wine- Data with all thre (50)   97.75 |

'Wine With Correlated Dat (50)   100.00 |

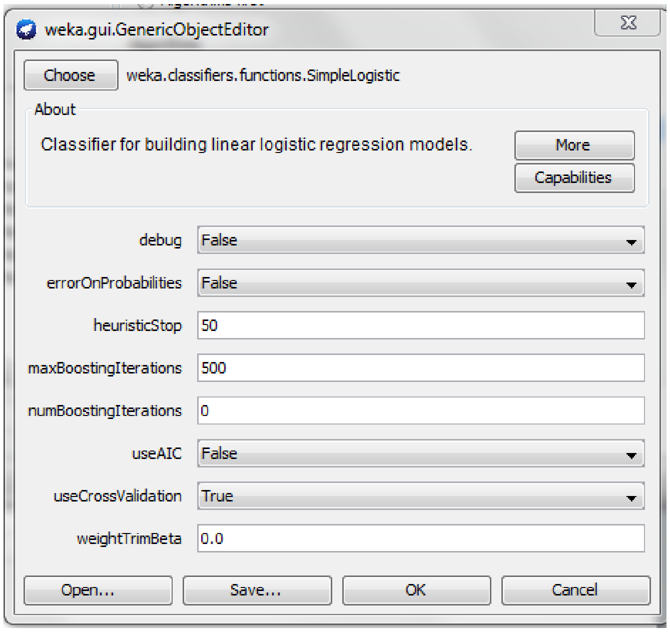
'Wine- Variables for firs (50)   83.72 |

'Wine- Data of Second Dim (50)   90.28 |

Wine-NoCorrelation-Data   (50)   89.32 |

15 folds are very computationally heavy on Weka, and the increase in accuracy is nominal at best.

I am going to keep the folds at 10 because the changes at 15 were not enough for me to justify risking over fitting the data. The next step is to explore the iterations within the simple logistic algorithm.



weka.gui.GenericObjectEditor

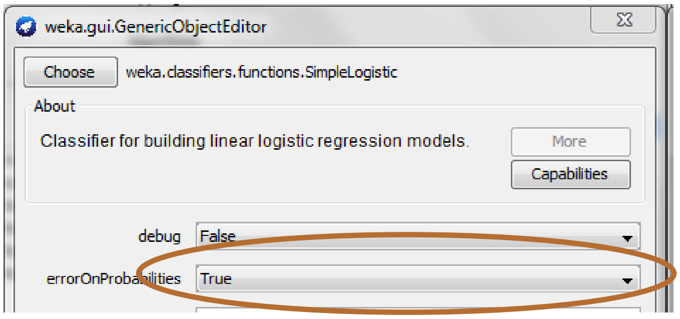
Choose weka.classifiers.functions.SimpleLogistic

About

Classifier for building linear logistic regression models.

debug	False
errorOnProbabilities	False
heuristicStop	50
maxBoostingIterations	500
numBoostingIterations	0
useAIC	False
useCrossValidation	True
weightTrimBeta	0.0

The options to the left are the different iterations available in Weka. I will not debug to True based on the fact that it takes well over 5 hours to run.



weka.gui.GenericObjectEditor

Choose weka.classifiers.functions.SimpleLogistic

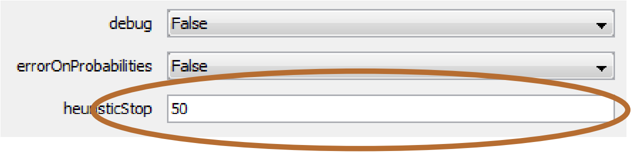
About

Classifier for building linear logistic regression models.

debug	False
errorOnProbabilities	True

The error on probabilities does the following according to Weka, “errorOnProbabilities -- Use error on the probabilties as error measure when determining the best number of LogitBoost iterations. If set, the number of LogitBoost iterations is chosen that minimizes the root mean squared error (either on the training set or in the cross-validation, depending on useCrossValidation). Given that I am trying to reduce the RMSE, I will move this tab to true. The overall model improved with turning the process to ‘True’.

```
'Wine with all data inclu (50)    97.63 |
'Wine- Data with all thre (50)    97.75 |
'Wine With Correlated Dat (50)   100.00 |
'Wine- Variables for firs (50)    83.72 |
'Wine- Data of Second Dim (50)    90.28 |
Wine-NoCorrelation-Data   (50)    89.32 |
```

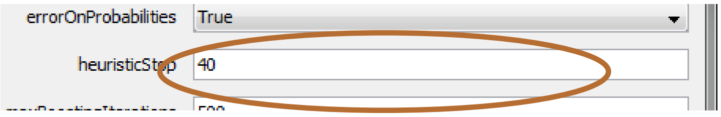


debug: False

errorOnProbabilities: False

heuristicStop: 50

Weka describes heuristicStop as, “heuristicStop -- If heuristicStop > 0, the heuristic for greedy stopping while cross-validating the number of LogitBoost iterations is enabled. This means LogitBoost is stopped if no new error minimum has been reached in the last heuristicStop iterations. It is recommended to use this heuristic, it gives a large speed-up especially on small datasets. The default value is 50”. I will start by decreasing then increasing the heuristic stop.



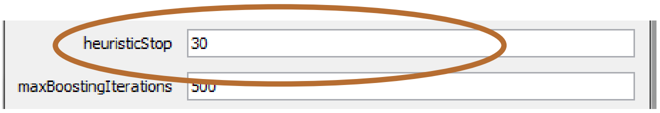
errorOnProbabilities: True

heuristicStop: 40

Notice 40.

```
'Wine with all data inclu(100)    97.58 |
'Wine- Data with all thre(100)    98.14 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    83.94 |
'Wine- Data of Second Dim(100)    90.60 |
Wine-NoCorrelation-Data  (100)    89.27 |
```

Decreasing the heuristic step increased accuracy. I will drop it again down to 30.

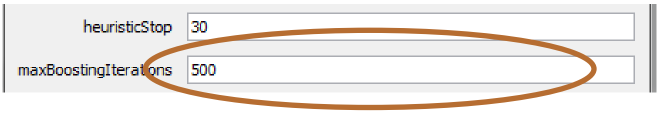


heuristicStop: 30

maxBoostingIterations: 500

```
'Wine with all data inclu(100)    97.58 |
'Wine- Data with all thre(100)    98.14 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    83.94 |
'Wine- Data of Second Dim(100)    90.60 |
Wine-NoCorrelation-Data  (100)    89.27 |
```

At 30, the model slightly increases and I will leave it at 30.



heuristicStop 30

maxBoostingIterations 500

Weka states the maxBoostingIterations does the following,

“maxBoostingIterations -- Sets the maximum number of iterations for LogitBoost. Default value is 500, for very small/large datasets a lower/higher value might be preferable.’ I see the iterations are at 100 for the datasets. Given how the iterations had little effect in the previous model, I am not going modify the option. For the other iteration options, they revolve around modifying cross-validation. I am not going to change the model of cross-validation, thus I will leave the other options alone.

```
'Wine with all data inclu(100)   0.07 |
'Wine- Data with all thre(100)   0.06 |
'Wine With Correlated Dat(100)   0.00 |
'Wine- Variables for firs(100)   0.28 |
'Wine- Data of Second Dim(100)   0.20 |
Wine-NoCorrelation-Data  (100)   0.22
```

The root mean square error is okay. I would have expected the last three data-sets to have a better RMSE. The RMSE is better than the Logistic model.

```
'Wine with all data inclu(100)   0.99 |
'Wine- Data with all thre(100)   0.99 |
'Wine With Correlated Dat(100)   1.00 |
'Wine- Variables for firs(100)   0.78 |
'Wine- Data of Second Dim(100)   0.93 |
Wine-NoCorrelation-Data  (100)   0.88 |
```

The F-measure is impressive, and in my opinion this model fits the data sets well.

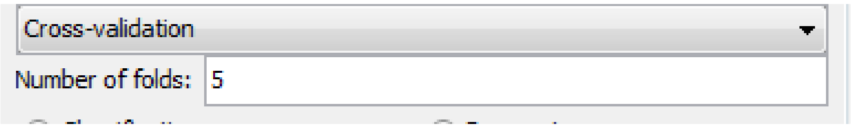
The Simple logistic model allowed more iterations and improved from its original state.

JRIP:

The last classifier algorithm is rule based, and in the preliminary analysis it performed very poorly.

```
 'Wine With Correlated Dat(100)   100 |  88.60 *   100     93.76     92.51     93.70     90.76
'Wine with all data inclu(100)   97.46 |   93.20     95.12     98.02     97.23     97.92     92.97 *
 'Wine- Data with all thre(100)   93 |   93.14     96.13     98.99     96.63     97.87     92.47 *
 'Wine- Variables for firs(100)   86.37 |   84.84     80.00 *   89.87     83.89     83.44     83.71
'Wine- Data of Second Dim(100)   91.89 |   87.25     91.61     91.12     90.71     90.94     85.41 *
 Wine-NoCorrelation-Data  (100)   92.29 |   89.55     86.30 *   90.60     89.27     89.15     86.01
```

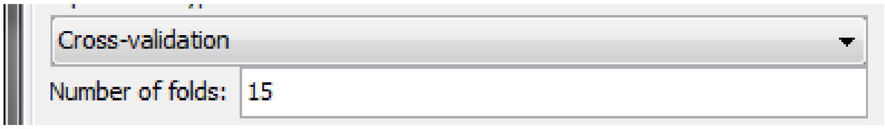
I will start with the folds option to analyze potential improvements on the overall model.


