Examining Income and Dermatology Datasets using Clustering and Dimensionality Reduction

Andrew Fang

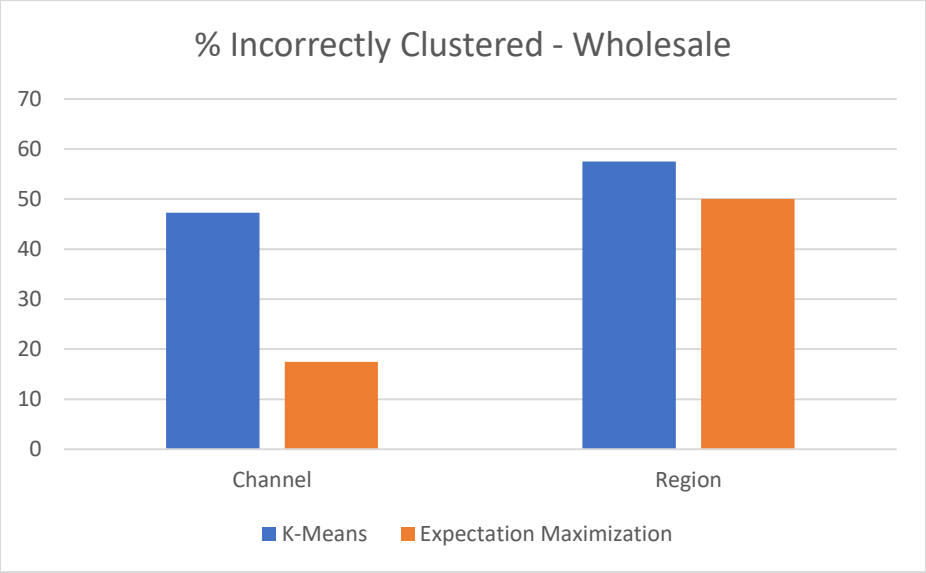
The Datasets

The wholesale dataset explored in this report describes spending at a wholesale retailer. It includes annual spending on different product categories, such as milk, grocery, frozen, and deli. It divides the data into 3 regions (of Portugal) and 2 channels. Purchasing patterns at wholesale retailers is very high volume and what people buy influence how the store is operated, which is interesting.

The Income dataset explored in this report describes incomes the general population. The dataset includes demographic information such as education, marital status, occupation, race, and sex. It attempts to predict whether the person makes more or less than $50,000 a year. I had to preprocess the data by changing the prediction class into a nominal attribute.

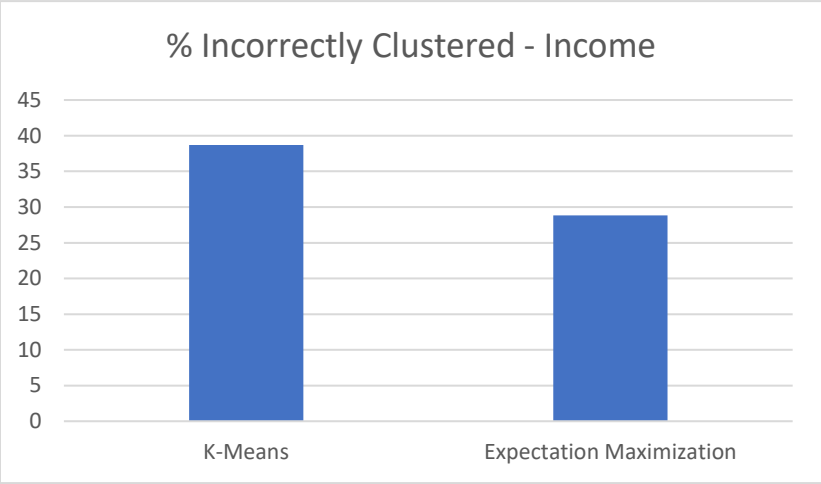
Clustering – K-Means vs Expectation Maximization

First, I examined the effects of simply running the k-means clustering algorithm on the two datasets. I evaluated the algorithms based on the number of incorrectly clustered instances. First, using the wholesale dataset, I ran the clustering algorithms, setting the number of clusters for the channel class to 2, and the region class to 3 because there are 2

