ML Assignment 3

Kuhu Gupta

1 Introduction

The assignment is for exploring unsupervised learning (clustering and expectation-maximization) and dimensionality reduction. The third section gives a detailed examination of clustering algorithms,specifically k-means and expectation maximization. The fourth section covers four dimensionality reduction algorithms, which are,principal components analysis, individual components analysis, randomized projections, and random forest. In the fifth section, after applying these six algorithms on original datasets and observing the results, apply combination of clustering and dimensionality reduction algorithms as well as pipe the results into neural network learner for further examination.

2 Datasets

The datasets used in this assignment are consistent from assignment 1 and 2. The first dataset is the US permanent visa dataset. This dataset is interesting due to its potential to aid in the visa application process. It could also enable confidence in those interested in applying for a US permanent visa but doubting their chances of acceptance. However, the goal is it to try to determine the application result before time, money, and other resources are spent. Similar to the previous assignments, 6 features are used.
The second dataset is a home sale price prediction dataset. This dataset is interesting due to its real-world applicability. Modeling home prices is both a difficult and lucrative task. It can help a person make large amounts of money investing in real estate as well as help in detecting outliers in listed price. Similar to the previous assignments, 11 features are used.

3 Clustering Algorithms

K-means clustering is the first algorithm applied to the datasets and expectation maximization is the second. Both algorithms work by clustering: gathering groups of instances together based upon their features. K-means clustering would do hard clustering where the element either belongs to the cluster or it does not while EM algorithm does soft clustering where clusters may overlap based on strength of association of clusters and instances.

3.1 K-Means Clustering

3.1.1 Overview

K-Means works by clustering n objects into k clusters by using least-squares Euclidean distance between the objects in which each object belongs to the cluster with the nearest mean. This method produces exactly k different clusters of greatest possible distinction by iterative computation. In this assignment, a variety of cluster sizes are tried to find the best parameters possible.

3.1.2 Analysis

On observing the graphs in Figure 1-6, it is clear that on varying the number of clusters clearly impacted the performance of K-Means Clustering. For both datasets, as the number of clusters increases, the clusters are more able to capture the structure of the data and represent the data. However, there are chances that increasing the number of clusters will decrease the accuracy instead of increasing it due to over fitting, such as when just starting out and before convergence (using Euclidean distance for converge properties).
The first measurement is the elbow test which is used to determine the optimal number of clusters based on the sum of squared distance (SSE) between data points and their assigned clusters' centroids. The spot where SSE starts to flatten out and forming an elbow is the point which gives the good number of clusters.As the number of clusters increases, the SSE noticeably drops and then converges. This makes sense because at a certain point, adding more clusters is over fitting and not necessary to get the all training data into its best possible fit. For US Visa data, the elbow point seems to be around 20 as after that SSE curve seems to flatten out. While for housing data it happens around 50.

From the scoring data, it is shown that the permanent visa data, performs remarkably well with a small number of clusters and does not show any noticeable improvement by increasing clusters as seen in Figure 3. This is due to the fact that the permanent visa data is extremely homogeneous and does not contain many outliers at all. On the other hand, the housing price data is much more susceptible to changes in number of clusters. As the number of clusters increases, the testing data gradually increases in accuracy before leveling off as seen in Figure 6. In the plot 6, the accuracy increases around 50. Since the housing data is much more varied and complex, there are intricacies of the data that require more clusters to capture well.

