Brad Ware
Unsupervised Learning & Dimensionality
CS 4641 Project 3

Datasets

In project 1, one of my classification problems was to identify the quality of an instance of *white* wine. Some of the attributes included pH, alcohol content, and levels of acidity. The data was modified so that there were three qualities: bad, good, and great. This same dataset will be used again for *k*-means clustering and EM, and will also undergo more dimensionality reduction algorithms.

I will also be introducing a new dataset, the voting results from 1984 US Congressional campaign. This dataset has numerous attributes of boolean values, some including standpoints on religion, education, and foreign affairs. There are two classes it is trying to predict: democrat and republican.

Clustering – Raw Data

White Wine

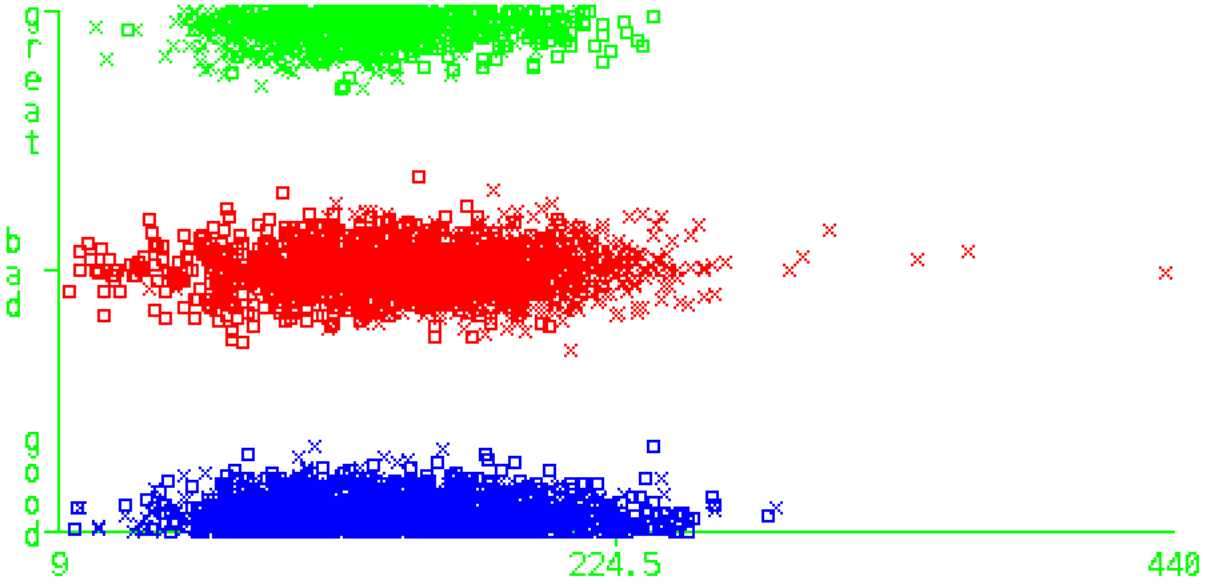
For the wine data, multiple values of k were used. First, after researching online, I found that a good start for *k* is $\sqrt{n/2}$ where *n* is the number of instances. After that, I ran iterations where *k* was 10 and then *k* = 3, and *k* = 2. The optimal error value was when *k* = 3, and it was **52%**. I tried iterations using the Euclidean and Manhattan distance, but there was minimal difference. I have included statistics about each cluster and their attributes. As you can see, there is a direct correlation between great quality of wine and the amount of total sulfur dioxide in the drink. Cluster 1, which had the most 'great' wine instances, had the lowest sulfur dioxide and residual sugar, but the highest alcohol content.

Cluster	0	1	2
Bad	1052	239	349
Good	817	751	630
Great	149	633	278

Attribute	Full Data	Cluster 0	Cluster 1	Cluster 2
Sulfur dioxide	138.3607	160.057	112.7911	136.5438
Alcohol	10.5143	9.4543	11.8973	10.4301
Residual Sugar	6.3914	9.4663	4.1085	4.4027

For EM, the optimal *k* was 3, and it had an error rate of **52%** again. It ran a total of 52 iterations to get these results, and it took 1.65 seconds to build. The log of likelihood was -3.98. EM was able to show even more clearly that sulfur dioxide definitely plays a role in the quality of wine. The bad cluster clearly had a higher rate, while the great cluster majority had a much lower average.

