Dataset 1) Annual Income Data

Abstract

The first dataset studied in this paper contains individual-record level data with attributes information such as education level, age, marital status etc. Each individual has a label classifying each as to whether he or she makes above or below $50,000 as an annual salary. Approximately 76% of the data points have a negative label (i.e. the individual makes below $50K), while the other 24% has a positive label (the individual makes above $50K). This paper primarily provides an unsupervised learning analysis of the data. This includes examining k-means clustering, EM clustering, and several dimensionality reduction algorithms—including principle components analysis and independent components analysis. In addition, the analysis covers training a neural network classifier using information gathered from the unsupervised algorithms.

The dataset were chosen because it demonstrates the strengths and weaknesses of the learning algorithms discussed in the analysis, while providing information about the underlying structure of the data not readily apparent. The data also provides a challenge of class imbalance as three quarters of the data has the same negative label.

Clustering Approach

K-means and EM clustering were each used on the dataset in this approach. In both cases, the class label was stripped from the dataset, so that only the attributes of each data point were input into the clustering algorithms. In the k-means method, several values of k were tested. Plotting within groups sum of squares by number of clusters as a k-value selection method, k = 3 was determined to be the result for k-means as the optimal number of clusters. This was true for most of the k-means implementations on this dataset (reduced data or otherwise). On the other hand, the EM clustering algorithm generated a collection of eight clusters as the optimal cluster split, using the Bayesian Information Criterion as its optimization metric.

K-Means Clustering

In performing k-means clustering, several choices of k were analyzed. The following tables show the distribution of the class labels across cluster for two sample choices of k. The k = 3 clustering shows relatively evenly-sized clusters; the spread of negative labels across the clusters is also relatively uniform. However, the density of the positive labels varies strongly between the clusters. Under the PCA and Information Gain sections later in the paper, we will discuss variables deemed “important” to the structure of the data by those algorithms. For comparison purposes, we use some of those attributes now to see how they compare across the clusters generated by k-means and EM. **For categorical data, the statistic shown is the percentage of that binary categorical variable belong to said category (i.e. percentage of data points in cluster that have the attribute “is husband”). For continuous variables, the mean is shown. Here, and in later tables, % Positive / % Negative refers to the proportion of labels within a given cluster having said label, %TP / %TN is the percentage of the total number of positive (or negative) labels captured by a cluster.**

The clusters seem to align with the "important" variables, as for example, in K = 3 below, two of the clusters have no individuals as husbands—all are contained in of the clusters. In K = 3 below, clusters 1 and 3 have similar ages, but differ widely in the gender category. There is also some difference apparent in the distribution of labels across clusters, with one of the clusters being almost all (over 97%) negative, versus the dataset proportion of 76%. The k-means implementations ran much faster than the EM—1.23 seconds for K = 3 versus 2165.48 for EM.

K = 3 and K = 5

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	32.28%	67.72%	20990	86.40%	57.51%	42	63.44%	63%	1412	91.13%
2	2.72%	97.28%	6658	2.31%	26.20%	26	0.08%	0.00%	269	39.76%
3	18.02%	81.98%	4912	11.29%	16.29%	40	34%	0%	745	0.31%

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	10.08%	89.92%	9691	12.46%	35.25%	38	0.04%	0%	716	59.93%
2	51.67%	48.33%	8142	53.65%	15.92%	44	99.95%	99%	2262	100.00%
3	31.55%	68.45%	3940	15.85%	10.91%	43	99.87%	99.21%	872	99.97%
4	9.67%	90.33%	9544	11.77%	34.88%	32	17%	0%	459	27.85%
5	39.50%	60.50%	1243	6.26%	3.04%	43	100.00%	98%	1539	100.00%

Expected Maximization (EM) Clustering

EM resulted in a "best" clustering consisting of eight clusters, as defined by BIC mentioned above. Interestingly, each individual cluster has at least 15% of its population with a positive label. There is some alignment between the labels and the attributes here, as over 70% of cluster 1 is positive and almost all of cluster 1 is a married male. This goes in line with gender and marital status being important variables (discussed more later). The age variable stands out here, as it varies little across cluster.

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	70.56%	29.44%	360	3.24%	0.43%	41	99.72%	99.72%	1265	99.72%
2	31.79%	68.21%	5467	22.17%	15.09%	40	51.69%	50.96%	1616	77.10%
3	15.26%	84.74%	9955	19.37%	34.13%	37	22.15%	12%	724	35.90%
4	26.54%	73.46%	1443	4.88%	4.29%	41	83.30%	83.09%	520	96%
5	15.26%	84.74%	6828	13.29%	23.41%	37	40.52%	31.28%	945	66.81%
6	45.21%	54.79%	3508	20.23%	7.78%	40	67.84%	66.16%	2276	88.31%
7	25.58%	74.42%	4249	13.86%	12.79%	39	58.53%	58.30%	720	90.44%
8	30.93%	69.07%	750	2.96%	2.10%	40	100%	100%	463	100%