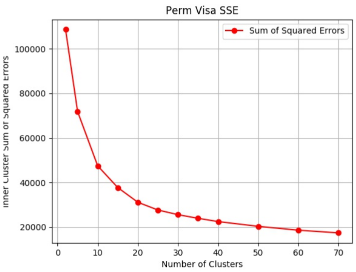

Figure 1: US Visa-Sum of Square Errors Vs. Clusters

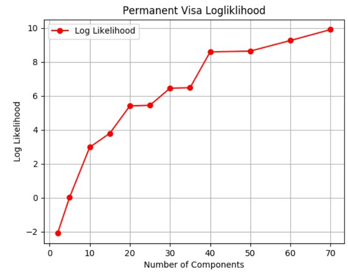

Figure 2: US Visa-Log Likelihood Vs. Components

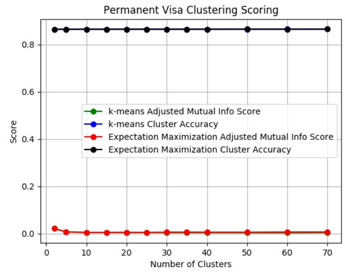

Figure 3: US Visa-Scoring for K-Means and EM

Figure 4: Housing-Sum of Square Errors Vs. Clusters

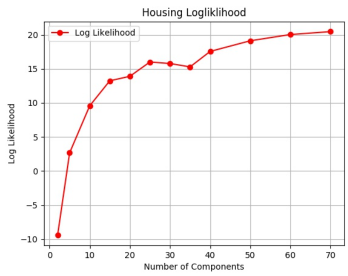

Figure 5: Housing-Log Likelihood Vs. Components

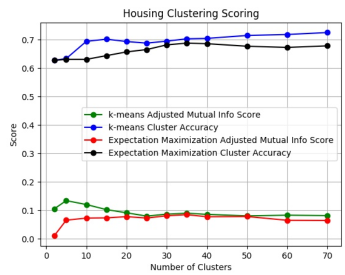

Figure 6: Housing-Scoring for K-Means and EM

3.2 Estimation-Maximization Algorithm

3.2.1 Overview

Expectation Maximization is the second algorithm applied to the datasets is a soft clustering algorithm. The algorithm involves by iterative calculation for the maximum likelihood of parameters leading to a labeling of an instance despite possibly not having all data or parameters. It starts with randomly placed Gaussians and checks for each point whether it belongs to a particular Gaussian or not. Then, adjust the parameters of the Gaussian to fit the points assigned to them. I have used Scikit-learn's Gaussian mixture models to implement the Expectation Maximization algorithm. A varying number of mixture components (or number of distributions) were used to determine the best possible parameters for the clustering.

3.2.2 Analysis

Expectation maximization performed slightly worse than k-means on the datasets. Instead of using a sum of square errors calculation, a log likelihood is calculated to effectively determine the probability of successful labeling. Interestingly, the housing dataset converges quite quickly to a near-peak log likelihood where as the permanent visa dataset takes a bit longer. This makes sense, as the permanent visa dataset is much larger and while an indicator of classification performance and determining factor for component count, it does not guarantee how well the algorithm will perform using such settings. Since the housing dataset is more complicated and contains outliers, however, it's reasonable to expect expectation maximization to perform worse than k-means.

In terms of scoring, k-means performed slightly better, for the housing dataset–though it was insignificantly better for the permanent visa dataset. The adjusted mutual info score, which helps to determine the differences between clusters while accounting for chance, also performs similarly for expectation maximization compared to k-means. Overall, while k-means performed better in our trials, it is reasonable to believe datasets exist that would fare better using expectation maximization. Such datasets would likely be more susceptible to a consistent structure such that maximizing priors is an effective strategy.

Clusters	2	5	10	15	20	25	30	35	40	50	60	70
PERM VISA												
SSE	108717	71834	47453	37701	31090	27611	25517	23874	22410	20267	18532	17331
Log Liklihood	-9.44	2.67	9.57	13.25	13.90	16.01	15.78	15.29	17.56	19.12	20.04	20.47
k-Means AMI	0.022	0.008	0.005	0.005	0.004	0.005	0.004	0.004	0.004	0.004	0.004	0.004
k-Means ACC	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865
EM AMI	0.022	0.007	0.005	0.005	0.006	0.005	0.006	0.007	0.006	0.006	0.007	0.007
EM ACC	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865	0.865
HOUSING												
SSE	14840	10217	8265	7256	6589	6106	5687	5339	5063	4589	4276	4024
Log Liklihood	-2.09	0.03	2.97	3.79	5.41	5.44	6.45	6.48	8.59	8.63	9.26	9.90
k-Means AMI	0.105	0.134	0.120	0.103	0.091	0.080	0.086	0.090	0.086	0.081	0.083	0.081
k-Means ACC	0.628	0.634	0.695	0.702	0.694	0.688	0.695	0.704	0.705	0.715	0.719	0.726
EM AMI	0.010	0.065	0.073	0.073	0.078	0.073	0.081	0.085	0.077	0.078	0.065	0.064
EM ACC	0.628	0.631	0.631	0.644	0.657	0.666	0.682	0.688	0.686	0.677	0.673	0.679

Figure 7: Caption

4 Dimensionality Reduction Algorithms

This section talks about dimensionality reduction algorithms. The four algorithms used are Principal Components Analysis(PCA), Individual Components Analysis(ICA), Randomized Projections, and Random Forests. After running these algorithms on both datasets, an analysis is provided on the results. Later, these results of the dimensionality reduction algorithms will be used in combination with clustering as well as inputs to a neural network.