Sulfur Dioxide Content vs. Quality of Wine



Attribute	Cluster 0 (Good)	Cluster 1 (Great)	Cluster2 (Bad)
Fixed Acidity	6.972	6.5911	6.9528
Volatile Acidity	0.2735	0.2785	0.2843
Citric Acid	0.3201	0.3176	0.3691
Residual Sugar	5.1505	2.6956	11.6223
Alcohol	10.3817	11.8924	9.3599

Voting

The same process for picking *k* was used on the voting dataset. However, the optimum value for eliminating error was when *k* =2, as the error rate was 10% higher by having 3 clusters. The final error rate was **16%**. Many key issues divided the parties including crime, religion in school, and budget approval. Some of the main one's are included below (Cluster 0 Republican, Cluster 1 Democrat).

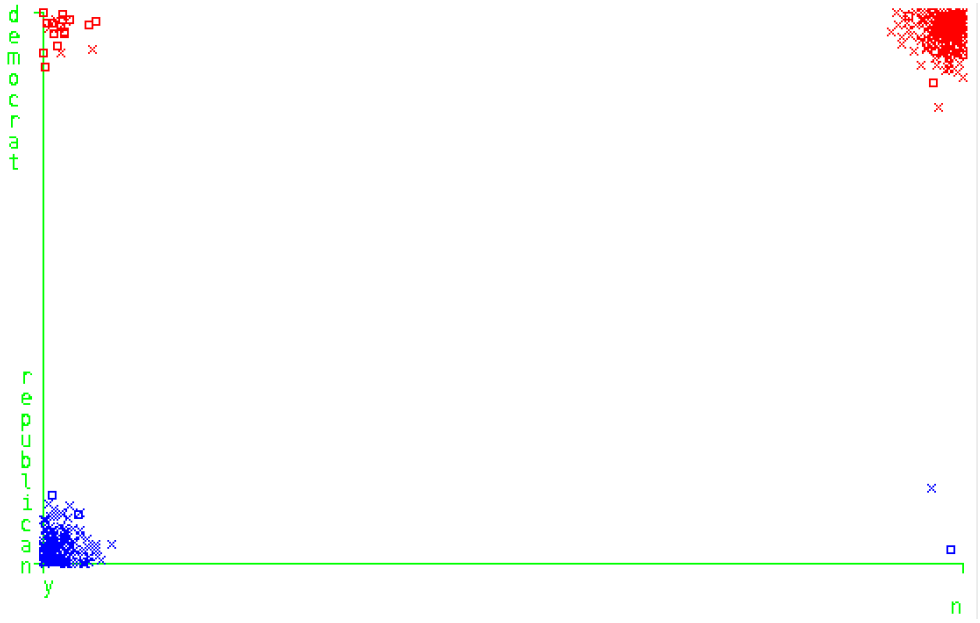
Attribute	Full Data	Cluster 0	Cluster 1
Religion in School	Yes	Yes	No
Education Spending	No	Yes	No
Crime	Yes	Yes	No
Budget resolution	Yes	No	Yes
*Handicapped Infants	No	No	Yes

*Voters who had infants with a disability

For EM k = 2, the error was **11%**, and it ran 8 iterations total. The log of likelihood was -5.87. EM found the drastic difference that Republicans voted in favor of *freezing the physicians fee*, where Democrats were strongly in favor of the opposite. Here is a visual to convey on how drastic EM clustered the two parties on this issue.

The Republic cluster is blue, and they voted almost completely yes on the issue. The top right is the Democrat cluster that voted mostly no. EM was able to differentiate the drastic differences with these clusters and find patterns in the voters' preferences, which is key for any data scientist.

Freezing Physicians Fee vote vs. Political Party



Dimensionality Reduction

Now I will be using the same two datasets, *Wine* and *Voting*, that have 4 dimensionality reduction algorithms applied to them: Principal Components Analysis, Independent Components Analysis, Randomized Projections, and Random Subsets. These reduction algorithms will be applied and then the datasets will be analyzed under k-means and EM.