Cross-validation

Number of folds: 5

```
'Wine with all data inclu (50)    91.34 |
'Wine- Data with all thre (50)    91.69 |
'Wine With Correlated Dat (50)   100.00 |
'Wine- Variables for firs (50)    81.54 |
'Wine- Data of Second Dim (50)    84.90 |
Wine-NoCorrelation-Data   (50)    86.24 |
```

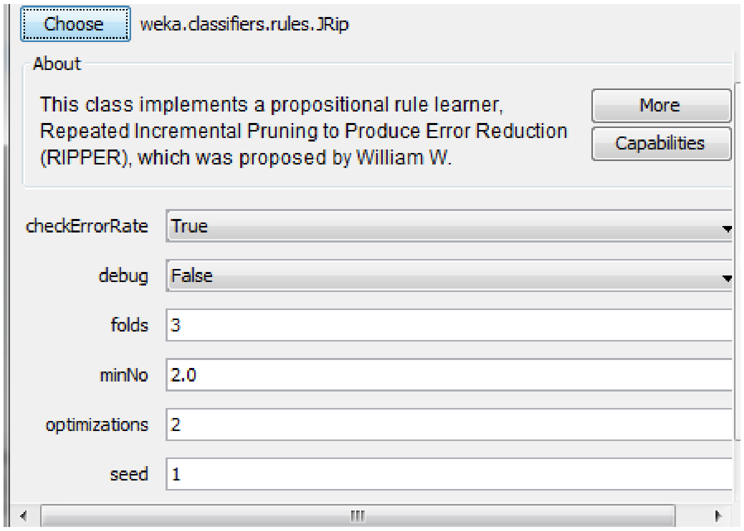
At 5 folds, the overall datasets do not improve, I will increase the folds.


Cross-validation

Number of folds: 15

```
'Wine with all data inclu(150)    92.37 |
'Wine- Data with all thre(150)    92.27 |
'Wine With Correlated Dat(150)   100.00 |
'Wine- Variables for firs(150)    83.17 |
'Wine- Data of Second Dim(150)    85.98 |
Wine-NoCorrelation-Data  (150)    86.15 |
```

At 15 folds, there is overall improvement, but it does not surpass the accuracy of 10 folds. For this algorithm, I will stay with the original 10 folds option.



weka.classifiers.rules.JRip

About

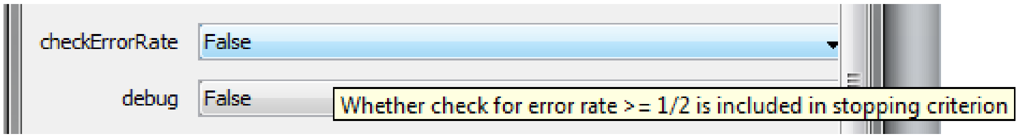
This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which was proposed by William W.

checkErrorRate: True
debug: False
folds: 3
minNo: 2.0
optimizations: 2
seed: 1

The options to the left are the different iterations available in Weka. I will not debug to True based on the fact that it takes well over 5 hours to run.

According to Weka the check Error Rate function does the following, “check for error rate >= 1/2 is included in stopping criterion.” I will move this to false to analyze the impact on accuracy.

Moving this function to false improved the overall model accuracy for all the datasets and I will keep it at false.



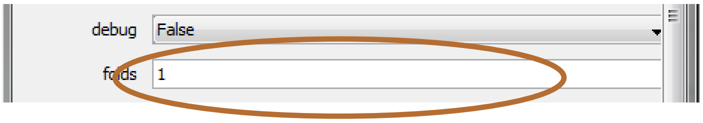
checkErrorRate: False
debug: False

Whether check for error rate >= 1/2 is included in stopping criterion

```
'Wine with all data inclu(100)    93.02 |
'Wine- Data with all thre(100)    92.36 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    83.76 |
'Wine- Data of Second Dim(100)    85.06 |
Wine-NoCorrelation-Data  (100)    85.66 |
```

Moving this function to false improved the overall model accuracy for all the datasets and I will keep it at false.

According to Weka the folds option does the following, "Determines the amount of data used for pruning. One fold is used for pruning, the rest for growing the rules."

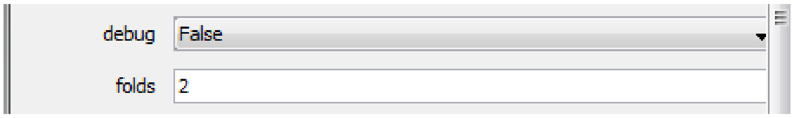


debug False

folds 1

```
'Wine with all data inclu(100)    39.91 |
'Wine- Data with all thre(100)    39.91 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    39.91 |
'Wine- Data of Second Dim(100)    39.91 |
Wine-NoCorrelation-Data  (100)    39.91 |
```

One fold had a negative impact on model accuracy; I now will move it up to 2 folds.

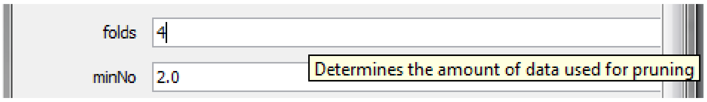


debug False

folds 2

```
'Wine with all data inclu(100)    91.79 |
'Wine- Data with all thre(100)    91.77 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    83.44 |
'Wine- Data of Second Dim(100)    84.50 |
Wine-NoCorrelation-Data  (100)    85.35
```

Changing the option to two folds greatly improved the model, but it is not as accurate as 3 folds. I will move on to four folds next.



folds 4

minNo 2.0

Determines the amount of data used for pruning

'Wine with all data inclu(100)   92.88 |

'Wine- Data with all thre(100)   92.19 |

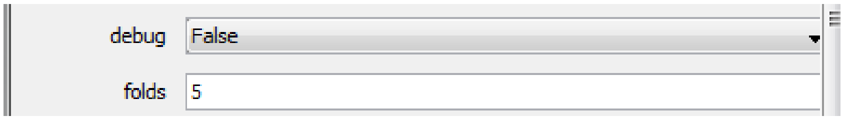
'Wine With Correlated Dat(100)   100.00 |

'Wine- Variables for firs(100)   83.34 |

'Wine- Data of Second Dim(100)   85.40 |

Wine-NoCorrelation-Data (100)   86.51

At four folds, the improvement is not as accurate as three folds. I will try five folds as one more option.



debug False

folds 5

'Wine with all data inclu(100)   92.70 |

'Wine- Data with all thre(100)   92.37 |

'Wine With Correlated Dat(100)   100.00 |

'Wine- Variables for firs(100)   82.75 |

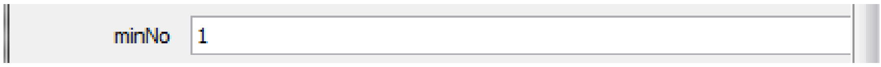
'Wine- Data of Second Dim(100)   84.02 |

Wine-NoCorrelation-Data (100)   86.42 |

At five folds the accuracy continues to wane, I've come to the conclusion that as I continue to add folds past three, the accuracy continues to worsen. I will stay at three folds.

minNo 2.0

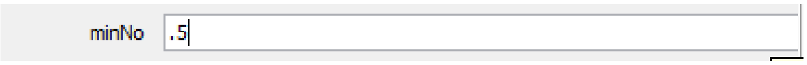
According to Weka the minNo option does the following, 'The minimum total weight of the instances in a rule.' I will move forward by adjusting this option.



minNo 1

```
'Wine with all data inclu(100)    93.08 |
'Wine- Data with all thre(100)    93.04 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    82.80 |
'Wine- Data of Second Dim(100)    85.25 |
Wine-NoCorrelation-Data  (100)    85.24
```

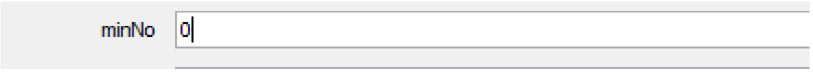
Moving this option to 1 had a slight overall positive impact, I will drop the value again to see the impact.



minNo .5

```
'Wine with all data inclu(100)    93.08 |
'Wine- Data with all thre(100)    93.04 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    82.80 |
'Wine- Data of Second Dim(100)    85.25 |
Wine-NoCorrelation-Data  (100)    85.24
```

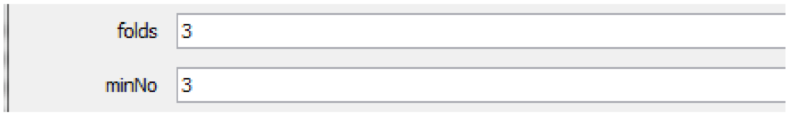
Moving the value to .5 had no effect on accuracy. I will move it to 0.



minNo 0

```
'Wine with all data inclu(100)    93.08 |
'Wine- Data with all thre(100)    93.04 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    82.80 |
'Wine- Data of Second Dim(100)    85.25 |
Wine-NoCorrelation-Data  (100)    85.24
```

Moving the value to 0 had no effect, I will increase the value.