4.1 Principal Components Analysis(PCA)

4.1.1 Overview

The first dimenstionality reduction algorithm applied is Principal Component Analysis. Given a set of data on n dimensions, PCA aims to find a linear subspace of dimension d lower than n such that the data points lie mainly on this linear subspace. It is a statistics approach for finding d orthogonal vectors that form a new coordinate system, called the 'principal components'. Each subsequent component is found with the intent to be orthogonal to the preceding component. The principal components are orthogonal, linear transformations of the original data points.

4.1.2 Analysis

Principal component analysis seeks to reduce the number of dimensions in the data without sacrificing data quality. The principal component analysis results show that both the eigenvalues and the variance decrease for both datasets as the number of components is increases. For the permanent visa dataset, the size of the eigenvalues is consistent with a slight continuous decrease. In the case that there was a sharp decrease and then level off, it would indicate that there are features that are potentially unnecessary and removable. Since the level off is gradual, it indicates that each feature is important to representing the initial data. While the housing dataset has a sharp very initial drop, it then has a continuous downwards trend for its eigenvalues, also indicating that removing too many features may not be a wise thing to do. The housing data set has a much wider distribution for its eigenvalues than the permanent visa data set–indicating that the importance of features is more skewed for the housing data as well.

The variance graphs also provide interesting views into the ability of PCA to reduce dimensionality without sacrificing data quality. While the housing dataset indicates a higher variance between different features, the permanent visa dataset proves to be more evenly distributed. Since we are trying to maximize variance between the different components (so that we most accurately represent the higher dimension data), we want to choose a number of components that demonstrates such. For the permanent visa dataset, that number appears to be around 5 components and for the housing dataset it appears to be around 9 components.

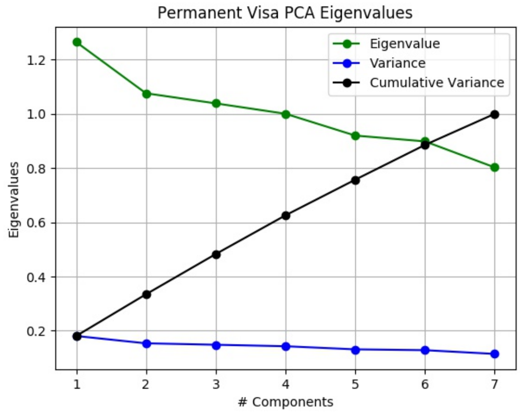

Figure 8: Permanent Visa Principal Components Analysis

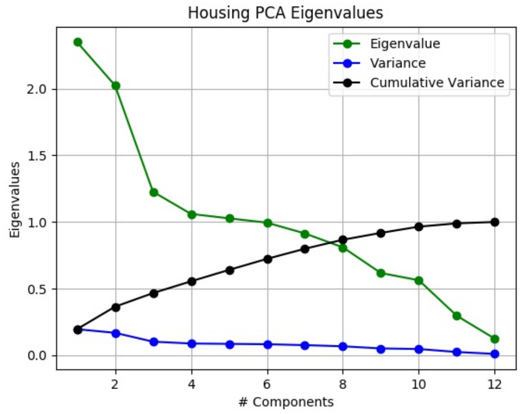

Figure 9: Housing Principal Components Analysis

4.2 Independent Components Analysis(ICA)

4.2.1 Overview

The second dimensionality reduction algorithm, independent components analysis, is an approach to separating a mixture of a data into appropriate sub-components. As discussed in lecture, a good example of what ICA is used for is the cocktail problem; where one needs to separate various sounds into their sources: a tv show, humans, car noises, etc. Kurtosis is used as a measurement of how it gives an indication of the gaussianity of a distribution.

4.2.2 Analysis

While PCA sought to maximize variance, ICA seeks to separate mixed data into subcomponents. As a dimensionality aglorithm, independent components analysis also wants to minimize dimesionality while preserving data quality. Using the kurtosis measurement, we are able to measure the spikiness of the data distribution. It's important to note that kurtosis is sensitive to outliers and therefore not always robust to measuring gaussianity.