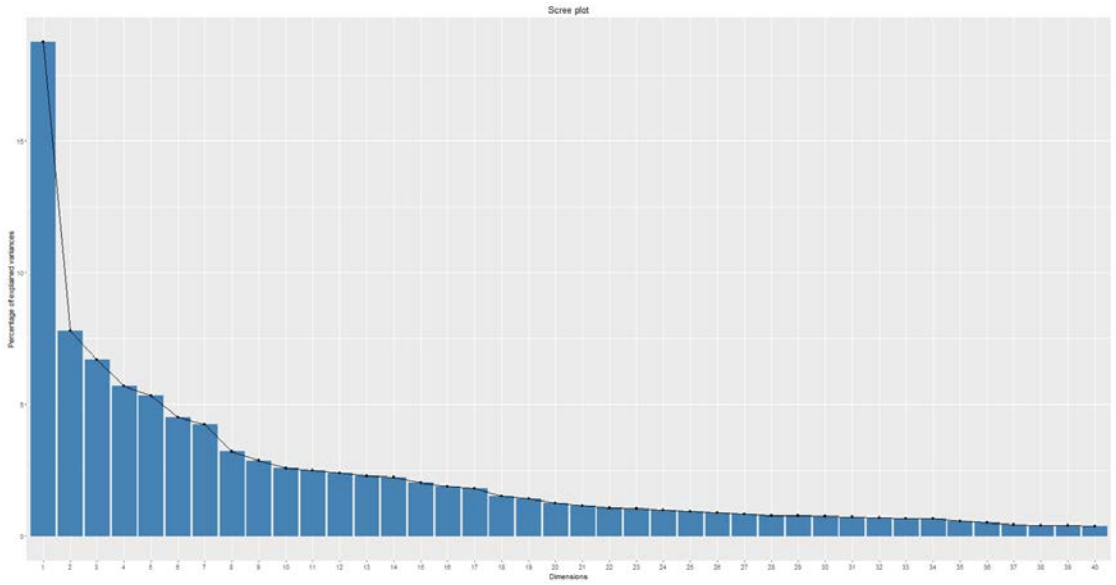
Principle Components Analysis

The dataset, after performing one hot encoding, has 107 features. The chart below shows the variances of the first forty principle components after implementing PCA. The table below shows the cumulative variance in the data explained by the first N number of components, for various values of N.

Number of Components	Percentage of Variance Explained	Reconstruction Error
10	61.60%	28725
20	80.90%	14315
30	89.90%	7500
40	95.40%	3441
50	98.10%	1450
75	99.70%	240
90	99.90%	44.3
100	~ 100%	~ 0

As can be seen in the scree plot below, the first component alone accounts for about 19% of the variance in the data. Looking at the underlying variables with the strongest contribution to this first component, the largest influencers revolve around marital status (six different possibilities in this dataset) and gender.

Underlying Variable	Coefficient Value in Linear Combination
Husband or Not (Binary Classification)	0.4942
Married-civ-spouse or Not (Binary Classification)	0.4797
Gender	0.3815
Marital Status: Never Married or Not (Binary Classification)	0.3091



The summary statistics of the eigenvalues are below. Of the 107 original features, 10 have corresponding eigenvalues less than 10^{-6}.

Min	1st Quartile	Median	Mean	3rd Quartile	Max	Std. Dev.
~ 0	0.0009	0.0048	0.043	0.0366	0.86	0.1038

Next, we use the principle components generated by this algorithm to train a neural network classifier. Varying the number of components (sorted in order from most variance explained to least), and using a backpropagation algorithm for the neural net, the following results are achieved:

Number of Components	Testing Accuracy	AUC	Clock Runtime	Precision	Specificity	Recall (Sensitivity)	Training Accuracy
2	77.10%	58.20%	186.9	59.20%	95.30%	21.10%	78.20%
3	78.20%	63.00%	15.9	60.00%	92.80%	33.20%	78.80%
5	81.20%	69.60%	173.6	66.60%	92.30%	46.90%	81.20%
10	81.80%	70.20%	36.5	68.40%	92.90%	47.50%	82.50%
30	84.10%	74.90%	361.3	72.30%	92.90%	56.90%	84.30%
50	84.70%	76.00%	8.9	73.20%	93.00%	59.10%	84.70%
Non-Transformed Data	85.60%	77.60%	102.11	74.70%	93.20%	62.20%	85.40%

Notice the rapid increase in recall as the number of components increases. This drastic difference across number of components could partially be due to the class imbalance in the data—that there is a 3 to 1 ratio of negative labels to positives. With fewer components, PCA is providing the neural net with less information about the actual structure of the data, so the class imbalance is more pronounced i.e. the neural net "believes" that a greater proportion of the data is negative than is actually the case. As this knowledge is increased, the neural network is able to risk more that data points may belong to the positive class—resulting in the above slow decrease in specificity, as recall increases.

K-means clustering on PCA-reduced data

K = 3 and K = 5 Implementation Clock Time: 0.67 seconds

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	44.60%	55.40%	13315	75.74%	29.84%	44	99.93%	99.08%	1785	99.99%
2	8.53%	91.47%	8476	9.22%	31.36%	33	0.17%	0%	613	100%
3	10.95%	89.05%	10769	15.04%	38.80%	37	15.38%	0%	569	0%

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	18.02%	81.98%	4912	11.29%	16.29%	40	33.69%	0%	745	0.31%
2	0.82%	99.18%	4636	0.48%	18.60%	24	0.11%	0%	156	57.10%
3	44.60%	55.40%	13321	75.77%	29.86%	44	99.93%	99.04%	1784	99.99%
4	8.26%	91.74%	5401	5.69%	20.05%	32	0%	0%	580	62.53%
5	12.38%	87.62%	4290	6.77%	15.21%	46	0.09%	0%	887	56.67%

The above K = 3 example for the principle components generated by PCA is able to get a cluster with a positive label proportion much closer to ½ than when the K = 3 done on the original data. One possible reason for this is that a strength of PCA is its ability to map correlated variables

into uncorrelated features; this is important in this particular dataset because it seems that several variables are highly correlated with each other.

EM clustering on PCA-reduced data

The EM clustering implementation of the PCA-reduced dataset took 2294.28 seconds to run, resulting in a collection of nine clusters. Here, EM is able to get much "purer" clusters in terms of the key important variables than when done on the original data. This may be the case because PCA taps in the underlying structure and variance of the data, thus providing the EM algorithm with more pertinent information. Observe the husband binary variable below as an example—with each cluster being 100% or virtually 0%.