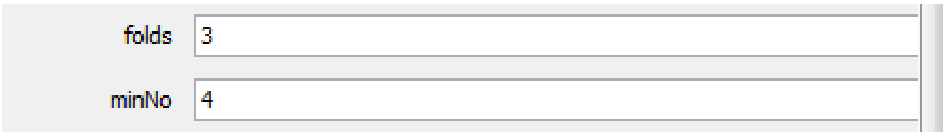


folds 3

minNo 3

```
'Wine with all data inclu(100)    92.40 |
'Wine- Data with all thre(100)    92.37 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    83.05 |
'Wine- Data of Second Dim(100)    84.78 |
Wine-NoCorrelation-Data  (100)    85.95 |
```

Increasing the minNo also decreases the model accuracy past two; I will increase the value once more for continuity.

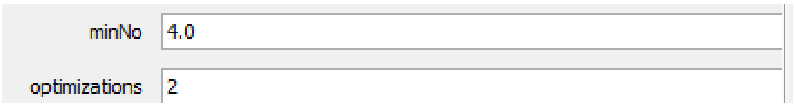


folds 3

minNo 4

```
'Wine with all data inclu(100)    91.73 |
'Wine- Data with all thre(100)    92.54 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    82.76 |
'Wine- Data of Second Dim(100)    84.32 |
Wine-NoCorrelation-Data  (100)    86.06 |
```

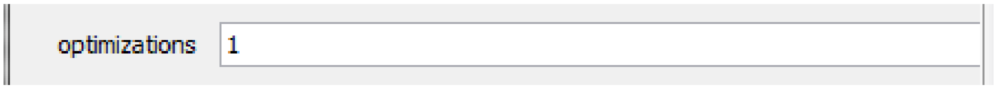
Again, the model accuracy drops as the minNo increases. I will settle on the original value of two.



minNo 4.0

optimizations 2

According to Weka the optimizations option does the following, ‘The number of optimization runs.’ I will move forward by adjusting this option.


optimizations 1

'Wine with all data inclu(100)   92.03 |

'Wine- Data with all thre(100)   92.42 |

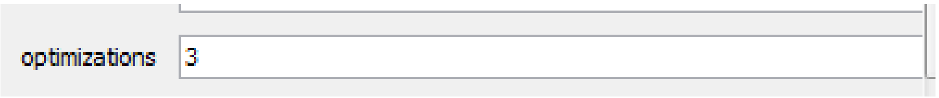
'Wine With Correlated Dat(100)   100.00 |

'Wine- Variables for firs(100)   82.25 |

'Wine- Data of Second Dim(100)   84.45 |

Wine-NoCorrelation-Data (100)   85.50

The model decreased in accuracy when moving the optimizations to one, I will move the optimizations above two.


optimizations 3

'Wine with all data inclu(100)   92.52 |

'Wine- Data with all thre(100)   92.70 |

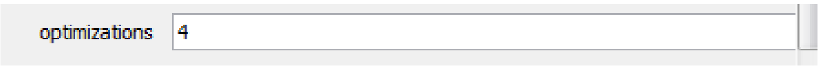
'Wine With Correlated Dat(100)   100.00 |

'Wine- Variables for firs(100)   82.71 |

'Wine- Data of Second Dim(100)   84.33 |

Wine-NoCorrelation-Data (100)   86.90 |

The model is still not at its default optimum. I will increase the value once more.


optimizations 4

'Wine with all data inclu(100)   92.59 |

'Wine- Data with all thre(100)   93.34 |

'Wine With Correlated Dat(100)   100.00 |

'Wine- Variables for firs(100)   83.10 |

'Wine- Data of Second Dim(100)   84.71 |

Wine-NoCorrelation-Data (100)   87.34

A few of the datasets increased, but overall the model did not improve with increasing the optimizations. I will keep this option at its default value of 2.

According to Weka the seed option does the following, The seed used for randomizing the data.' Initially, I think this will not have an effect on the overall model accuracy given that this option pertains to randomizing the data.

seed 2

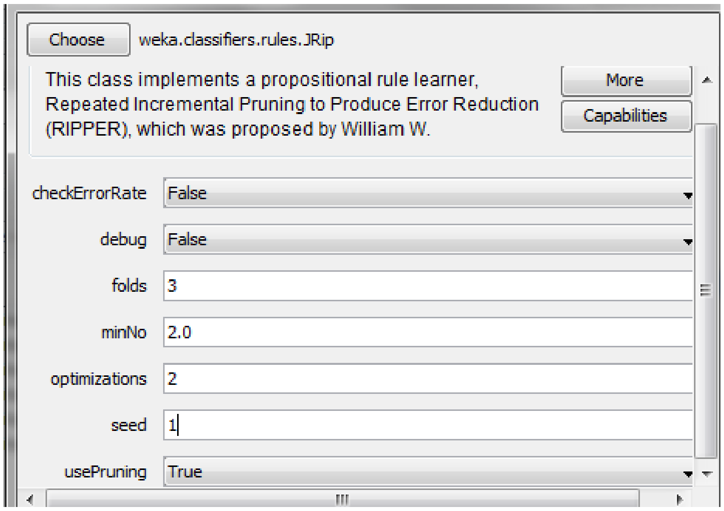
```
'Wine with all data inclu(100)    92.59 |
'Wine- Data with all thre(100)    93.34 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    83.10 |
'Wine- Data of Second Dim(100)    84.71 |
Wine-NoCorrelation-Data  (100)    87.34
```

Increasing the seed to two decreased model accuracy. I will increase this value once more.

seed 3

```
'Wine with all data inclu(100)    92.81 |
'Wine- Data with all thre(100)    92.63 |
'Wine With Correlated Dat(100)   100.00 |
'Wine- Variables for firs(100)    82.48 |
'Wine- Data of Second Dim(100)    84.09 |
Wine-NoCorrelation-Data  (100)    86.68
```

A few of the datasets increased, but overall the accuracy did not increase. I am going to keep this value at its default setting as well.



weka.classifiers.rules.JRip

This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), which was proposed by William W.

```
checkErrorRate  False
         debug  False
         folds  3
         minNo  2.0
 optimizations  2
          seed  1
    usePruning  True
```

The final iterations have the following output on the overall model.

```
'Wine with all data inclu(100)   0.18 |
'Wine- Data with all thre(100)   0.19 |
'Wine With Correlated Dat(100)   0.00 |
'Wine- Variables for firs(100)   0.29 |
'Wine- Data of Second Dim(100)   0.28 |
Wine-NoCorrelation-Data  (100)   0.28 |
```

The root mean square error does not reflect a well fit model compared to the other algorithms.

```
'Wine with all data inclu(100)   0.93 |
'Wine- Data with all thre(100)   0.92 |
'Wine With Correlated Dat(100)   1.00 |
'Wine- Variables for firs(100)   0.84 |
'Wine- Data of Second Dim(100)   0.88 |
Wine-NoCorrelation-Data  (100)   0.83 |
```

The final F-Measure as well shows that this algorithm is the worst for the datasets.

Compilation of Best Algorithms

Before modifying the data and algorithms, the benchmark can be seen below.

```
 'Wine With Correlated Dat(100)   100 |   88.60 *   100      93.76     92.51     93.70     90.76
'Wine with all data inclu(100)   97.46 |   93.20     95.12     98.02     97.23     97.92     92.97 *
 'Wine- Data with all thre(100)   93 |   93.14     96.13     98.99     96.63     97.87     92.47 *
'Wine- Variables for firs(100)   86.37 |   84.84     80.00 *   89.87     83.89     83.44     83.71
'Wine- Data of Second Dim(100)   91.89 |   87.25     91.61     91.12     90.71     90.94     85.41 *
Wine-NoCorrelation-Data  (100)   92.29 |   89.55     86.30 *   90.60     89.27     89.15     86.01
```

After all the modifications, the algorithms had the following values:

	Naive Bayes	J48	IBK	ANN	Logistic	Simple L	JRIP
Wine with all data inclu	98.37	93.21	95.12	98.25	97.06	97.58	93.02
Wine- Data with all thre	98.37	93.15	96.13	98.71	97.23	98.14	92.36
'Wine With Correlated Dat	100	100	100	100	100	100	100
Wine- Variables for firs	83.89	85.35	80	88.49	84.45	83.94	83.76
'Wine- Data of Second Dim	89.32	86.92	91.61	89.88	89.7	90.6	85.06
Wine-NoCorrelation-Data	93.65	90.21	86.3	91.02	89.67	89.27	85.66

All the algorithms increased in some capacity. Yellow denotes most accurate, Green denotes Second, and Blue denotes third. JRIP, IBK, and Logistic did are algorithms I would drop from the model building process based on the lack of accuracy for any one dataset. In my opinion, Simple Logistic improved the most based on its accuracy and the fact that the iterations improved the model. Moving forward, Naive Bayes, ANN, and Simple Logistic are the initial top performers for accuracy.