Principal Components Analysis

White Wine

After applying PCA to the White wine dataset, the attributes went from 12 to 10, so the 2 eliminated were residual sugar and total sulfur dioxide. The eigenvalues of the remaining attributes are listed in the excel file under the PCA tab, but the lowest two were eliminated. Residual sugar and total sulfur dioxide had the least useful information for clustering the data.

For *k*-means, the number of iterations was 32 this time, and **55%** were incorrectly classified, which is actually higher than without dimension analysis. Cluster 2's quality was Good, cluster 0 was Bad, and cluster 1 was Great. Below is a further breakdown of the results. There were 1515 instances sorted into the Bad cluster (31%), 1532 instances sorted into the Great cluster (31%), and 1851 instances sorted into the Good cluster (38%). Overall, eliminating the two attributes had a very little effect on *k*-means.

Attribute	Full Data	Bad	Great	Good
0	0	-2.0609	1.477	0.4643
1	0	0.11	-0.3062	0.1634
2	0	-0.0748	0.0143	0.0494
3	0	0.2111	0.0971	-0.2531
4	0	0.0866	0.6284	-0.591
5	0	0.0576	-0.1122	0.0457
6	0	-0.0354	-0.0041	0.0324
7	0	0.2963	0.3174	-0.5052

For EM, it took 1.5 seconds to build the full model, ran through 41 iterations. **53%** were incorrectly classified and the log of likelihood was -12.2. EM sorted cluster 0 as Good, cluster 1 as Great, and cluster 2 as Bad. The results are detailed below.

Attribute	Full Data	Good	Great	Bad
0	0	-.2982	1.9034	-.6994
1	0	0.024	-.1403	0.0329
2	0	0.0426	-0.0829	-0.2409
3	0	0.0689	0.4879	-1.5127
4	0	-0.1027	.2743	0.447
5	0	0.1729	-.5055	0.0457
6	0	0.0685	-.1962	-.2744
7	0	-.0681	.1819	.2964
8	0	-.0889	0.3081	0.2591

There were 3820 instances sorted into the Good cluster (78%), 734 instances sorted into the Great cluster (15%), and 344 instances sorted into the Bad cluster (7%). EM drastically categorized the data differently after PCA was applied, being much more lenient on categorizing wine as Good.

Voting

After applying PCA to the Voting dataset, the number of attributes dropped from 13 to 11. The two attributes that were dropped are physician's freeze and aid to Nicaragua. The eigenvalues of the all of the attributes are listed in the excel file, under PCA tab, but the lowest two were eliminated again.

For *k*-means, there were 2 clusters, it ran 14 iterations, and the error rate was **18%.** This error rate is again higher without any dimensionality reduction. Cluster 0 was labeled as Democrat, and Cluster 1 was Republican. There were 210 (48%) instances in the Democrat cluster, and 225 (52%) in the Republican one. Further details are included below.

Attribute	Full Data	Democrat	Republican
0	0	0.0075	-.007
1	0	0.0855	-0.0798
2	0	-0.0614	0.0573
3	0	0.0188	-0.0176
4	0	-0.0211	0.0197
5	0	0.0912	-0.0852
6	0	-0.0308	0.0287
7	0	0.0455	-0.0425
8	0	0.0036	-0.0034
9	0	-0.9325	0.8703

PCA affected the Voting dataset almost exactly the same as Wine.

For EM, it took 0.07 seconds to build the model, and it took 43 iterations. It had **39%** incorrectly clustered instances, which is dramatically higher than last time. Cluster 0 (Democrat) only had 23% of the total instances, where the Republican cluster had 336 or 77%. Here is a further analysis of EM attributes.