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	10.94%	89.06%	10773	15.04%	38.81%	37	15.38%	0.01%	568	0.03%
2	11.47%	88.53%	6406	9.37%	22.94%	36	2.12%	0%	788	100.00%
3	80.05%	19.95%	1554	15.87%	1.25%	42	100.00%	100.00%	4047	100%
4	0.59%	99.41%	2198	0.17%	8.84%	24	0.00%	0.00%	112	100.00%
5	53.15%	46.85%	3255	22.06%	6.17%	48	100.00%	100.00%	2757	100.00%
6	36.15%	63.85%	2470	11.39%	6.38%	43	100.00%	100.00%	1222	100.00%
7	43.78%	56.22%	1631	9.11%	3.71%	41	100%	100%	981	100%
8	31.77%	68.23%	1363	5.52%	3.76%	48	100%	100%	1060	100%
9	30.93%	69.07%	2910	11.48%	8.13%	41	100%	100%	807	100%

Independent Components Analysis

Our initial analysis with ICA for the dataset examines implementing the algorithm with varying numbers of components. The implementations took an average of 20.79 seconds to run. The table below shows the amount of variance explained by the top three components when ICA was implemented using varying numbers of total components**.** The second table shows the total variance explained by the data for N components.

Component	3	5	10	30	50
1	18.60%	18.20%	17.40%	14.40%	14.10%
2	8.00%	6.90%	6.30%	5.70%	5.80%
3	6.70%	6.70%	6.30%	4.80%	4.70%

Total Number of Components	Percentage of Variance Explained
3	33.30%
5	44.30%
10	61.60%
15	73.10%
20	80.90%
30	90.00%
50	98.10%
107	100.00%

The below table shows summary statistics for the kurtosis. ICA should increase the kurtosis of the dimensions, and this is the case here, though it is not as readily apparent until the kurtosis is calculated for a greater number of components. Between 5 and 10 components, the median of the dimensions is close to a Gaussian kurtosis number of 3. Overall, the transformed data's kurtosis levels are not much higher than the original data because it takes almost as many transformed dimensions as the original dataset to reach the former dataset's kurtosis levels. This indicates lower selectivity, and a larger number of attributes contributing little information about the data. This could be because of the underlying structure of the data, with several categorical attributes mapped by one hot encoding—creating a collection of binary variables that say little about the data. The smaller number of projected axes (components) capture an underlying normality in the data, implied by the kurtosis values around 3 with lower number of components. This underlying normality is not as apparent when greater number of components are present, nor as visible in the original data.

# Components	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3	1.346	1.656	1.967	2.038	2.384	2.801
5	1.288	1.627	1.981	2.242	3.014	3.300
10	1.196	2.360	3.909	3.506	4.116	5.663
15	1.275	2.636	4.899	4.880	5.972	12.907
20	1.123	3.631	6.043	6.073	8.021	14.591
30	1.574	5.957	10.764	11.322	16.076	23.337
50	1.627	7.466	21.954	29.094	32.371	116.315
99	1.637	19.853	94.168	510.978	590.728	4648.254
Original Data	1.026	12.262	54.546	488.418	531.916	4649.429

K-Means Clustering on ICA-reduced Data Implementation Clock Time: 0.67 seconds

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	12.49%	87.51%	2001	3.19%	7.08%	38	45.70%	40.16%	1103	66.74%
2	23.14%	76.86%	22931	67.68%	71.30%	43	49.16%	44.78%	835	67.22%
3	29.94%	70.06%	7628	29.13%	21.62%	48	55.57%	51.55%	234	75.23%

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	26.04%	73.96%	987	3.28%	2.95%	45	71.53%	66.84%	10414	84.03%
2	11.75%	88.25%	11506	17.24%	41.08%	38	26.58%	21.40%	599	50.67%
3	31.04%	68.96%	18112	71.70%	50.53%	38	47.16%	42.06%	913	69.52%
4	38.42%	61.58%	950	4.68%	2.37%	39	49.86%	43.13%	715	63.82%
5	24.38%	75.62%	1005	3.12%	3.07%	40	48.58%	40.88%	1203	61.89%

The clusters generated by k-means in this case have a distinctly different distribution than the ones developed on the PCA-reduced dataset, as well as the original dataset. Here, the split between positive and negative seems to be closer to the overall 24 / 76 split, with no cluster in either the k = 3 or k = 5 case having more than a 40% cut of positive labels. Also, the cluster sizes generated here are much less evenly distributed, with both k-values shown resulting in a single cluster that contains ~23,000 of the data points—i.e. about 70% of the total individuals in the dataset. Here, the clusters could be comparing different characteristics than the ones shown, because none of the clusters here show much extreme variation in the above attributes. Also, since ICA searches for components as independent of each other as possible, whether than looking for those that explain the most amount of variation in the data, like in PCA, this would probably lead the clustering results to not have as much pure extremity between the clusters as in PCA. K-means took 0.37 seconds to run.

EM Clustering on ICA-reduced Data

The EM clustering implementation of the PCA-reduced dataset took 2196.37 seconds to run, resulting in a collection of nine clusters. EM has more extreme variation in the attributes shown than in k-means. This could be because ICA looks for components as independent as possible, and EM is better able to use this information than k-means in terms of separating attribute differences between clusters, due to its maximum likelihood nature.

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	0.58%	99.42%	1907	0.14%	7.67%	23	0.00%	0%	148	0.00%
2	7.20%	92.80%	1986	1.82%	7.46%	32	0.00%	0.00%	536	0%
3	31.18%	68.82%	4281	17.03%	11.92%	43	99.93%	99.88%	887	99.88%
4	18.36%	81.64%	6808	15.94%	22.48%	42	26.32%	0.16%	915	38.26%
5	6.22%	93.78%	3475	2.75%	13.18%	40	0.00%	0.00%	451	23.25%
6	0.91%	99.09%	2523	0.29%	10.11%	24	0%	0%	147	100%
7	43.26%	56.74%	2427	13.39%	5.57%	42	100%	100%	1146	100%
8	54.46%	45.54%	6482	45.02%	11.94%	45	100%	100%	2639	100%
9	10.60%	89.40%	2671	3.61%	9.66%	32	0%	0%	709	100%

Next, we using the components from performing ICA to train a neural network classifier on the data. Like with the PCA-based neural net, this also uses a backpropagation-based algorithm.