Conclusion

Accuracy is a KPI, but root mean square error (RMSE) and F-Measure also form two other dimensions for the 3-dimensional KPI analysis. I view the RMSE as describing the goodness of fit for a model. This analysis falls along the lines of precision. The process of analyzing RMSE per algorithm falls along the same process as the analysis for accuracy. With each KPI, the best fitting algorithm for a dataset will be ranked with a number 3 for being the best fitting, 2 for second best fitting, and 1 for third. The worst fitting algorithm will be ranked with a -3, the second

Naive Bayes

```
'Wine With Correlated Dat(100) 94.42 |

'Wine with all data inclu(100) 97.46 |

'Wine- Data with all thre(100) 97.41 |

'Wine- Variables for firs(100) 86.37 |

'Wine- Data of Second Dim(100) 91.89 |

Wine-NoCorrelation-Data (100) 92.29 |
```

To the left, one can see the Naïve Bayes accuracy before iterations.

```
Dataset (1) bayes.Naive

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

'Wine with all data inclu (50) 97.41 |

'Wine- Data with all thre (50) 97.86 |

'Wine With Correlated Dat (50) 100.00 |

'Wine- Variables for firs (50) 85.40 |

'Wine- Data of Second Dim (50) 92.14 |

Wine-NoCorrelation-Data (50) 93.42 |
```

The final iterations for the model are shown to the left.

```
Dataset (1) bayes.Nai
---------------------------------------
'Wine with all data inclu (50) 0.07 |

'Wine- Data with all thre (50) 0.07 |

'Wine With Correlated Dat (50) 0.03 |

'Wine- Variables for firs (50) 0.28 |

'Wine- Data of Second Dim (50) 0.23 |

Wine-NoCorrelation-Data (50) 0.18 |
```

After the final iterations, the RMSE is shown to the left. For the most part, the RMSE reflects a solid fit in line with what was shown with the accuracy output. Moving forward, I will compare the RMSE for multiple algorithms for a frame of reference as opposed to assessing in isolation.

Wine with Correlated Data

The dataset ‘Wine with Correlated Data’ had perfect classification across all algorithms. The next step to find the best algorithm is to analyze the RMSE for goodness of fit.

```
Dataset (1) bayes.Nai
---------------------------------------
'Wine with all data inclu (50) 0.07 |

'Wine- Data with all thre (50) 0.07 |

'Wine With Correlated Dat (50) 0.03 |

'Wine- Variables for firs (50) 0.28 |

'Wine- Data of Second Dim (50) 0.23 |

Wine-NoCorrelation-Data (50) 0.18 |
```

```
J48

'Wine with all data inclu(150) 0.16 |

'Wine- Data with all thre(150) 0.16 |

'Wine With Correlated Dat(150) 0.00 |

'Wine- Variables for firs(150) 0.28 |

'Wine- Data of Second Dim(150) 0.24 |

Wine-NoCorrelation-Data (150) 0.21 |
IBK
---------------------------------------
'Wine with all data inclu(100) 0.14 |

'Wine- Data with all thre(100) 0.12 |

'Wine With Correlated Dat(100) 0.03 |

'Wine- Variables for firs(100) 0.35 |

'Wine- Data of Second Dim(100) 0.21 |

Wine-NoCorrelation-Data (100) 0.28 |
```

```
ANN
---------------------------------------
'Wine with all data inclu(150) 0.07 |

'Wine- Data with all thre(150) 0.06 |

'Wine With Correlated Dat(150) 0.00 |

'Wine- Variables for firs(150) 0.24 |

'Wine- Data of Second Dim(150) 0.20 |

Wine-NoCorrelation-Data (150) 0.20 |
```

```
Logistic Regression
---------------------------------------
Wine with all data inclu(100) 0.08 |

'Wine- Data with all thre(100) 0.09 |

'Wine With Correlated Dat(100) 0.00 |

'Wine- Variables for firs(100) 0.30 |

'Wine- Data of Second Dim(100) 0.23 |

Wine-NoCorrelation-Data (100) 0.21 | |
```

```
Simple Logistic
---------------------------------------
'Wine with all data inclu(100)   0.07 |

'Wine- Data with all thre(100)   0.06 |

'Wine With Correlated Dat(100)   0.00

'Wine- Variables for firs(100)   0.28 |

'Wine- Data of Second Dim(100)
0.20 |

Wine-NoCorrelation-Data  (100)   0.22
```

```
JRIP
---------------------------------------
'Wine with all data inclu(100) 0.18 |

'Wine- Data with all thre(100) 0.19 |

'Wine With Correlated Dat(100) 0.00 |

'Wine- Variables for firs(100) 0.29 |

'Wine- Data of Second Dim(100) 0.28 |

Wine-NoCorrelation-Data (100) 0.28 | |
```

From the seven different Algorithms, the rank is:

1. JRIP, Simple Logistic, Logistic Regression, ANN, J48

6. IBK, Naïve Bayes.

From ranking the RMSE, I have eliminated two algorithms and the F-measure will be utilized to further analyze the top five.

Wine with all Data

```
Dataset (1) bayes.Nai
--------------------------------------
'Wine with all data inclu (50) 0.07 |

'Wine- Data with all thre (50) 0.07 |

'Wine With Correlated Dat (50) 0.03 |

'Wine- Variables for firs (50) 0.28 |

'Wine- Data of Second Dim (50) 0.23 |

Wine-NoCorrelation-Data (50) 0.18 |
```

```
J48

'Wine with all data inclu(150) 0.16 |

'Wine- Data with all thre(150) 0.16 |

'Wine With Correlated Dat(150) 0.00 |

'Wine- Variables for firs(150) 0.28 |

'Wine- Data of Second Dim(150) 0.24 |

Wine-NoCorrelation-Data (150) 0.21 |
```

```
IBK
--------------------------------------
'Wine with all data inclu(100) 0.14 |

'Wine- Data with all thre(100) 0.12 |

'Wine With Correlated Dat(100) 0.03 |

'Wine- Variables for firs(100) 0.35 |

'Wine- Data of Second Dim(100) 0.21 |

Wine-NoCorrelation-Data (100) 0.28 |
```

```
ANN
--------------------------------------
'Wine with all data inclu(150) 0.07 |

'Wine- Data with all thre(150) 0.06 |

'Wine With Correlated Dat(150) 0.00 |

'Wine- Variables for firs(150) 0.24 |

'Wine- Data of Second Dim(150) 0.20 |

Wine-NoCorrelation-Data (150) 0.20 |
```

```
Logistic Regression
--------------------------------------
Wine with all data inclu(100) 0.08 |

'Wine- Data with all thre(100) 0.09 |

'Wine With Correlated Dat(100) 0.00 |

'Wine- Variables for firs(100) 0.30 |

'Wine- Data of Second Dim(100) 0.23 |

Wine-NoCorrelation-Data (100) 0.21 | |
```

```
Simple Logistic
--------------------------------------
'Wine with all data inclu(100) 97.58 |

'Wine- Data with all thre(100) 98.14 |

'Wine With Correlated Dat(100) 100.00

'Wine- Variables for firs(100) 83.94 |

'Wine- Data of Second Dim(100) 90.60 |

Wine-NoCorrelation-Data (100) 89.27 | |
JRIP
--------------------------------------
'Wine with all data inclu(100) 0.18 |

'Wine- Data with all thre(100) 0.19 |

'Wine With Correlated Dat(100) 0.00 |

'Wine- Variables for firs(100) 0.29 |

'Wine- Data of Second Dim(100) 0.28 |

Wine-NoCorrelation-Data (100) 0.28 | |
```

The ranking for RMSE:

1. Naïve Bayes, ANN

3. Logistic Regression

4. IBK

5. J48

6. JRIP

Wine Data with all Three

```
Dataset (1) bayes.Nai
--------------------------------------
'Wine with all data inclu (50) 0.07 |

'Wine- Data with all thre (50) 0.07 |

'Wine With Correlated Dat (50) 0.03 |

'Wine- Variables for firs (50) 0.28 |

'Wine- Data of Second Dim (50) 0.23 |

Wine-NoCorrelation-Data (50) 0.18 |
```

```
J48

'Wine with all data inclu(150) 0.16 |

'Wine- Data with all thre(150) 0.16 |

'Wine With Correlated Dat(150) 0.00 |

'Wine- Variables for firs(150) 0.28 |

'Wine- Data of Second Dim(150) 0.24 |

Wine-NoCorrelation-Data (150) 0.21 |
```

```
Logistic Regression
--------------------------------------
Wine with all data inclu(100) 0.08 |

'Wine- Data with all thre(100) 0.09 |

'Wine With Correlated Dat(100) 0.00 |

'Wine- Variables for firs(100) 0.30 |

'Wine- Data of Second Dim(100) 0.23 |

Wine-NoCorrelation-Data (100) 0.21 | |
```

```
ANN
--------------------------------------
'Wine with all data inclu(150) 0.07 |

'Wine- Data with all thre(150) 0.06 |

'Wine With Correlated Dat(150) 0.00 |

'Wine- Variables for firs(150) 0.24 |

'Wine- Data of Second Dim(150) 0.20 |

Wine-NoCorrelation-Data (150) 0.20 |
IBK
---------------------------------------
'Wine with all data inclu(100) 0.14 |

'Wine- Data with all thre(100) 0.12 |

'Wine With Correlated Dat(100) 0.03 |

'Wine- Variables for firs(100) 0.35 |

'Wine- Data of Second Dim(100) 0.21 |

Wine-NoCorrelation-Data (100) 0.28 |
```

```
Simple Logistic
---------------------------------------
'Wine with all data inclu(100) 97.58 |

'Wine- Data with all thre(100) 98.14 |

'Wine With Correlated Dat(100) 100.00

'Wine- Variables for firs(100) 83.94 |

'Wine- Data of Second Dim(100) 90.60 |

Wine-NoCorrelation-Data (100) 89.27 | |
```

```
JRIP
---------------------------------------
'Wine with all data inclu(100) 0.18 |

'Wine- Data with all thre(100) 0.19 |

'Wine With Correlated Dat(100) 0.00 |

'Wine- Variables for firs(100) 0.29 |

'Wine- Data of Second Dim(100) 0.28 |

Wine-NoCorrelation-Data (100) 0.28 | |
```

1. ANN

2. Naïve Bayes

3. Logistic Regression

4. IBK

5. J48

6. JRIP

<u>Wine Variables for First (Green Above)</u>

1. ANN

2. J48, Naïve Bayes

4. JRIP

5. Logistic Regression

6. JRip

Wine-Data of Second Dim (in Yellow)

```
Dataset (1) bayes.Nai
--------------------------------------
'Wine with all data inclu (50) 0.07 |

'Wine- Data with all thre (50) 0.07 |

'Wine With Correlated Dat (50) 0.03 |

'Wine- Variables for firs (50) 0.28 |

'Wine- Data of Second Dim (50) 0.23 |

Wine-NoCorrelation-Data (50) 0.18 |
```

```
J48

'Wine with all data inclu(150) 0.16 |

'Wine- Data with all thre(150) 0.16 |

'Wine With Correlated Dat(150) 0.00 |

'Wine- Variables for firs(150) 0.28 |

'Wine- Data of Second Dim(150) 0.24 |

Wine-NoCorrelation-Data (150) 0.21 |
```

```
IBK
--------------------------------------
'Wine with all data inclu(100) 0.14 |

'Wine- Data with all thre(100) 0.12 |

'Wine With Correlated Dat(100) 0.03 |

'Wine- Variables for firs(100) 0.35 |

'Wine- Data of Second Dim(100) 0.21 |

Wine-NoCorrelation-Data (100) 0.28 |
```

```
ANN
--------------------------------------
'Wine with all data inclu(150) 0.07 |

'Wine- Data with all thre(150) 0.06 |

'Wine With Correlated Dat(150) 0.00 |

'Wine- Variables for firs(150) 0.24 |

'Wine- Data of Second Dim(150) 0.20 |

Wine-NoCorrelation-Data (150) 0.20 |
```

```
Logistic Regression
--------------------------------------
Wine with all data inclu(100) 0.08 |

'Wine- Data with all thre(100) 0.09 |

'Wine With Correlated Dat(100) 0.00 |

'Wine- Variables for firs(100) 0.30 |

'Wine- Data of Second Dim(100) 0.23 |

Wine-NoCorrelation-Data (100) 0.21 | |
```

```
Simple Logistic
--------------------------------------
'Wine with all data inclu(100)   0.07 |

'Wine- Data with all thre(100)   0.06 |

'Wine With Correlated Dat(100)   0.00 |

'Wine- Variables for firs(100)   0.28 |

'Wine- Data of Second Dim(100)   0.20 |

Wine-NoCorrelation-Data  (100)   0.22
```

```
JRIP
---------------------------------------
'Wine with all data inclu(100) 0.18 |

'Wine- Data with all thre(100) 0.19 |

'Wine With Correlated Dat(100) 0.00 |

'Wine- Variables for firs(100) 0.29 |

'Wine- Data of Second Dim(100) 0.28 |

Wine-NoCorrelation-Data (100) 0.28 | |
```

1. ANN

2. IBK

3. Naïve Bayes, Logistic Regression

5. J48

6.JRIP

<u>Wine –No Correlation (in Green)</u>

1. Naïve Bayes

2. ANN

3. J48, Logistic Regression

5. IBK, JRip

The final matrice can be seen below:

RMSE							
Wine with all data inclu	0.07	0.16	0.14	0.07	0.08	0.07	0.18
Wine- Data with all thre	0.07	0.16	0.12	0.06	0.09	0.06	0.19
'Wine With Correlated Dat	0.03	0	0.03	0	0	0	0
Wine- Variables for firs	0.28	0.28	0.35	0.24	0.3	0.28	0.29
'Wine- Data of Second Dim	0.23	0.24	0.21	0.2	0.23	0.2	0.28
Wine-NoCorrelation-Data	0.18	0.21	0.28	0.2	0.21	0.22	0.28

F-Measure

Accuracy is a KPI, but root mean square error (RMSE) and F-Measure also form two other dimensions for the 3-dimensional KPI analysis. I view the F-Measure as describing the goodness of fit for a model and accuracy. This KPI combines both accuracy and precision. The process of analyzing RMSE per algorithm falls along the same process as the analysis for accuracy. With each KPI, the best fitting algorithm for a dataset will be ranked with a number 6 for being the best fitting, 5 for second best fitting, and 4 for third, 3 for fourth, 2 for fifth, and 1 for sixth. Given that each ranking carries a numerical value, at the end of the analysis a total summation matrice will be created that weight each KPI equally. From this 3 dimensional analysis, one will gather one numerical value that represents the best algorithm for a specific dataset.

Wine with all Data F-Measure in Yellow

Naïve Bayes
'Wine with all data inclu (50) 0.99 |

'Wine- Data with all thre (50) 0.99 |

'Wine With Correlated Dat (50) 1.00 |

'Wine- Variables for firs (50) 0.85 |

'Wine- Data of Second Dim (50) 0.93 |

Wine-NoCorrelation-Data (50) 0.94 |

J48
'Wine with all data inclu(150) 0.95 |

'Wine- Data with all thre(150) 0.95 |

'Wine With Correlated Dat(150) 1.00 |

'Wine- Variables for firs(150) 0.87 |

'Wine- Data of Second Dim(150) 0.90 |

Wine-NoCorrelation-Data (150) 0.90 |

IBK
Wine with all data inclu(100) 0.96 |

'Wine- Data with all thre(100) 0.97 |

'Wine With Correlated Dat(100) 1.00 |

'Wine- Variables for firs(100) 0.75 |

'Wine- Data of Second Dim(100) 0.94 |

Wine-NoCorrelation-Data (100) 0.89 |

ANN
Wine with all data inclu(150) 0.99 |

'Wine- Data with all thre(150) 1.00 |

'Wine With Correlated Dat(150) 1.00 |

'Wine- Variables for firs(150) 0.85 |

'Wine- Data of Second Dim(150) 0.93 |

Wine-NoCorrelation-Data (150) 0.91 |

Logistic Regression

'Wine with all data inclu(100) 0.98 |

'Wine- Data with all thre(100) 0.98 |

'Wine With Correlated Dat(100) 1.00 |

'Wine- Variables for firs(100) 0.74 |

'Wine- Data of Second Dim(100) 0.91 |

Wine-NoCorrelation-Data (100) 0.91 |

Simple Logistic

'Wine with all data inclu(100) 0.99 |

'Wine- Data with all thre(100) 0.99 |

'Wine With Correlated Dat(100) 1.00 |

'Wine- Variables for firs(100) 0.78 |

'Wine- Data of Second Dim(100) 0.93 |

Wine-NoCorrelation-Data (100) 0.88 |

JRIP

Wine with all data inclu(100) 0.93 |

'Wine- Data with all thre(100) 0.92 |

'Wine With Correlated Dat(100) 1.00 |

'Wine- Variables for firs(100) 0.84 |

'Wine- Data of Second Dim(100) 0.88 |

Wine-NoCorrelation-Data (100) 0.83 |

1.