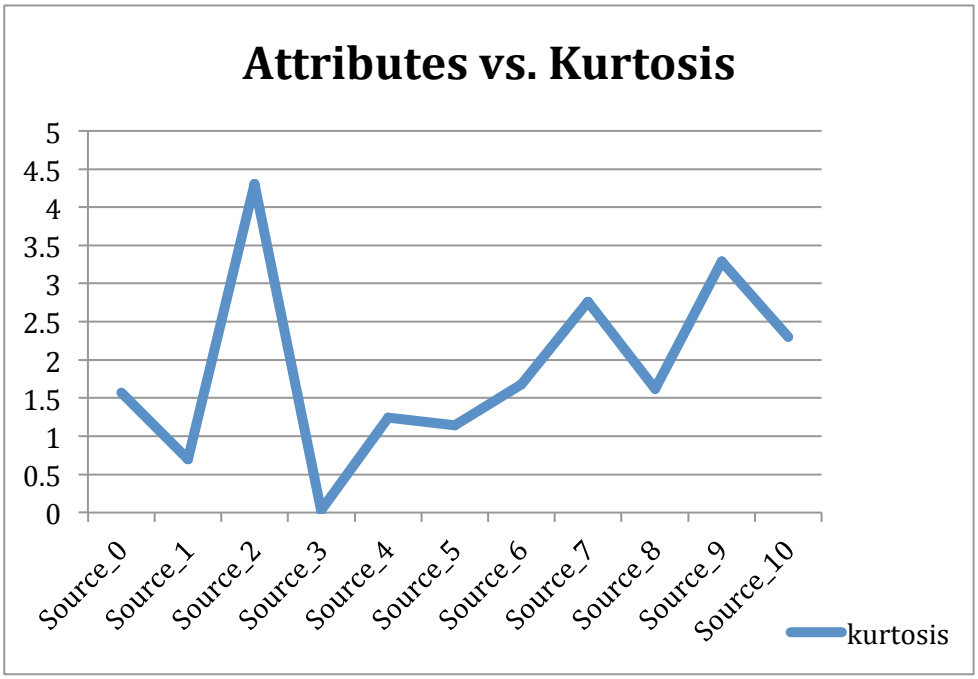
Attribute	Democrat	Republican
0	0.171	-0.0499
1	0.4141	-0.121
2	.0093	-.0027
3	0.3403	-.0994
4	-.1292	0.0377
5	-.0773	0.0226
6	0.3946	-.1153
7	-0.296	0.0865
8	-0.008	0.0023
9	-1.196	0.3492

Independent Components Analysis

White Wine

When ICA was applied to the wine dataset 2 attributes were removed, volatile acid and residual sugar. The kurtosis of these values were 0.7 and 0.03, and thus because these were the closest to 0 ICA identified them as Gaussian, or unnecessary. The largest gap between the average and the lowest kurtosis value was 0.03. The highest kurtosis was 4.3, giving the set a range of 4.27. The full set of kurtosis values can be found in the excel file under the ICA tab. There is a figure on the next page.

When *k*-means was run with ICA, it incorrectly classified **50%**, which is better than the original. It took 0.09 seconds to build the model and it ran 29 iterations. K-means found cluster 0 as Bad, 1 as Great, and 2 as Good. The Bad cluster had 1764 instances (36%), Great had 1341 (27%) instances, and Bad had 1793 instances (37%).



After running k-means with ICA it was much easier to determine wine instances that were of Bad quality, where before it was confusing it with the Good cluster.

Attribute	Good Cluster	Great Cluster	Bad Cluster
Source_0	0.068	0.0958	0.075
Source_2	-.0384	-.0424	-.0293
Source_4	.0215	.0182	.015
Source_5	-.0552	-.0614	-.0546
Source_6	-.0253	-.0327	-.0255
Source_7	.0064	.0124	.0037
Source_8	.067	.0804	.0697
Source_9	.0203	.0189	.0436
Source_10	.0705	.0942	.0753

For EM, I used 3 clusters, and the error rate was **53%.** It took 1.24 seconds to build a model, and the log of likelihood was 25.7. Cluster 0 was labeled as Good and it had 3155 instances (64%). Cluster 1 was labeled Great, had 1304 instances (27%), and cluster 2 was Bad, and it had 439 instances (9%). The details are below.

ICA changed the attributes so that EM labeled everything as Good. This definitely was not the most accurate because there should be many bad attributes as well, but the Bad cluster only had 9%. This was definitely a quality specific error, but the overall error was around the same.