Number of Components	Testing Accuracy	AUC	Clock Runtime	Precision	Specificity	Recall (Sensitivity)	Training Accuracy
2	75.50%	50.00%	8.44	N/A	100.00%	0.00%	76.20%
3	75.50%	50.00%	1.5	N/A	100.00%	0.00%	76.20%
5	75.50%	50.00%	6.84	N/A	100.00%	0.00%	76.20%
10	77.80%	59.60%	16.85	62.10%	95.30%	23.90%	78.40%
30	80.70%	67.70%	72.23	66.70%	93.20%	42.20%	81.00%
50	83.70%	74.30%	18.57	71.60%	92.80%	55.70%	84.00%
Non-Transformed Data	85.60%	77.60%	102.11	74.70%	93.20%	62.20%	85.40%

Similarly to the PCA-based neural network, this version of the neural net shows a vast difference in recall as the number of components increases, even equaling zero when the number of components is less or equal to five. This greater extreme versus the PCA-based model is most likely due to the ICA process of seeking to maximize the independence of the output components, rather than outputting components that explain the most amount of variance in the data. In other words, the nature of PCA looks for a global structure in the data, whereas ICA finds local structures in the data. Since the class labels are disproportionately imbalanced, this shows a weakness of ICA in the case, as only having a smaller number of independent components can be misleading to the overall global structure of the data.

Randomized Projection

RCA is implemented here, using varying numbers of target output components. This algorithm ran the fastest of any of the dimensionality reduction algorithms—at 0.41 seconds. The reconstruction errors from the result components are much higher than those following PCA.

Number of Components	Reconstruction Error
10	132,568
20	123,155
30	114,074
40	108,905
50	103,510

After running the RP model 100 times, the reconstruction error for 50 components still range between ~ 95,000 and 123,000, with a standard deviation of 4847.72. This could be because of extraneous variables present in the data, skewing the results. RCA generates random matrices assuming a Gaussian distribution, which may be skewed in this case because of underlying data issues of non-normally distributed variables.

K-Means Clustering on RCA-reduced Data Implementation Clock Time: 5.53 seconds

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	32.28%	67.72%	20990	86.40%	57.51%	39	0.19%	0.01%	639	45.51%
2	2.72%	97.28%	6658	2.31%	26.20%	44	99.00%	98.09%	1774	99.60%
3	18.02%	81.98%	4912	11.29%	16.29%	28	24.14%	0.06%	496	41.10%

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	10.08%	89.92%	9691	12.46%	35.25%	38	0.17%	0.03%	763	45.63%
2	51.67%	48.33%	8142	53.65%	15.92%	45	99.18%	97.29%	2392	98.60%
3	31.55%	68.45%	3940	15.85%	10.91%	28	21.98%	0.02%	464	42.46%
4	9.67%	90.33%	9544	11.77%	34.88%	39	1.99%	0.79%	448	45.61%
5	39.50%	60.50%	1243	6.26%	3.04%	42	98.98%	95.52%	1338	97.23%

Similar to other k-means implementations, here it was able to capture clusters with almost entirely male individuals, and entirely “is husband=TRUE” individuals. The cluster label proportions are very similar to when run on the original dataset. This makes sense, given that RCA seeks to preserve the distances between data points, thus feeding this “same” (or as close to as it makes possible) information to k-means.

EM Clustering on RCA-reduced Data Implementation Clock Time: 2196.37 seconds

The EM clustering resulted in a collection of nine clusters. Here, the attributes across clusters are similar to how they were when EM was done on the original data—keeping in line with the fact that distances between data points are sought to be preserved when RCA is implemented.

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	70.56%	29.44%	360	3.24%	0.43%	41	99.72%	99.72%	1265	100%
2	31.79%	68.21%	5467	22.17%	15.09%	40	51.69%	50.96%	1616	77.10%
3	15.26%	84.74%	9955	19.37%	34.13%	37	22.15%	11.70%	724	35.90%
4	26.54%	73.46%	1443	4.88%	4.29%	41	83.30%	83.09%	520	96.26%
5	15.26%	84.74%	6828	13.29%	23.41%	37	41%	31%	945	67%
6	45.21%	54.79%	3508	20.23%	7.78%	40	68%	66%	2276	88%
7	25.58%	74.42%	4249	13.86%	12.79%	39	59%	58%	720	90%
8	30.93%	69.07%	750	2.96%	2.10%	40	100%	100%	463	100%

Next, we using the components from performing RCA to train a neural network classifier on the data. Like with the PCA and ICA-based neural nets, this also uses a backpropagation-based algorithm. The overall performance of the RCA-based neural network shows results between the PCA and ICA-reduced datasets.

Number of Components	Testing Accuracy	AUC	Clock Runtime	Precision	Specificity	Recall (Sensitivity)	Training Accuracy
2	75.50%	50.00%	9.32	N/A	100.00%	0.00%	76.20%
3	75.50%	50.00%	11.46	N/A	100.00%	0.00%	76.20%
5	75.50%	50.00%	23.81	N/A	100.00%	0.00%	76.20%
10	80.20%	68.10%	19.44	63.90%	91.80%	44.30%	80.10%
30	82.30%	71.10%	47.29	70.00%	93.20%	49.00%	82.50%
Non-Transformed Data	85.60%	77.60%	102.1	74.70%	93.20%	62.20%	85.40%

Information Gain as Feature Selection

The final feature selection approach uses information gain to assigned importance weights to each attribute. This approach assigned 48 of the 107 attributes positive information gain, while the remaining attributes have zero gain. The top five attributes by information gain are listed below. Information gain was calculated both prior to and after one hot encoding (left and right tables respectively). The most important attributes by this metric are similar to those selected when PCA was performed, with the top two most significant attributes being the same. This feature selection implementation took 17.97 seconds to run.

Attribute	Information Gain
Marital Status – "Married-Civil Spouse" or NOT (Binary)	0.0863
Relationship – Husband or NOT (Binary)	0.0673
Capital Gain (continuous)	0.0643
Education Number (continuous)	0.0541
Age (continuous)	0.0332

Attribute	Information Gain
Marital Status	0.1146
Relationship	0.1085
Capital Gain	0.0794
Age	0.067
Education	0.0648

K-Means Clustering on Information Gain-Reduced Data

Reducing the 107-attribute dataset to the 48 with positive information gain, k-means clustering yields the results below. With the exception of age, the clusters seems to capture the variation between the data very well—with several clusters have virtually all males versus no males, and capital gain being much higher in some clusters than in others. The distribution of age in the dataset could be making it more difficult for the clustering to adequately separate the data points on age across cluster.