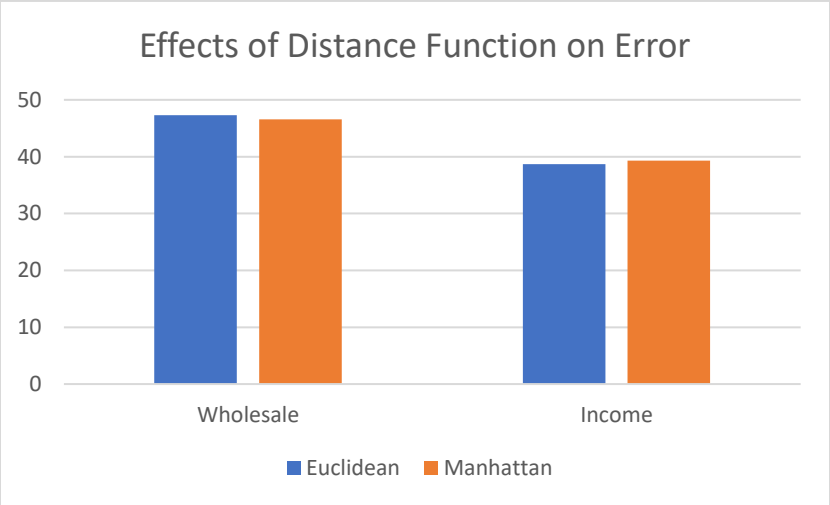


channels (Horeca and Retail) and 3 regions (Lisbon, Oporto, and Other). Error here was relatively high. For both algorithms, at least 50% of instances were incorrectly clustered (57.5 with K-Means and 50% with Expectation Maximization). It is notable, however, that the expectation maximization algorithm did better than the k-means algorithm in this case.

I next examined the clustering algorithms on the income dataset. Again, we can see that we have pretty high amounts of error (with k-means getting the cluster wrong 38.71% of the time and expectation maximization getting it wrong 28.83% of the time). Again, expectation maximization performs better than K-Means. This could be because the Expectation Maximization algorithm "weights" the distances where K-Means does not. This could produce more accurate results.



Next, I examined the effects of using Euclidean distance vs Manhattan distance as the distance function of the K-Means algorithm on both datasets. As it turns out, the difference in error is very small. The error when I changed the distance function changed by less than one percent.



One last thing, I let the Expectation Maximization algorithm run without giving it a set number of clusters, just to see what would happen. The results were that there was a lot more error. Using the wholesale dataset, the EM algorithm produced 61.81% incorrectly clustered instances.

Performance of these algorithms may be increased through various methods. I can

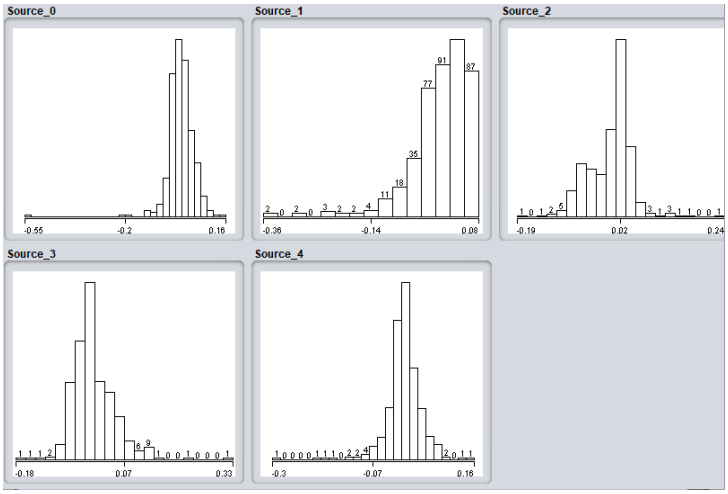
Dimensionality Reduction

First, I applied PCA to the wholesale dataset, which produced the following correlation matrix and these ranked attributes:

```
Correlation matrix
  1      0.1   -0.01   0.35  -0.1    0.24
  0.1    1      0.73   0.12   0.66   0.41
Ranked attributes:
 0.5592  1 0.579Grocery+0.549Detergents_Paper+0.545Milk+0.249Delicassen+0.051Frozen...
 0.2754  2 -0.611Frozen-0.528Fresh-0.504Delicassen+0.255Detergents_Paper+0.146Grocery...
 0.1521  3 -0.812Fresh+0.524Delicassen+0.178Frozen-0.136Detergents_Paper-0.108Grocery...
 0.0581  4 -0.769Frozen+0.552Delicassen+0.237Fresh-0.172Detergents_Paper-0.106Grocery...
 0.0105  5 0.827Milk-0.34Detergents_Paper-0.315Grocery-0.315Delicassen-0.049Fresh...
```