Attribute	Full Data	Bad	Great	Good
Source_0	-0.0229	-.0188	-.0285	-.0228
Source_2	.0516	.0475	.0623	.0475
Source_4	.0027	-.0016	.0148	-.0023
Source_5	.087	.0876	.0931	.0818
Source_6	.0069	.0093	.0009	.009
Source_7	.0613	.055	.0693	.0616
Source_8	.1013	.0815	.1207	.1062

Source_9	.0067	.0049	.0129	.004
Source_10	.0377	.0346	.0385	.0401

Voting

When ICA was applied to the Voting dataset no attributes were actually removed. I then calculated the kurtosis values, and decided to manually eliminate the 3 closest to 0, South Africa Act, Physician's freeze, and religion in school. However after running *k*-means and EM with these attributes eliminated, the algorithms error rates were over 20% worse than by leaving all the attributes in there. That being said I left the dataset unmodified after ICA. For all of the kurtosis values check out the excel file under the ICA tab.

For k-means the error rate was **6%** and it only took 3 iterations. Cluster 0 was labeled Republican with 43% of the instances, where cluster 1 was Democrat and had the rest of the instances. The cluster details were left out because of repetitiveness.

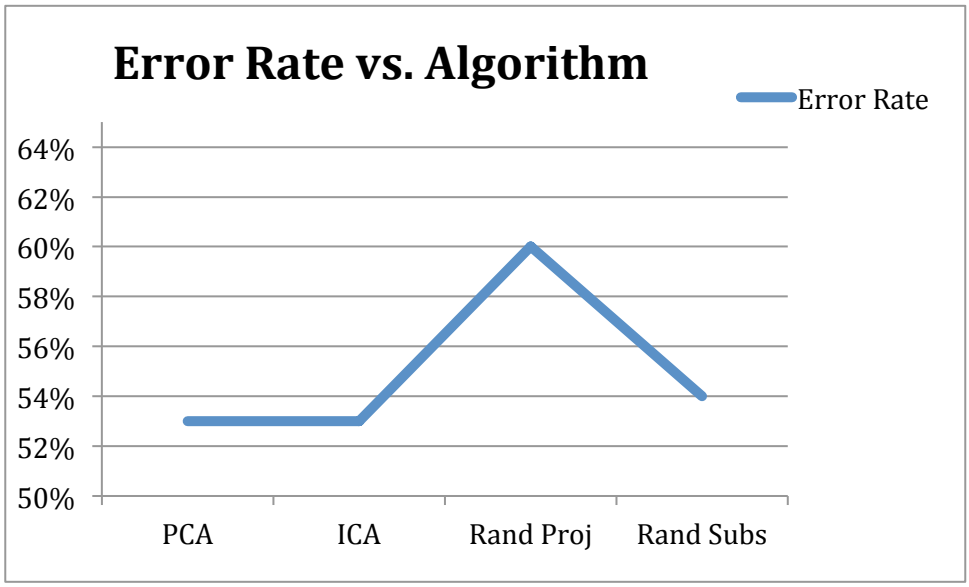
For EM, the error rate was **5%** and it ended up only running 7 iterations. Cluster 0 was Republican and it contained 43% of the instances, where Cluster 1 was Democrat and contained the rest. The log of likelihood was 20. The cluster details were left out because of repetitiveness.

Randomized Projections

White Wine

Because this algorithm projects onto lower random subspaces, I ran it through 3 values of attributes: a=11, a=9, and a=6. However, whether using 6 or 11 attributes the percentage error did not change, for *k*-means or EM, as they were **59%** and **60%**. The average number of iterations was 26, and cluster 0 was Bad (25%), cluster 1 was Great (35%), and cluster 2 was Good (43%). Tables were excluded because these values would be averages.

For EM, it performed 67 iterations and the log of likelihood was -34. The error rate was extremely high compared to the other dimensionality reduction algorithms used for EM.



Voting

I tried breaking up the attributes into different values of subsets, but this had very little affect on the outcome. The average was when there were 8 subsets. For k-means, it ran 12 iterations, and the clustering error was **34%**. Cluster 0 was Republican and was 35% of the instances, while cluster 1 was Democrat and claimed the rest.