The parameter to tune was number of components (dimensions). Similar to PCA, it is observed that the permanent visa dataset is significantly more homogenous then the housing dataset. It also suggest that 5 dimensions be kept for the permanent visa dataset and approximately 5-9 for the housing dataset.

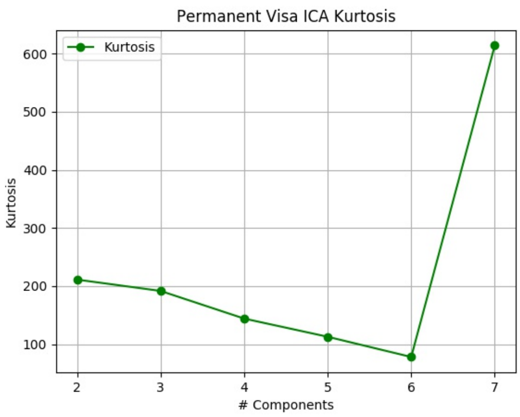

Figure 10: Permanent Visa Independent Components Analysis

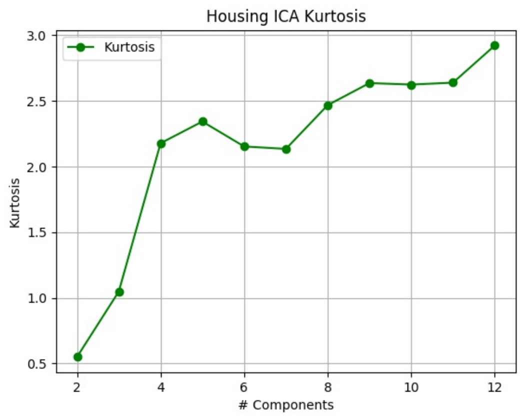

Figure 11: Housing Independent Components Analysis

4.3 Random Projections

4.3.1 Overview

The third dimensionality reduction algorithm is randomized projections. It is an approach that randomly generates a projection matrix that attempts to create a lower dimension representation of the data that is approximately accurate to its original state. By varying the number of components to project, we can run various tests on how well the lower dimension data captures the original. We can also run tests to determine how well the original data can be reconstructed from the lower dimension data.

4.3.2 Analysis

Randomized projections, of all the dimensionality reduction algorithms, was the most susceptible to variation in performance due to its random nature. In such, various trials were run, each varying the random state and maintaining that same random state for a variety of number of components. We ran 10 trials for each dataset (and kept the most relevant 7).

The first measure used to determine the appropriate number of dimensions to reduce to was pairwise distance between the original and reduced data. From the graphs, it is easy to see that the distance (akin to difference in the instances) converges around 5 components for the permanent visa dataset and 9-10 for the housing dataset.

The second, measurement used was reconstruction error. For both datasets, the reconstuction error decreases signficiantly as the number of dimensions is increased. This makes sense as the more dimensions available, the more easily the initial data can be reconstructed. While reconstruction error doesn't give us a clear indicator as to how well a learner will perform on the deconstructed data, it does give insight into how much data is thrown away at each reduction of dimension.