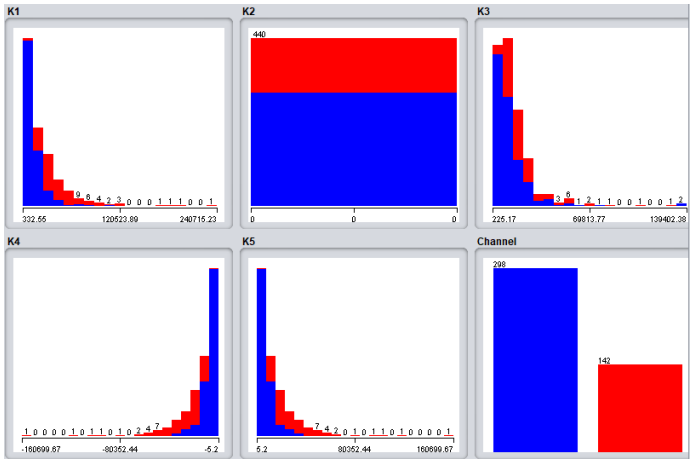
Here, we can see that grocery (attribute 3) and milk (attribute 2) have a high correlation, as do milk and detergent (attribute 5). But the highest correlation of .92 is between grocery and detergent. This is represented in the resulting ranked attributes. The top ranked attribute uses mostly combinations of the Grocery, Detergents, and Milk attributes.

Next, I applied ICA to the datasets using the IndependentComponents filter from the Students.Filters plugin found at https://github.com/cgearhart/students-filters. In order to do this, I had to convert all the attributes to numeric attributes by representing them as binary attributes. This increased the number of attributes significantly.



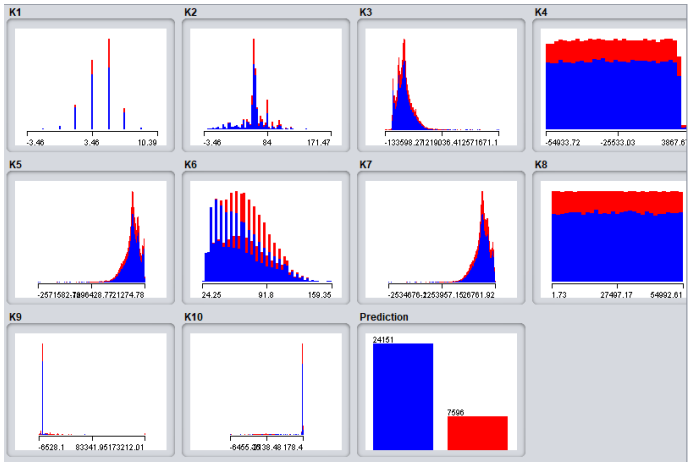
I then applied PCA to the wholesale dataset. I set the number of attributes to 5 because it does not seem like having more attributes than I started with is very useful. Weka did not display any particular quirks about these specific attributes.

Lastly, I applied randomized projection using the preprocessing filter in Weka. For the wholesale dataset, I reduced the data to 5 attributes (from 7). The following is what the attributes looked like:



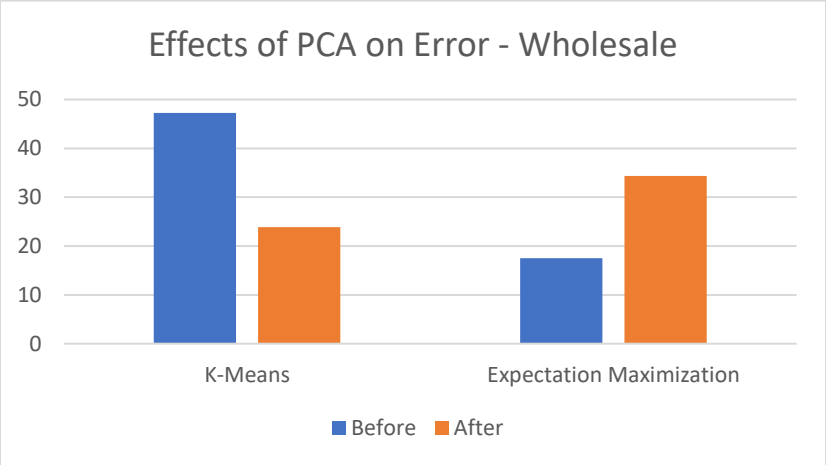
One interesting aspect of these projections is that the K4 and K5 attributes are reflections of each other about the Y axis. Another interesting aspect of this projection is that K2 is exactly zero.