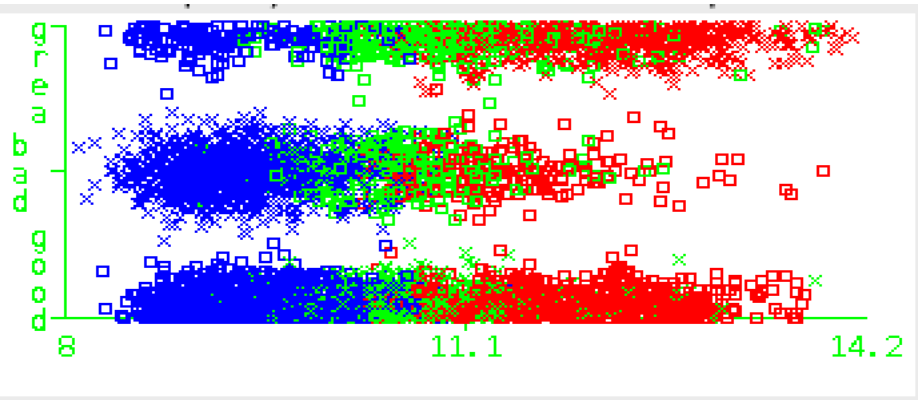
For EM, it ran 2 iterations, the clustering error was again **34%**, and the log of likelihood was -7.26. EM classified the data exactly the opposite of k-means, where cluster 0 was Republican and had 65% of the instances, and the rest were in cluster 1 as Democrat. I have left off visuals for this section as well to avoid redundancy.

Random Subsets

White Wine

For my 4th algorithm I used Random Subsets, which just returns a random subset of attributes. It kept 6 of the original 12 attributes.

For *k*-means, it incorrectly classified **52%** of the instances, ran 9 iterations, and sorted Cluster 0 as Bad (47%), Cluster 1 as Great (30%), and Cluster 2 as Good (23%). An interesting image is the way alcohol content was sorted per cluster. Each cluster the instances of wine with poorer quality had the lowest percentage of alcohol. Blue represents Bad, red is Great, and green is Good.



For EM, the inaccuracy was **54%**, it ran 43 iterations, and the log of likelihood was -4.69. Cluster 0 was Good (42%), cluster 1 was Great (29%), and cluster 2 was Bad (29%).

Voting

Attribute	Good Cluster	Great Cluster	Bad Cluster
Fixed Acidity	6.9474	6.5798	6.9868
Residual Sugar	5.0054	2.4159	12.2045
Total Sulfur Dioxide	133.6695	110.0841	172.3027
Density	0.9938	0.9907	0.9976
Sulfates	0.4827	0.4927	0.4973
Alcohol	10.4139	11.8743	9.3489

The Random Subsets algorithm reduced the number of attributes from 12 to 9. For *k*-means, it incorrectly classified **21%** of the instances, and went through 4 iterations. Cluster 0 was Republican and had 58% of the total instances, and cluster 1 was Democrat. Below are more details about each cluster.

Attribute	Full Data	Republican	Democrat
Handicapped-infants	No	No	Yes
Water cost	Yes	Yes	No
Physician freeze	No	Yes	No
El-Salvador Aid	Yes	Yes	No
Religion in School	Yes	Yes	No
Aid Nicaragua	Yes	No	Yes
Immigration	No	Yes	No
Crime	Yes	Yes	No
South Africa Act	Yes	Yes	Yes

For EM, it classified **14%** incorrectly, ran 11 iterations, and its log of likelihood was -4.88. I did not include any tables to avoid redundancy. The behavior was expected and very similar to not running any dimensionality reduction at all.

Dimensionality Reduction Summary

Every reduction algorithm had a different effect on the clustering of Wine and Voting. PCA didn't improve either algorithms performance, and it actually made it worse. Even though attributes were eliminated, the amount correctly classified became worse. ICA greatly improved each algorithm's ability to correctly identify each instance. It had a much more dramatic effect on Voting (even though no attributes were eliminated) and reduced the Wine dataset inaccuracy. Random Projections had a negative effect on both datasets, especially for EM. It caused the EM inaccuracy for Wine to increase 10% higher than any of the other algorithms. Random subsets improved the EM accuracy for each dataset, but actually had a negative effect on *k*-means clustering.