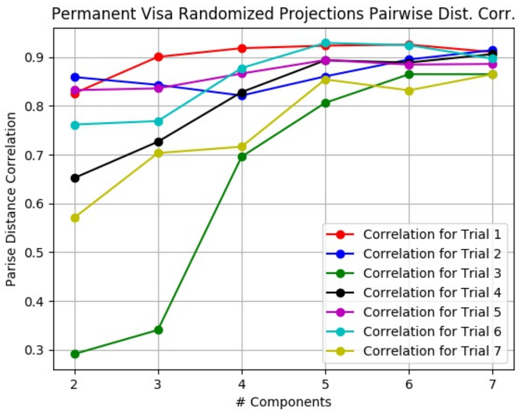

Figure 12: Permanent Visa Randomized Projections Pairwise Correlation

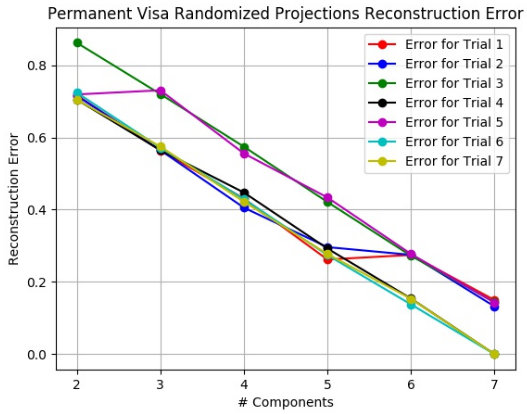

Figure 13: Permanent Visa Randomized Projections Reconstruction Error

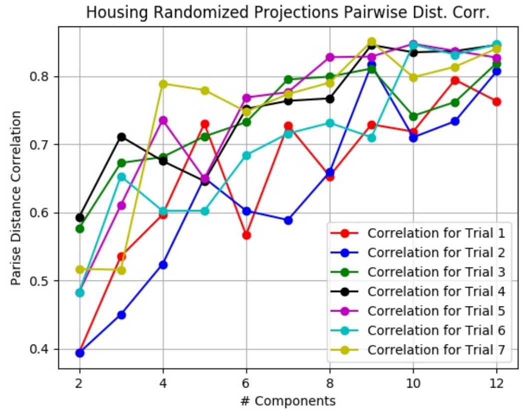

Figure 14: Housing Randomized Projections Pairwise Correlation

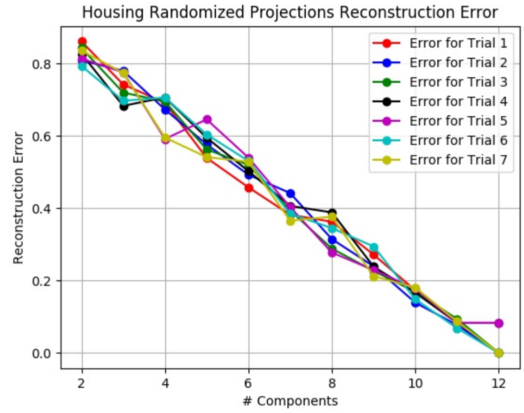

Figure 15: Housing Randomized Projections Reconstruction Error

4.4 Random Forest Feature Selection

4.4.1 Overview

The fourth, and last, dimensionality reduction algorithm is random forest feature selection. It is an approach that uses an ensemble of decision trees conditioned on different features. By training the decision tree and observing the impact of each feature by its ability to classify data correctly, we can select the most important features and disregard unimportant features.

4.4.2 Analysis

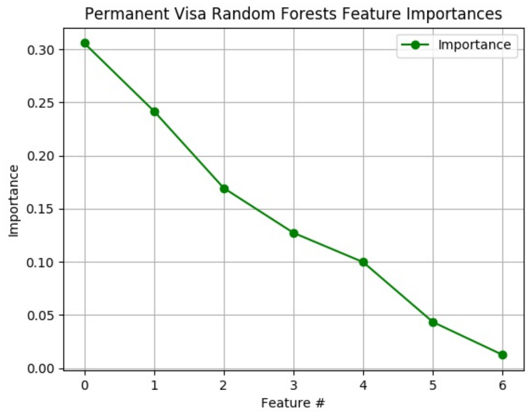

Figure 16: Visa- Random Forest

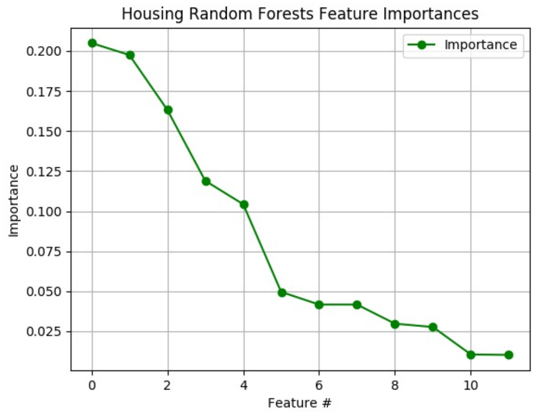

Figure 17: Housing-Random Forest

The last dimensionality reduction algorithm produces results consistent with the others. The number of estimators used was 100 to give a robust learner and all class weights were treated equally. To keep trials consistent, the same random state, number of estimators, and initial class weights were used for each trial of the random forest feature selector.

For the permanent visa dataset, approximately 5 of the features have high importance in terms of a random forest classying data correctly, whereas approximately 9 of the features have high importance for the housing dataset. Interestingly enough, the feature importance graph for the housing dataset indicates that there is a quick dropoff between 5 features in terms of importance. These findings are consistent with the results of the other dimensionality reduction algorithms.