Wine Variables for First F-Measure in Yellow

Naïve Bayes

'Wine with all data inclu (50) 0.99 |

'Wine- Data with all thre (50) 0.99 |

'Wine With Correlated Dat (50) 1.00 |

'Wine- Variables for firs (50) 0.85 |

'Wine- Data of Second Dim (50) 0.93 |

Wine-NoCorrelation-Data (50) 0.94 |

J48

'Wine with all data inclu(150) 0.95 |

'Wine- Data with all thre(150) 0.95 |

'Wine With Correlated Dat(150) 1.00 |

'Wine- Variables for firs(150) 0.87 |

'Wine- Data of Second Dim(150) 0.90 |

Wine-NoCorrelation-Data (150) 0.90 |

IBK

Wine with all data inclu(100) 0.96 |

'Wine- Data with all thre(100) 0.97 |

'Wine With Correlated Dat(100) 1.00 |

'Wine- Variables for firs(100) 0.75 |

'Wine- Data of Second Dim(100) 0.94 |

Wine-NoCorrelation-Data (100) 0.89 |

ANN

Wine with all data inclu(150) 0.99 |

'Wine- Data with all thre(150) 1.00 |

'Wine With Correlated Dat(150) 1.00 |

'Wine- Variables for firs(150) 0.85 |

'Wine- Data of Second Dim(150) 0.93 |

Wine-NoCorrelation-Data (150) 0.91 |

Logistic Regression

'Wine with all data inclu(100) 0.98 |

'Wine- Data with all thre(100) 0.98 |

'Wine With Correlated Dat(100) 1.00 |

'Wine- Variables for firs(100) 0.74 |

'Wine- Data of Second Dim(100) 0.91 |

Wine-NoCorrelation-Data (100) 0.91 |

Simple Logistic

'Wine with all data inclu(100) 97.58 |

'Wine- Data with all thre(100) 98.14 |

'Wine With Correlated Dat(100) 100.00

'Wine- Variables for firs(100) 83.94 |

'Wine- Data of Second Dim(100) 90.60 |

Wine-NoCorrelation-Data (100) 89.27 |

JRIP

Wine with all data inclu(100) 0.93 |

'Wine- Data with all thre(100) 0.92 |

'Wine With Correlated Dat(100) 1.00 |

'Wine- Variables for firs(100) 0.84 |

'Wine- Data of Second Dim(100) 0.88 |

Wine-NoCorrelation-Data (100) 0.83 |

Final Matrice for F-Measure

F-Measure							
Wine with all data inclu	0.99	0.95	0.96	0.99	0.98	0.99	0.93
Wine- Data with all thre	0.99	0.95	0.97	1	0.98	0.99	0.92
'Wine With Correlated Dat	1	1	1	1	1	1	1
Wine- Variables for firs	0.85	0.87	0.75	0.85	0.74	0.78	0.84
'Wine- Data of Second Dim	0.93	0.9	0.94	0.93	0.91	0.93	0.88
Wine-NoCorrelation-Data	0.94	0.9	0.89	91	0.91	0.88	0.83

Based on the percent's above for all three KPI's, I will rank all the algorithms 7-1. 7 equals the best ranking…and 1 equals the worst ranking. If it is a tie the numerical value is divided by the rankings and n-number of tied algorithms.

F-Measure	Naïve Bayes	J48	IBK	ANN	LR	Simple L	JRIP
Wine with all data inclu	6	2	3	6	4	6	1
Wine- Data with all three	5.5	2	3	7	4	5.5	1
Wine With Correlated Dat	4	4	4	4	4	4	4
Wine- Variables for firs	5.5	7	1	5.5	2	3	4
Wine- Data of Second Dim	5	2	7	5	3	5	1
Wine-NoCorrelation-Data	7	4	3	6	5	2	1
Root Mean Square	Naïve Bayes	J48	IBK	ANN	LR	Simple L	JRIP
Wine with all data inclu	6	2	3	6	4	6	1
Wine- Data with all three	5	2	3	6.5	4	6.5	1
Wine With Correlated Dat	1.5	5	1.5	5	5	5	5
Wine- Variables for firs	4.6	4.6	1	7	2	4.6	3
Wine- Data of Second Dim	3.5	2	5	6.5	3.5	6.5	1
Wine-NoCorrelation-Data	7	4.5	1.5	6	4.5	3	1.5
Accuracy	Naïve Bayes	J48	IBK	ANN	LR	Simple L	JRIP
Wine with all data inclu	7	2	3	6	4	5	1
Wine- Data with all three	6	2	3	7	4	5	1
Wine With Correlated Dat	4	4	4	4	4	4	4

Wine- Variables for firs	3	6	1	7	5	4	2
Wine- Data of Second Dim	3	2	7	5	4	6	1
Wine-NoCorrelation-Data	7	5	2	6	4	3	1
Final Ranking (Sum)	Naïve Bayes	J48	IBK	ANN	LR	Simple L	JRIP
Wine with all data inclu	19	6	9	18	12	17	3
Wine- Data with all three	16.5	6	9	20.5	12	17	3
Wine With Correlated Dat	9.5	13	9.5	13	13	13	13
Wine- Variables for firs	13.1	17.6	3	19.5	9	11.6	9
Wine- Data of Second Dim	11.5	6	19	16.5	10.5	17.5	3
Wine-NoCorrelation-Data	21	13.5	6.5	18	13.5	8	3.5

Taking into account all three aspects of Accuracy, Root Mean Square Error, and F-Measure equally, the final algorithm selection can be seen below (Ranking Order).

Wine All Data: Naive Bayes, ANN, Simple L. There is quite a distance from fourth to seventh: LR, IBK, J48, JRIP.

Wine-Data with all three: ANN –Solid first place, Simple L, Naive Bayes. There is quite a distance from fourth to seventh: LR, IBK, J48, JRIP.

Wine with Correlated Data: Tie between ANN, Simple L, LR, JRIp, J48. The last two are IBK and Naive Bayes.

Wine- Variables for first: ANN, J48, Naive Bayes, Simple L, LR and JRIP, IBK.

Wine- Data of Second Dim: IBK, Simple L, ANN, Naive Bayes, LR, J48, JRIP.

Wine-NoCorrelation-Data: Naive Bayes, ANN, LR and J48, Simple L, IBK, JRIP.

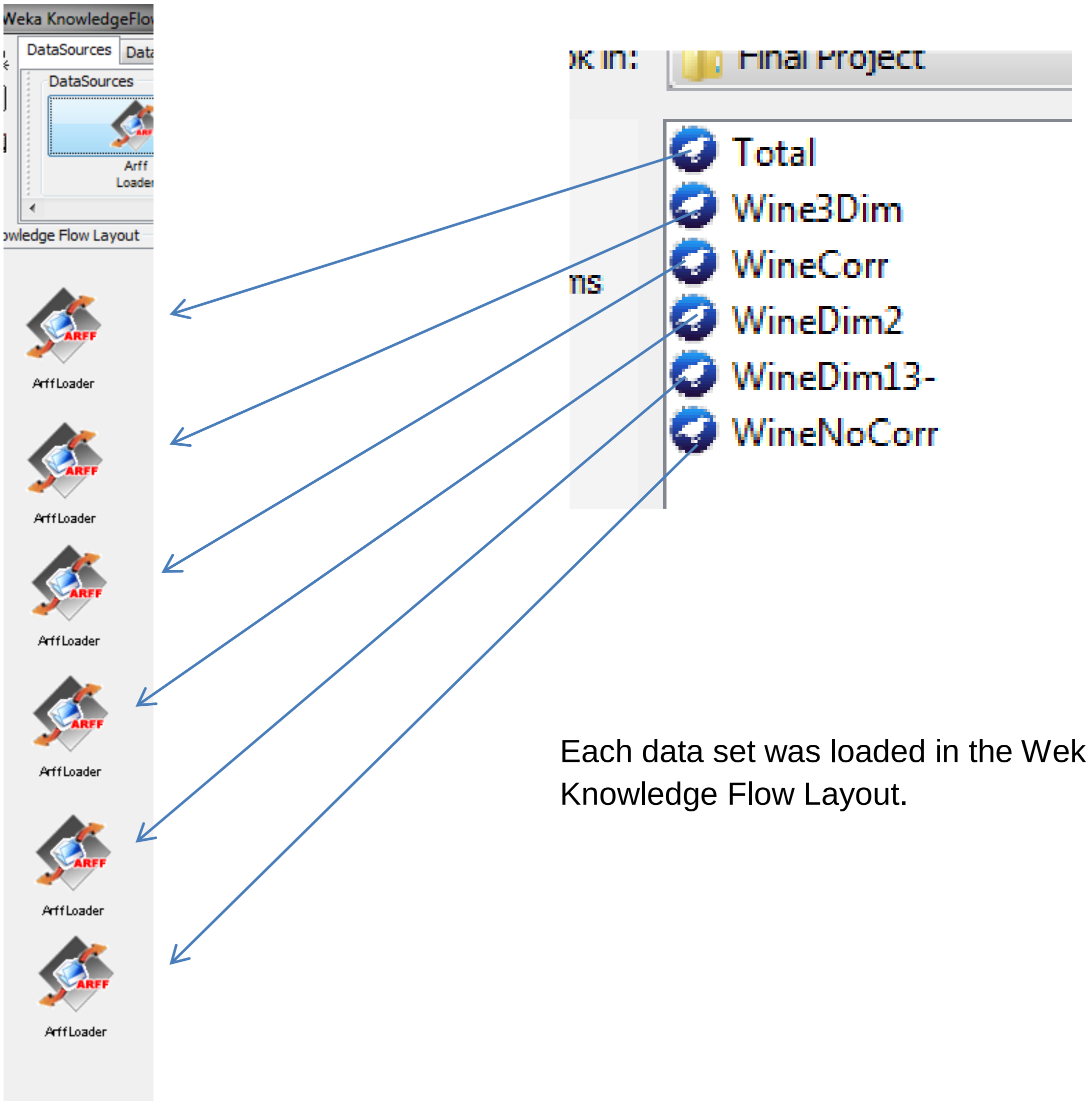
If one was looking for one algorithm the total sum for all the datasets are:

Final Ranking (Sum)	Naïve Bayes	J48	IBK	ANN	LR	Simple L	JRIP
Total Sum	90.6	62.1	56	105.5	70	84.1	34.5

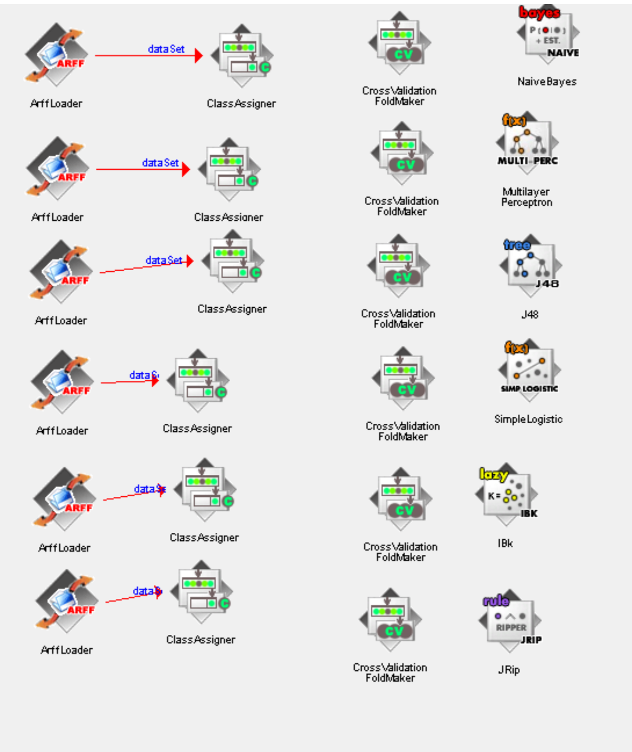
ANN is the best, followed by Naive Bayes, Simple L, LR, J48, IBK, JRIP.

Knowledge Flow

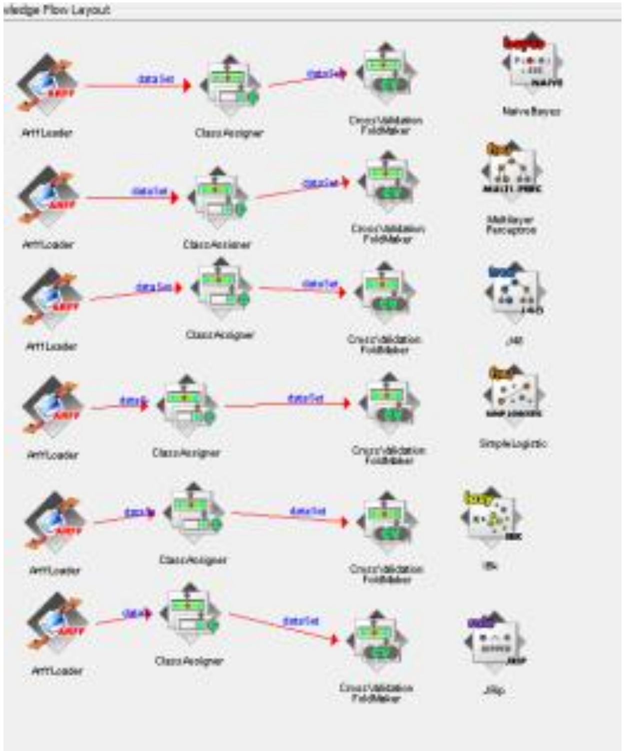
One of the parameters of this project is to utilize the Knowledge Flow application within Weka. This application allows one to engineer and reverse engineer the algorithmic process. Given that this application has not been used throughout this course, I will highlight and demonstrate a basic overview of the application for the top three performing algorithms. Throughout my demonstration, I will provide screenshots for a virtual walkthrough.



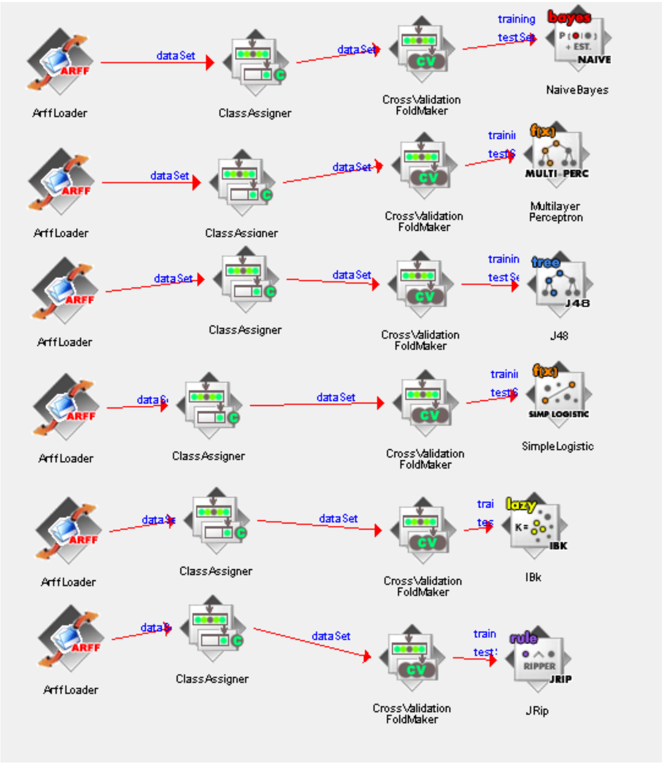
Each data set was loaded in the Weka Knowledge Flow Layout.



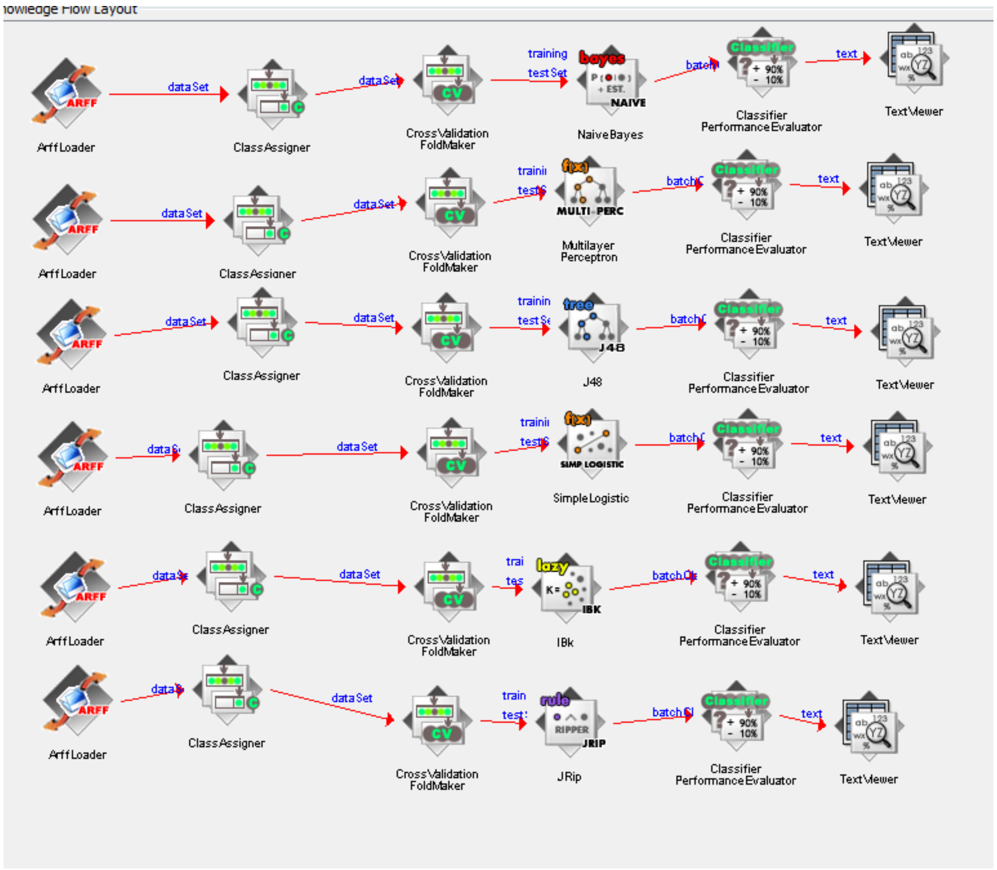
Each dataset was configured to the class assigner, which was based on the class.



Each model was configured with the cross validation fold maker option.