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	10.95%	89.05%	10769	15.04%	38.80%	37	15.38%	0.00%	569	0.00%
2	44.61%	55.39%	13314	75.74%	28.84%	44	99.93%	99.09%	1785	99.99%
3	8.53%	91.47%	8477	9.22%	31.37%	33	0.18%	0.00%	613	100.00%

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Mean Age	% Civ. Spouse	% Husband	Cap. Gain	% M
1	37.59%	62.45%	7108	34.04%	17.96%	42	99.92%	98.82%	1316	99.99%
2	66.69%	33.31%	2501	21.27%	3.37%	43	98.24%	97.28%	2553	100.00%
3	6.52%	93.48%	12139	10.09%	45.91%	30	0.00%	0.00%	496	66.13%
4	15.48%	84.52%	7023	13.86%	24.01%	43	23.58%	0%	742	5.20%
5	42.91%	57.09%	3789	20.74%	8.75%	48	99.26%	98.60%	2143	100%

EM Clustering on Information Gain-Reduced Data

Performing this algorithm took 3972.74 seconds to run, resulting in a single cluster containing all of the dimension-reduced data points. EM was re-performed several times, but each time resulted in a single cluster. The more probabilistic nature of EM versus k-means, along with a fourth of the variables being taken out of consideration, may be causing this.

Training a neural network classifier on Information Gain-Reduced Data

Training a neural network classifier in this case was tried with both one and three hidden layers, with similar results. However, the recall and specificity levels seem to switch in performance in this scenario in comparison to when the neural net was trained previously. The overall test accuracy and specificity here is significantly worse than most versus when the neural net was trained under the other forms of dimension-reduced data, but the recall is significantly higher versus the other reduced data, as well as in comparison to the initial, non-reduced dataset. The information gain criterion is not taking into account the effects of how the removed attributes interact with each other, and the information from this interaction is lost on the neural network. This could be causing a greater false negative rate (i.e. lower specificity). This, in conjunction, with only the more "important" attributes (defined by the information gain criterion), being used in the neural net training process, the learner may be better figuring out what switches a label from the more prevalent negative to positive—what causes greater change to the structure of the data. Because of this, the learner is classifying more data points as "positive" than in the other training attempts, thus leading to the higher recall values.

# Hidden Layers	Testing Accuracy	AUC	Clock Runtime	Precision	Specificity	Recall (Sensitivity)	Training Accuracy
1	71.10%	76.00%	10.19	45.20%	66.30%	85.70%	71.00%
3	71.10%	76.00%	1837.14	45.20%	66.30%	85.60%	71.00%
Non-Reduced Data	85.60%	77.60%	102.11	74.70%	93.20%	62.20%	85.40%

Using Clusters for Feature Construction – Training Neural Network Classifier

In this part of the analysis for dataset, the clusters generated from performing k-means and EM are used as features in training a neural net. The clustering results are taken from when the clustering was performed on the dimensionality reduced data—two cluster-based attributes each from the PCA-reduced, ICA-reduced, RCA-reduced, and Information Gain-reduced datasets. Each pair of cluster-based attributes consists of one from the k-means clustering done, and one from the EM clustering performed. One hot encoding was then performed on this newly

constructed dataset, to get the final dataset used to train the neural network classifier. Different values of k from the corresponding k-means clustering are used in the training of the neural net.

K	Testing Accuracy	AUC	Clock Runtime	Precision	Specificity	Recall (Sensitivity)	Training Accuracy
3	80.70%	66.90%	306.52	68.40%	94.00%	39.70%	81.00%
5	80.60%	66.00%	103.39	69.50%	94.70%	37.40%	81.10%
15	81.90%	70.20%	720.82	69.10%	93.10%	47.30%	82.50%

Next, we take the features from dataset constructed above (using k = 3) with the features of the information gain-reduced dataset, and train a neural net, with these results (DRM = Dimensionality Reduced Model):

DRM	Testing Accuracy	AUC	Clock Runtime	Precision	Specificity	Recall (Sensitivity)	Training Accuracy
PCA	82.40%	71.10%	326.22	70.30%	93.30%	49.00%	82.70%
ICA	81.80%	69.10%	69.9	70.40%	94.00%	44.20%	82.00%
RCA	80.50%	66.80%	457.81	67.10%	93.60%	40.00%	81.00%
Info. Gain	84.70%	75.80%	920.94	73.80%	93.30%	58.40%	85.20%

Dataset 2) Exercise Data

Abstract:

This problem is designed to analyze the exercise movement an individual is performing based off various body movements, angles, etc. Each record in the datasets corresponds to an individual attempting to perform a Unilateral Dumbbell Biceps curl, with a positive or negative label indicating whether the exercise was correctly or incorrectly performed. The algorithms used to study the data include k-means clustering, EM clustering, and several dimensionality reduction algorithms—including PCA and ICA. The citation for the dataset is below. The dataset is interesting because it shows relatively wide variation between the different clustering techniques employed. Performance variation also exists between the reduction algorithms used.

K-Means Clustering

In performing k-means clustering, several parameters were examined, including number of clusters and type of algorithm. The tables below show examples of the clustering using k = 3 and k = 5. In both scenarios, along with several other choices of k, the positive / negative label distribution is around 30 / 70% across the clusters. Here, and for each k-means implementation on this dataset, the groups sums of squares by number of clusters was plotted to determine the value of k. In most cases, K = 3 was the optimal choice given this method (except for the PCA-reduced data having k = 5). For comparison purposes, K = 3 and 5 are each provided. For K = 3, clusters 1 and 3 appear to be rather similar across attributes. Likewise, clusters 1 and 4 for the K = 5 example are similar across attributes. It is possible that other, not shown, attributes could be causing the separation of those respective data points into different clusters.