Dimensionality Reduction Inferences

Each DR algorithm tuned its 'k', or number of final dimensions, for each dataset based on its own strategy. By using the various algorithms in tandem, it gives us confidence to safely pick a dimensionality of 5 for the permanent visa dataset and 9 for the housing dataset. As we'll see later, these numbers hold consistent when running grid search on a combination of dimensionality reduction and neural networks.

5 Dimensionality Reduction and Clustering

5.1 Overview

In this section, clustering algorithms are run on the results of the dimensionality reduction algorithms and then compared. All dimenstionality reduction and all clustering algorithms from above are used.

k-Means after Dimensionality Reduction

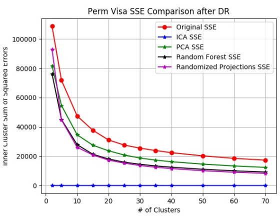

Figure 18: Perm Visa SSE

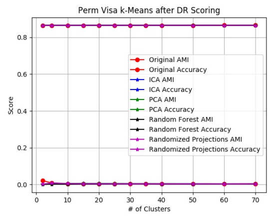

Figure 19: Perm Visa Scoring

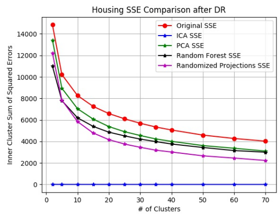

Figure 20: Housing SSE

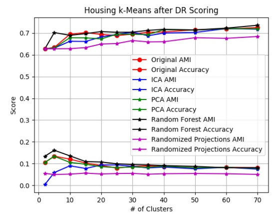

Figure 21: Housing Scoring

5.1.1 Analysis

Above, graphs were computed showing the baseline clustering from Part 1, as well as clustering after each of the dimensionality reductions. Based on the results of part 2, a dimension of 5 was used for the permanent visa dataset and 9 for the housing dataset.

As is immediately apparent, the types of clusters generated vary greatly based on the dimensionality reduction algorithm used. A great way of seeing this is the sum of squared errors for each number of clusters. Each algorithm calculated a different SSE–while part of this can be attributed to the random start of the cluster centers, they do not converge to the same error rates (and thus locations for the testing data).

When looking at the scoring graph, it is observed that k-Means performs roughly consistently after each of the dimensionality algorithms. Other than Randomized projections, which performs slightly worse than the other, the algorithms tend to converge in both accuracy and adjusted mutual information. Similar to before, however, the permanent visa dataset is easily classified and does vary significantly depending on the algorithm used.

For the housing dataset, it becomes clear that random forest feature selection provides the best manner of selecting features. In a way, this makes sense as the random forest feature selector is a robust, generic way of viewing which features contribute the most to classification.

5.2 Expectation Maximization after Dimensionality Reduction

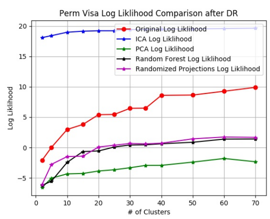

Figure 22: Perm Visa Log Liklihood

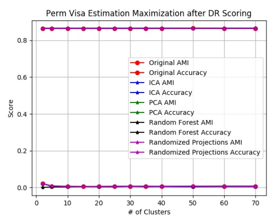

Figure 23: Perm Visa Scoring

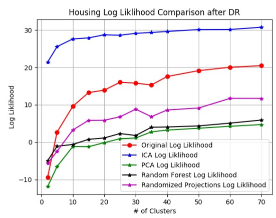

Figure 24: Housing Log Liklihood

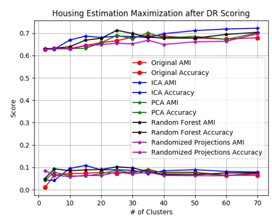

Figure 25: Housing Scoring

5.2.1 Analysis

Similar to k-Means, based on part 2, 5 and 9 were the dimension parameters used for expectation maximization.