I paired the top performing algorithm with the corresponding dataset.


nowledge Flow Layout
ArffLoader
dataSet
ClassAssigner
dataSet
CrossValidation FoldMaker
training
testSet
NaiveBayes
batch
Classifier PerformanceEvaluator
text
TextViewer
ArffLoader
ClassAssigner
CrossValidation FoldMaker
Multilayer Perceptron
Classifier PerformanceEvaluator
TextViewer
ArffLoader
ClassAssigner
CrossValidation FoldMaker
J48
Classifier PerformanceEvaluator
TextViewer
ArffLoader
ClassAssigner
CrossValidation FoldMaker
SimpleLogistic
Classifier PerformanceEvaluator
TextViewer
ArffLoader
ClassAssigner
CrossValidation FoldMaker
IBk
Classifier PerformanceEvaluator
TextViewer
ArffLoader
ClassAssigner
CrossValidation FoldMaker
JRip
Classifier PerformanceEvaluator
TextViewer

I attached the Classifer evaluator to each algorithm along with the text view.

The output from the models can be seen below:

```
=== Evaluation result ===

Scheme: NaiveBayes
Relation: Wine with all data included


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
  0  0 48 |  c = C
```

```
=== Evaluation result ===

Scheme: MultilayerPerceptron
Options: -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: Wine- Data with all three dimension


Correctly Classified Instances        177               99.4382 %
Incorrectly Classified Instances        1               0.5618 %
Kappa statistic                    0.9915
Mean absolute error                    0.0164
Root mean squared error                  0.0791
Relative absolute error                3.7313 %
Root relative squared error             16.8919 %
Total Number of Instances              178

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0         1         1         1          1       A
                 0.986     0         1        0.986     0.993      1       B
                 1        0.008      0.98      1        0.99       1       C
Weighted Avg.    0.994     0.002      0.994     0.994     0.994      1

=== Confusion Matrix ===

  a  b  c   <-- classified as
 59  0  0 |  a = A
  0 70  1 |  b = B
  0  0 48 |  c = C
```

```
=== Evaluation result ===

Scheme: J48
Options: -C 0.25 -M 2
Relation: Wine With Correlated Data


Correctly Classified Instances          49              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                     1
Mean absolute error                     0
Root mean squared error                    0
Relative absolute error                 0      %
Root relative squared error               0      %
Total Number of Instances                49

=== Detailed Accuracy By Class ===

             TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
               1         0          1          1         1          ?        A
               0         0          0          0         0          ?        B
               0         0          0          0         0          ?        C
Weighted Avg.    1         0          1          1         1          0

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  0  0 |  a = A
  0  0  0 |  b = B
  0  0  0 |  c = C
```

```
=== Evaluation result ===

Scheme: SimpleLogistic
Options: -I 0 -M 500 -H 50 -W 0.0
Relation: Wine- Data of Second Dimension


Correctly Classified Instances       162              91.0112 %
Incorrectly Classified Instances      16               8.9888 %
Kappa statistic                    0.8629
Mean absolute error                  0.087
Root mean squared error                0.2049
Relative absolute error             19.825  %
Root relative squared error           43.7366 %
Total Number of Instances            178

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.949     0.034      0.933     0.949     0.941      0.995    A
                 0.93      0.075      0.892     0.93      0.91       0.984    B
                 0.833     0.031      0.909     0.833     0.87       0.979    C
Weighted Avg.    0.91      0.049      0.91      0.91      0.91       0.987

=== Confusion Matrix ===

  a  b  c   <-- classified as
 56  2  1 |  a = A
  2 66  3 |  b = B
  2  6 40 |  c = C
```

```
=== Evaluation result ===

Scheme: IBk
Options: -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A
\"weka.core.EuclideanDistance -R first-last\""
Relation: Wine- Variables for first dimension


Correctly Classified Instances        146               82.0225 %
Incorrectly Classified Instances       32               17.9775 %
Kappa statistic                    0.7288
Mean absolute error                    0.1258
Root mean squared error                    0.3431
Relative absolute error                28.6572 %
Root relative squared error            73.2316 %
Total Number of Instances              178

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.831     0.151      0.731     0.831     0.778      0.824    A
                 0.704     0.103      0.82      0.704     0.758      0.782    B
                 0.979     0.023      0.94      0.979     0.959      0.978    C
Weighted Avg.    0.82      0.097      0.823     0.82      0.819      0.849

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49 10  0 |  a = A
 18 50  3 |  b = B
  0  1 47 |  c = C
```

```
=== Evaluation result ===

Scheme: JRip
Options: -F 3 -N 2.0 -O 2 -S 1
Relation: Wine-NoCorrelation-Data


Correctly Classified Instances        152               85.3933 %
Incorrectly Classified Instances       26               14.6067 %
Kappa statistic                     0.7766
Mean absolute error                   0.1166
Root mean squared error                 0.3003
Relative absolute error              26.5559 %
Root relative squared error            64.0919 %
Total Number of Instances             178

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.814     0.076      0.842     0.814     0.828      0.901    A
                 0.901     0.131      0.821     0.901     0.859      0.911    B
                 0.833     0.023      0.93      0.833     0.879      0.937    C
Weighted Avg.    0.854     0.083      0.857     0.854     0.854      0.915

=== Confusion Matrix ===

  a  b  c   <-- classified as
 48 10  1 |  a = A
  5 64  2 |  b = B
  4  4 40 |  c = C
```

Utilizing Weka Knowledge Flow Enviornment is another approach to analyze how the variables and algorithms interact with different iterations on the datasets. In hindsight, I would have enjoyed learning this application earlier in the class and applying it on certain homework assignments.

Conclusion

The overall objective of this exploratory data analysis (EDA) was to analyze the relationship between thirteen variables throughout six different data sets utilizing an array of different algorithms. At the end of the EDA, a concluding recommendation was given based on the best suited algorithm that best classifies and explains the datasets as a whole, as well as individually.

Management supplied the data, and was aggregated into six different sets. Each set was made up of 178 instances, and these instances are the same throughout each set. There are no unique instances to a particular data set. In an effort to retain the efficacy throughout the datasets, modifications were not made preliminarily such that the variables retained their objective nature. In an effort to simulate different dataset sizes, the fold option was expansively used to explore the size dimension for each algorithm. This was done in effort to capture the dynamic relationship of the individual algorithms. Scatter plots and correlation coefficients were used to study the nature of the relationships between the variables. Of the thirteen variables, Alcohol, Total Phenols, Flavanoids, 0D280_OD315, and Proline were relatively correlated with one another and proved to influence the accuracy and precision of the different algorithms.

After assessing the data and algorithms, the following discoveries and insights are listed below:

Naïve Bayes: This algorithm was top ranking for the dataset that included all variables along with the no correlated dataset. While this algorithm is rather simple in nature, it proved to work well with less folds (bigger datasets). I would recommend Naïve Bayes as the top algorithm used in the EDA based on its KPI analysis, as well as its robust nature.

Artificial Neural Network: As a whole, this algorithm had the highest points for the KPI metrics. In my opinion, this algorithm worked well on smaller datasets (given the fold options) but would struggle in a 'big data' environment.

Simple Logistic: This algorithm was not used throughout the course, but proved to have quite a few iteration options that conformed well to the data. Given the ternary response variable of the variable classification, this algorithm was well fit to the data. A concern for this algorithm is being over-fit and I would recommend further testing on larger datasets.

Logistic Regression: Similar to the Simple Logistic algorithm, this algorithm proved to be less versatile to multiple datasets and lacked the iteration options of Simple Logistic. In a larger dataset environment, I would expect this model to perform better based on the lack of over fitting throughout the iteration process.

J48: This algorithm was one of the bottom performing three algorithms. The continuous variables along with the size of the datasets worked against the pros of the J48.

IBK: The Second Dimension dataset proved to be a hard dataset for the overall algorithms to classify well. IBK had the highest ranking between all three KPI's for this dataset. Other than the Second Dimension dataset, IBK performed poorly and I would recommend dropping it.

JRIP: Across all the datasets, JRIP performed very poorly. The datasets were not structured in a way that catered to this algorithms strength. I would not see this algorithm performing well with either large or small datasets. The algorithms listed above were expansive of the techniques we learned in this class and spanned many different mathematical classifiers.

Bibliography

Ratner, Bruce. *Statistical and Machine-Learning Data Mining* . 2nd ed. Boca Raton, FL: CRC Press, 2012. Print.

Wittwer, Glyn, and Jeremy Rothfield. "Projecting the world wine market from 2003 to 2010." *Australasian Agribusiness Review* 13.1 (2005): 21. *Centre of Policy Studies, Monash University*. Web. 18 Feb. 2013.

http://gautam.lis.illinois.edu/monkmiddleware/public/analytics/decisiontree.html

Witten IH, Frank E. *Data Mining: Practical Machine Learning Tools and Techniques*. Second edition, 2005. Morgan Kaufmann.

Han J, Kamber M. *Data Mining: Concepts and Techniques*. Second edition, 2006. Morgan Kaufmann.

http://wiki.pentaho.com/display/DATAMINING/JRip