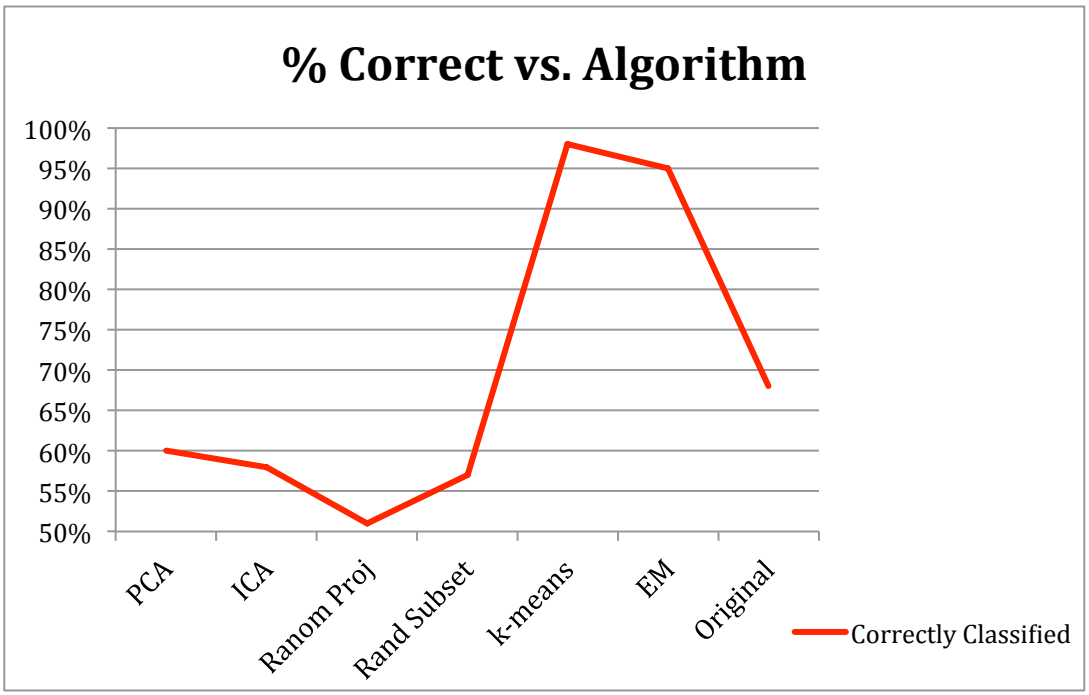
Neural Networks – Wine Dataset

In the last task, I added all of the Dimensionality Reduction algorithms as well as the clustering as pre-processors for my data. I then ran it through a neural network, comparing it to the original results from Project 1. The results are below.

For the neural network using EM and k-means, it was classified based on the clusters generated, which matched the amount of classes. It's ridiculous to see the increase in the amount correctly classified because these clustering algorithms were used, and it only took 4 more seconds to build

Algorithm	Time	Correctly Classified	Root Mean Squared Error	Rel. Absolute Error
PCA	4.71 s	60%	0.4194	80%
ICA	5.87 s	58%	0.4204	79%
Random Projections	4.76 s	51%	0.44	89%
Rand Subset	3 s	57%	0.43	83%
k-means	7.17 s	98%	0.0792	2.23%
EM	7.21 s	95%	0.157	8.52%
Original Data	2.3 s	68%	unknown	unknown

the model. I also ran experiments where the raw data went through dimensionality reduction and clustering algorithm before being classified with the neural network. This caused all of the results to become extremely high, due to also using EM and k-means as a pre-processor on the data. I have included the table for those results under the graph.



Neural Networks Summary

It's easy to see that all of the dimensionality reduction algorithms reduce the neural net's ability to classify the wine. What I found most interesting was the difference it made to classify off of the clusters instead of the quality. Because the clustering algorithms were successful in dividing the wine it made classification nearly perfect for the ANN. This is because the clusters contain highly relevant feature data that it can now learn from. Each cluster contains information pertaining to classifying the wine. It is interesting to see that clustering your data before running your ANN makes it classify much more successfully.