K = 3 and K = 5 Implementation Clock Time: 0.64 seconds

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	27.88%	72.12%	15813	39.51%	40.61%	59.01	57.63	46.68	56.36
2	30.12%	69.88%	7801	21.06%	19.41%	29.32	39.29	36.11	49.8
3	28%	72%	15627	39.43%	40%	58.96	58.51	46.33	56.83

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	28.16%	71.84%	7365	18.59%	18.84%	29.33	37.46	30.92	41.29
2	29%	71%	9532	25.11%	24%	79.57	68.2	54.12	63.29
3	24%	76%	5273	11.54%	14%	25.97	49.04	48.63	64.69
4	28%	72%	7612	19.27%	19%	28.86	38.87	31.15	41.32
5	30%	70%	9459	25.49%	24%	79.51	68.89	53.57	63.94

EM Clustering

Implementation Clock Time: 1660.43 seconds

Based off the attributes tested, clusters 8 and 9 below appear to be rather similar, but overall, there appears to be more variation between clusters here than in k-means—possibly due to the greater number of clusters.

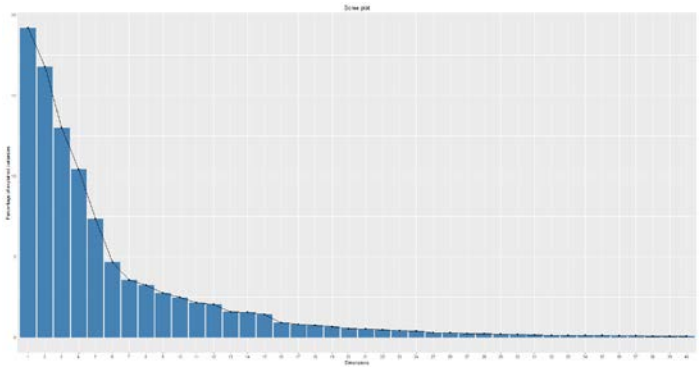
Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	30.51%	69.49%	3782	10.34%	9.36%	29.75	38.55	36.27	50
2	25.19%	74.81%	2723	6.15%	7.25%	26.36	48.57	48.7	63.97
3	23.85%	76.15%	6676	14.27%	18.10%	27.55	42.21	33.42	48.88
4	29.97%	70.03%	3991	10.72%	9.95%	29.21	40.14	36.07	50
5	24.30%	75.70%	8738	19.03%	23.56%	79.19	68.44	51.86	61.4
6	23.88%	76.12%	8841	18.92%	23.96%	79.28	67.8	52.5	60.83
7	29%	71%	3078	7.92%	8%	28.54	35.83	27.42	22.93
8	100%	0%	720	6.45%	0%	83.38	74.39	74.43	94.77
9	100%	0%	692	6.20%	0%	83.31	73.4	74.66	94.81

Principle Components Analysis

The dataset has 52 attributes. The first table below shows the cumulative variance in the data explained by the first N number of components. The first component, alone, accounts for about 19% of the variance in the data. The summary statistics of the eigenvalues are in the second table below. The second from last table below shows the variables deemed as the most important by their weights in the eigenvalues. The final table below shows the cumulative variance in the data explained by the first N number of components, for various values of N.

Number of Components (N)	Percentage of Variance Explained
10	83.20%
20	95.50%
30	98.70%
40	99.60%

Min	1st Quartile	Median	Mean	3rd Quartile	Max	Std. Dev.
2E-04	0.0013	0.0052	0.041	0.0332	0.409	0.0877

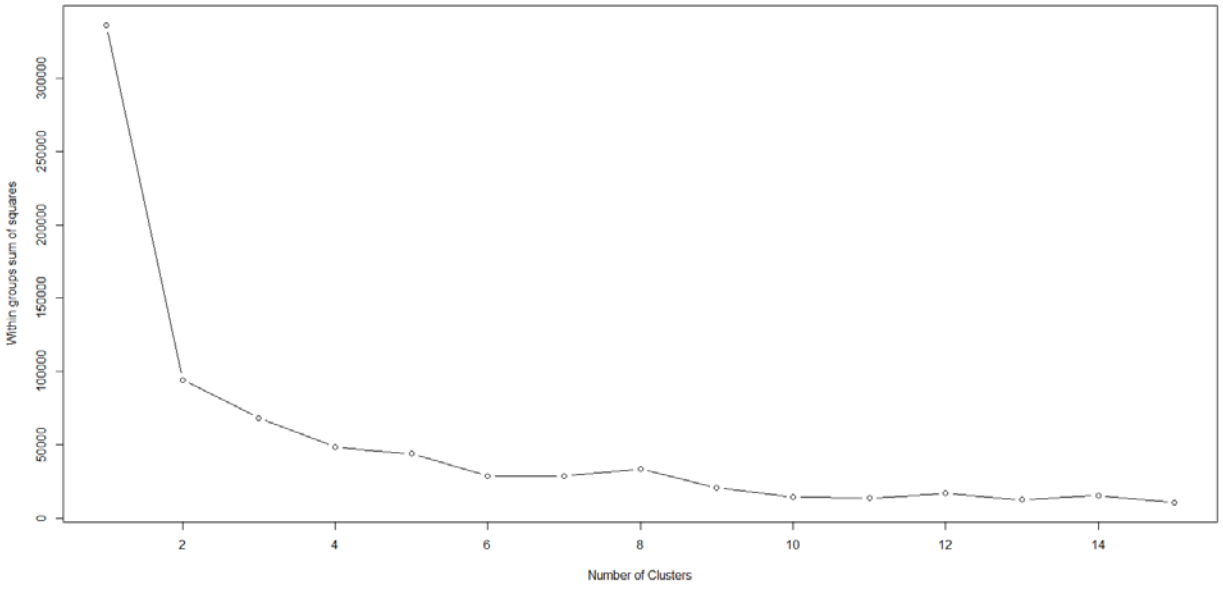


Underlying Variable	Coefficient Value in Linear Combination
Accel Belt Z	0.3944
Accel Arm Y	0.2265
Yaw Forearm	0.1696
Accel Forearm X	0.1636

Number of Components (N)	Percentage of Variance Explained	Reconstruction Error
10	83.80%	6894
20	96.00%	1715.48
30	98.80%	524.33
40	99.70%	142.6
50	99.99%	4.38

K-Means Clustering on PCA-reduced Data Implementation Clock Time: 0.53 seconds

Here, K = 5 is determined to be the optimal number of clusters by finding the "elbow" in the plot of the group sums of squares error by cluster. This seems to agree with how the attributes shown are distributed across clusters as well, as K = 3 does not have much variation in attributes across its clusters.



Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	29%	71%	17071	44.76%	43%	56.92	55.51	43.58	53.87
2	25%	75%	8793	19.78%	23%	27.13	43.88	39.23	51.48
3	30%	70%	13377	35.46%	34%	65.25	59.7	48.96	59.48

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	25%	75%	5842	12.91%	16%	77.93	66.62	50.47	59.79
2	39%	61%	3617	12.57%	8%	82.06	72.56	58.58	70.63
3	28%	72%	7612	19.27%	19%	28.86	38.87	31.15	41.32
4	24%	76%	5273	11.54%	14%	25.97	49.04	48.63	64.69
5	29%	71%	16897	43.70%	43%	57.67	54.8	44.01	53.7

EM Clustering on PCA-Reduced Data – *Implementation Time: 1472.93 seconds*

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	24.48%	75.52%	8152	17.89%	21.92%	29.75	38.55	36.27	50
2	28.16%	71.84%	7364	18.59%	18.84%	26.36	48.57	48.7	63.97
3	16.96%	83.04%	2518	3.83%	7.45%	27.55	42.21	33.42	48.88
4	55.52%	44.48%	1385	6.89%	2.19%	29.21	40.14	36.07	50
5	36.78%	63.22%	2860	9.43%	6.44%	79.19	68.44	51.86	61.4
6	25%	75%	6259	13.95%	17%	79.28	67.8	52.5	60.83
7	27%	73%	6600	16.06%	17%	28.54	35.83	27.42	22.93
8	58%	42%	1380	7.22%	2%	83.38	74.39	74.43	94.77
9	25%	75%	2723	6.15%	7%	83.31	73.4	74.66	94.81

Here, EM seems to capture well the variation between the attributes, though interestingly, there is not much extreme variation in the proportion of positive / negative labels between the clusters. It could be that the attributes shown are important to the underlying structure of the data, but not as important in actually classify the label, which were stripped out for clustering.

Independent Component Analysis Implementation Clock Time: 0.34 seconds

The tables below shows the amount of variance explained by the top three components when ICA was implemented using varying numbers of total components**,** as well as the total amount of variance explained by N components (where $3 \leq N \leq 50$).

Component	3	5	10	30	50
1	18.70%	18.60%	18.60%	17.50%	16.80%
2	18.60%	17.60%	18.20%	6.10%	5.20%
3	12.80%	11.70%	10.30%	5.60%	4.90%

Total Number of Components	Percentage of Variance Explained
3	50.00%
5	67.50%
10	83.80%
15	92.40%
20	96.00%
30	98.80%
50	~ 100%

The below table shows summary statistics for the kurtosis. The higher kurtosis values resulting from more components indicate higher selectivity. Overall, the transformed data is much more kurtotic overall in comparison to the original data because it takes almost as many transformed dimensions as the original dataset to reach the former dataset's kurtosis levels. This indicates success on the part of the ICA, as ICA should increase kurtosis. The median kurtosis of the original data is relatively close to Gaussian at around 2.7 (versus 3.0). It is more apparent that this dataset is more normally distributed than the first dataset.

# Components	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3	1.062	1.121	1.179	1.461	1.660	2.141
5	1.038	1.286	1.566	1.483	1.613	1.914
10	1.037	1.600	1.771	2.480	2.552	6.058
15	1.039	1.802	4.841	4.501	6.211	9.768
20	1.045	1.996	5.363	6.065	8.605	18.370
30	1.191	4.677	8.787	9.584	11.620	45.730
50	1.248	7.509	10.110	54.140	18.150	1281.000
52	1.279	7.513	10.490	175.300	16.450	7648.000
Original Data	1.001	2.168	2.699	96.720	3.738	4759.000

K-means Clustering on ICA-Reduced Data Implementation Clock Time: 0.39 seconds

Here, the k-means implementations find clusters with similar positive / negative label proportions; however, the attributes shown do show variation across clusters, so it seems that k-means is finding subsets of the data that vary according to the attributes deemed important by PCA / Information gain, but do not necessarily corresponding to specific positive / negative labels. EM is better in this case at finding clusters that are more purely positively or negatively labeled.

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	28%	72%	28758	72.32%	74%	57.03	56.75	46.47	54.44
2	31%	69%	7692	21.06%	19%	29.48	39.3	36.23	50
3	26%	74%	2791	6.62%	7%	77.49	70.88	46.09	77.99

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	30%	70%	2246	6.05%	6%	24.6	51.98	51.16	72.12
2	35%	65%	12010	37.56%	28%	33.87	44.2	41.45	50.42
3	24%	76%	5656	11.98%	15%	78.36	69.72	54.2	59.61
4	25%	75%	12124	27.61%	32%	80.11	67.99	52.86	68.7
5	26%	74%	7205	16.80%	19%	28.68	36.9	25.49	31.94

EM Clustering on ICA-Reduced Data *Implementation Clock Time: 1522.56 seconds*

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	33.26%	66.74%	7805	23.27%	18.55%	29.75	38.55	36.27	50
2	0.16%	99.84%	1213	0.02%	4.31%	26.36	48.57	48.7	63.97
3	99.71%	0.29%	692	6.18%	0.01%	27.55	42.21	33.42	48.88
4	37.64%	62.36%	1015	3.42%	2.25%	29.21	40.14	36.07	50
5	32%	68%	9046	25.97%	22%	79.19	68.44	51.86	61.4
6	21%	79%	7260	13.60%	20%	79.28	67.8	52.5	60.83
7	20%	80%	7607	13.42%	22%	28.54	35.83	27.42	22.93
8	23%	77%	3914	7.95%	11%	83.38	74.39	74.43	94.77
9	100%	0%	689	6.17%	0%	83.31	73.4	74.66	94.81

Randomized Projections

RP took only 0.23 seconds to generate reduced dimensions, by far the fastest of the dimensionality reduction algorithms. The reconstruction errors from the result components are much higher than those following PCA.