Next, I applied the randomized projection on the income dataset, reducing it to 10 attributes from 15. Visualizations of the attributes are as follows:

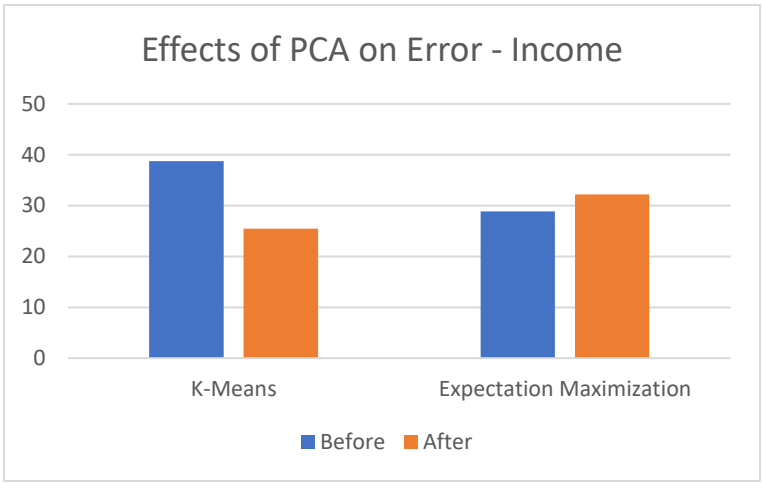


Clustering with Dimensionality Reduction

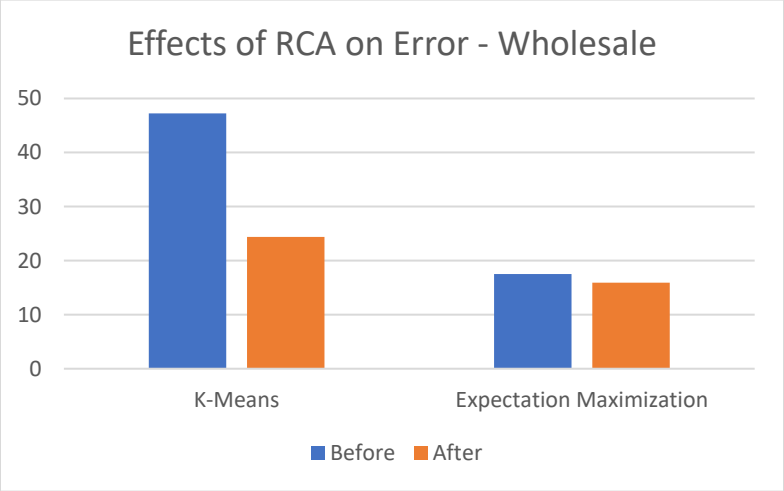
Then I examined the effect of these dimensionality reductions on the clustering algorithms that I ran before. First, I looked at the effect of PCA on the wholesale dataset. I ran a K-Means analysis on the dataset with PCA applied to it. The error for K-Means with PCA was 50% of the original error without PCA, which is a huge improvement. However, PCA combined with the Expectation Maximization algorithm increased error significantly. Almost doubling it. Initially, I guessed that this is due to loss of information through feature reduction.