Number of Components	Reconstruction Error
10	273,755
20	239,075
30	272,356
40	296,679
50	332,990

After running the RP model 100 times, the reconstruction error for 50 components still range between ~ 247,000 and 333,000, with a standard deviation of 27,854. This is most likely for similar reasons to the first dataset, where there are a number of non-normally distributed attributes causing the errors here because of RCA's assumption of a Gaussian distribution when it generates random matrices.

K-Means Clustering on RCA-Reduced Data Implementation Clock Time: 0.74 seconds

In both k-means and EM, the clusters generated are similar to those done on the original data; this is probably due to RCA's ability to preserve the distances between the data points after the data is projected.

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	25%	75%	16445	37.10%	44%	57.03	56.75	46.47	54.44
2	22%	78%	10222	20.51%	28%	29.48	39.3	36.23	50
3	38%	62%	12574	42.39%	28%	77.49	70.88	46.09	77.99

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	24%	76%	4791	10.39%	13%	24.6	51.98	51.16	72.12
2	33%	67%	7385	21.97%	18%	33.87	44.2	41.45	50.42
3	19%	81%	6801	11.62%	20%	78.36	69.72	54.2	59.61
4	30%	70%	14803	39.31%	37%	80.11	67.99	52.86	68.7
5	34%	66%	5461	16.71%	13%	28.68	36.9	25.49	31.94

EM Clustering on RCA-Reduced Data Implementation Time: 1471.83 seconds

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	31%	69%	7641	21.46%	19%	58.5	53.84	43.43	54.29
2	24%	77%	2647	5.57%	7%	30.45	36.11	28.48	49.72
3	20%	80%	7577	13.86%	21%	77.72	65.18	51.47	49.12
4	29%	71%	4384	11.21%	11%	26.94	49.21	50.56	74.73
5	16%	84%	7506	11.07%	22%	32.19	42.45	30.06	36.27
6	38%	63%	2787	9.37%	6%	82.75	73.25	61.29	82.49
7	47%	53%	2089	8.78%	4%	64.47	64.82	48.51	57.44
8	44%	56%	2678	10.49%	5%	30.38	43.48	40.38	49.93
9	47%	53%	1932	8.19%	4%	82.97	72.9	61.59	85.84

Information Gain as Feature Selection Implementation Clock Time: 18.19 seconds

The final feature selection approach uses information gain to assigned importance weights to each attribute. This approach assigned all 52 attributes positive (varying degrees of) information gain, with the top five attributes listed below. These differ from those considered important by the PCA shown earlier in the paper.

Attribute	Information Gain
Total Accel Dumbbell	0.4294
Total Accel Belt	0.4206
Gyros Dumbbell Y	0.4091
Accel Belt Y	0.4002
Gyros Belt Z	0.3981

K-means Clustering on Information Gain-Reduced Data Implementation Clock Time: 0.44 seconds

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	25%	75%	12531	28.07%	33%	52.9	53.72	43.22	51.67
2	36%	64%	10508	33.61%	24%	63.22	60.34	51.06	65.3
3	26%	74%	16202	38.32%	42%	46.66	50.91	41.09	51.49

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	30%	70%	3821	10.34%	10%	29.74	38.62	36.2	50
2	34%	66%	7318	22.54%	17%	53.95	55.34	46.89	61.08
3	25%	75%	12531	28.07%	33%	52.9	53.72	43.22	51.67
4	25%	75%	12303	27.61%	33%	52.19	54.37	42.63	51.96
5	39%	61%	3268	11.44%	7%	82.56	72.69	60.05	74.38

The clusters here do not seem to show extreme variation in positive / negative label proportions, but there is some variation between the attributes across clusters. Again, this could be because these attributes are important to the structure of the data, but other variables play an important part in determining the labels. The EM clustering below is similar, with little extremity between clusters in terms of positive / negative label proportions.

EM Clustering on Information Gain-Reduced Data *Implementation Clock Time: 1295.38 seconds*

Cluster	% Positive	% Negative	Cluster Size	% TP	% TN	Accel Belt Z	Accel Arm Y	Accel Forearm X	Yaw Forearm
1	39.05%	60.95%	3268	11.44%	7.09%	29.75	38.55	36.27	50
2	30.20%	69.80%	3821	10.34%	9.50%	26.36	48.57	48.7	63.97
3	38.61%	61.39%	3419	11.83%	7.47%	27.55	42.21	33.42	48.88
4	31%	69%	3899	10.71%	10%	29.21	40.14	36.07	50
5	25%	75%	6264	13.96%	17%	79.19	68.44	51.86	61.4
6	25%	75%	6039	13.65%	16%	79.28	67.8	52.5	60.83
7	25%	75%	2723	6.15%	7%	28.54	35.83	27.42	22.93
8	24%	76%	6265	13.68%	17%	83.38	74.39	74.43	94.77
9	26%	74%	3543	8.25%	9%	83.31	73.4	74.66	94.81

Dataset Sources

Becker, Barry. (1996). UCI Machine Learning Repository [https://archive.ics.uci.edu/ml/datasets/Adult] Irvine, CA: University of California, School of Information and Computer Science.

Ugulino, W.; Cardador, D.; Vega, K.; Velloso, E.; Milidiu, R.; Fuks, H. Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements. Proceedings of 21st Brazilian Symposium on Artificial Intelligence. Advances in Artificial Intelligence - SBIA 2012. In: Lecture Notes in Computer Science. , pp. 52-61. Curitiba, PR: Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-34458-9. DOI: 10.1007/978-3-642-34459-6_6.