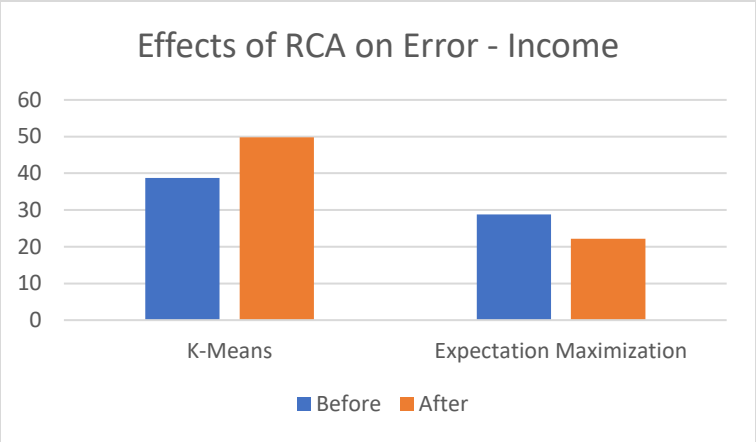
However, I saw similar results when applying PCA to t he income dataset. When I applied K-Means analysis to the reduced dataset, the number of individuals put in the wrong cluster decreased drastically. However, I, again, saw an (smaller) increase in the percentage of mis-clustered instances. This pattern of PCA decreasing error for K-Means but also increasing error for Expectation Maximization is most likely related to



Lastly, I examined randomized projection reduction. I applied RCA to the wholesale dataset and ran a K-Means analysis using the Channel attribute as a class. Again, K-Means saw a drastic reduction in the amount of error from 47.3% to 24.4% of instances mis-clustered. I then applied the RCA toExpectation Maximization. I then also saw a slight decrease in the number of mis-clustered individuals from 17.5% to 15.9% of the whole set. This follows a trend that has been evident in all of the data sets so far that show K-Means benefits from dimensionality reduction more than Expectation Maximization does.

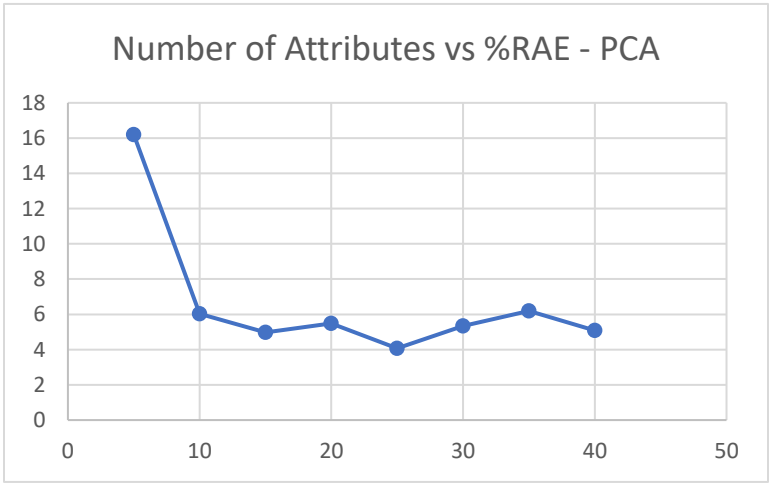


I then applied the randomized projection to the income dataset and examined its effects on the rate of mis-clustering. The interesting thing is that despite expectation maximization on the data with RCA applied to it decreasing in error (from 28.8% to 22.2%), the K-Means error increased. This is an exception to what I mentioned previously where K-Means benefits more from Dimensionality Reduction than Expectation Maximization. It is possible that this came about because even though I chose the same number of attributes to reduce down to, the larger size of the dataset influenced the error.



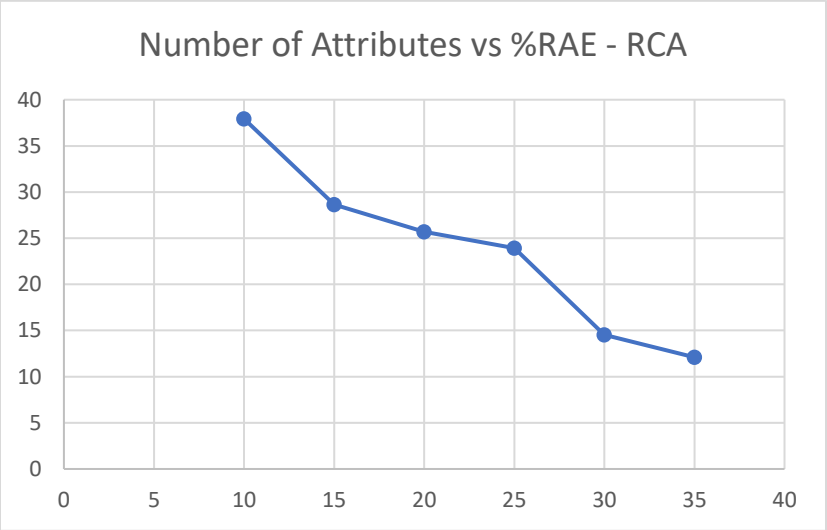
Dimensionality Reductions and Neural Nets

I applied PCA to the dermatology dataset from the previous assignment and created a neural net under default settings in Weka. Recall that the relative absolute error for this setting without any dimensionality reduction is 3.97%. I evaluated the model by training it on 66% of the data set and tested it on the remaining 34%. The relative absolute error was a less accurate 8.18%. I then experimented with changing the number of attributes. I found that increasing the number



of attributes caused a decrease in RAE. I also found that RAE eventually begins to increase as attributes increase on the neural network. Past a certain point (I examined "reductions" of 90 and 100 attributes), relative absolute error plateaus at 8.18%, a familiar number. This is likely because, since the dataset only has 35 attributes, going much past that probably does not yield any useful results.

I then ran RCA on the dermatology dataset (reducing it to 10 attributes), and ran a neural net algorithm, again training on the first two thirds of data and testing on the last. It returned a dismal 37.93% relative absolute error. Increasing number of attributes to 15 decreased the error to 28.64%. I eventually was able to get error down to around 12.1% at 35 attributes, which is still not as good as the base neural net algorithm. It is very important to note, however, that the original neural net took 25.23 seconds to build the model and the most accurate RCA reduced dataset only took 2.68 seconds.



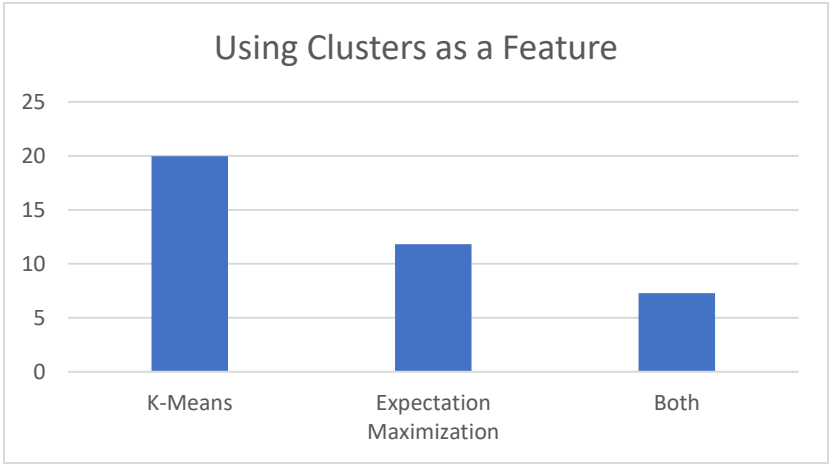
Effects of Using Clustering as a "Reduction" Technique

I applied the K-Means algorithm to the dermatology dataset and added the clusters to the feature set of the rest of the data. I then ran a neural net algorithm on it to see what would

happen. First, I noticed that the neural net took very little time to complete. It took 2.17 seconds compared to the 25.23 seconds without the reduction. Error was higher than PCA, but also lower than RCA at 19.96% relative absolute error.

I then replaced the clustering data from the K-Means algorithm with clustering data from the Expectation Maximization algorithm and ran it through the neural net algorithm. I, again, trained the model on two thirds of the data set and tested it on the remainder. The result was a more impressive 11.82% relative absolute error. The model also took a cool 2.17 seconds to create.

I then had the bright idea to add both of the clustering data to the feature space and ran a neural net on that, using the same techniques as before. The model took almost a second longer to create at 3.13 seconds. But it seemed worth it because the relative absolute error was a nice 7.29%. Although the error was higher than just the original neural net, the time it took to construct the models was much smaller than the original.



Conclusions

Clustering is different from classification in that it can group data points together without a target class. It is useful when you have a dataset and do not have a specific class to separate the instances from each other. However, it is not as accurate when compared to more

“focused” learning algorithms. It can, however, be used in conjunction with classification algorithms to to speed up the process of model generation/

I’ve found that generally, dimensional reduction algorithms tend to improve the accuracy of K-Means clustering, but is temperamental when used with expectation maximization. Sometimes, dimensionality reduction may improve the accuracy of clustering, but that may not be the best usage of the technique. Instead, the best usage of dimensionality reduction is probably to improve the speed of model creation.