Alg	Training Time	Avg Test Score	Avg Train Score	Alpha	Layers	Param
Base						
Base #1	0.0620	**0.7139**	0.7191	0.01	(100,)	
Base #2	0.0605	0.7133	0.7191	0.1	(100,)	
Base #3	0.0685	0.7119	0.7193	0.001	(100,)	
Dim. Reduction						
ICA	0.0274	0.6478	0.6528	0.001	(100,)	2 components
PCA	0.0581	0.7050	0.7048	0.1	(100,)	3 components
Random Forest	0.4450	**0.7291**	0.7482	0.01	(50, 50)	9 dimensions
Randomized Project.	0.0814	0.6940	0.6971	0.1	(50, 50)	10 components
Clustering						
Expectation Maxim.	0.3171	0.6395	0.6290	0.01	(100,)	50 components
k-means	0.2958	**0.7195**	0.7163	0.1	(100,)	50 clusters

Figure 28: Table of Housing Data Results for Cluster

Again, the clusters generated vary signficiantly based on the dimensionality reduction algorithm used. This can be observed through the log liklihood calculations plotted above for each dimensionality reduction algorithm and baseline expectation algorithm. In EM, the log liklihood represents the liklihood of the data being correctly classified. After dimensionality reducing, the number of clusters generally increases this liklihood for both datasets.

Performance results, again, were similar to the k-Means analysis above. One noticeable difference, however, was that ICA proved to be the most succesful algorithm for the more complex housing dataset. While random forests still performed well, they did not stand out as before. Since expectation maximization tries to maximize likilhood and ICA seeks to separate mixed data, this is a viable outcome for scoring. On average, expectation maximization performs slightly worse across algorithms than k-Means, however.

6 Dimensionality Reduction, Clustering, and Neural Networks

6.1 Overview

In this section, similar to part 3, neural networks are run on the results of the dimensionality reduction algorithms and the clustering algorithms, and then compared. The housing dataset is used, since it contains more complex data with more featurse and outliers. Baseline results of simply running the neural network using grid searched parameters on the base dataset are used. For the dimensionality reduction and clustering algorithms, grid search and cross validation were used on paremeters of the neural network and the best result for each algorithm is included below.

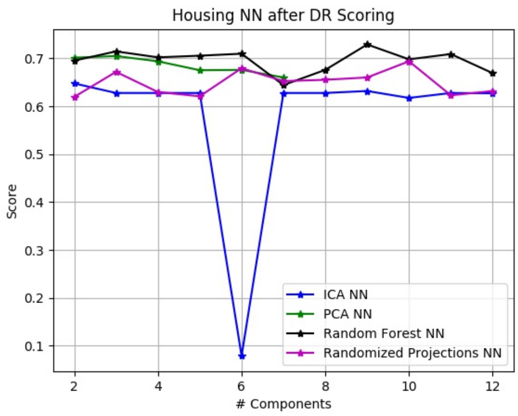

Figure 26: Housing NN after dimentionality reduction

Figure 27: Housing NN after Clustering

6.1.1 Dimensionality Reduction + NN Analysis

For each dimensionality reduction + NN pairing, the best mean test score was selected. When comparing to the base test results, the training time for each dimensionality reduced neural net was faster than the baseline, except for the random forest feature selection and the randomized projection reducer. This makes sense, since both decided to use a different NN layer structure. For the algorithms that grid searched and landed on the same network structure, training time was much faster due to the reduction in data dimensions.

In terms of scoring performance, the best algorithm turned out to be the random forest classifier in both test and training score. As the random forest classifier was able to make informed decisions about which features were not as important to keep around, it was actually able to throw out data that was not relevant (and potentially harmful). In this case, the robustness of the random forest classifier turned out to be exceedingly useful in a dataset that is prone to outliers.

6.1.2 Clustering + NN Analysis

For each clustering algorithm + NN pairing, the best mean test score was also selected. When comparing to the base test results, the clustering algorithms took significantly more time to train (almost 5x). Though still a very small amount of time (less than 1 second)–it demonstrates the resources necessary to calculate across various component parameters. In our case, 50 was found to be the most effective size of component/clusters for both algorithms. This is logical due to the algorithms wanting to find highly-representative structures but not use too many components/clusters so that it overfits and does unuseful work.

In terms of performance, the k-means clustering + NN combo peformed even better than the baseline algorithm, while the expectation maximization algorithm performed significantly worse. This makes sense, as the k-means algorithm is prone to capturing outliers and complex datasets well (especially at a high number of clusters) whereas expectation maximization has a tough time calculating liklihood for complex datasets.

7 Conclusion

In total, 4 different groupings of machine learning algorithsm were run. First, we looked at two clustering algorithms, k-means and expectation maximization, and their impact on input data. Then, we explored four dimensionality reduction algorithms (PCA, ICA, randomized projections, and random forests) in an attempt to reduce the size and complexity of the data without sacrificing important feature information. Next, we applied the clustering algorithms to data that had already been dimension reduced, and found that, on average, doing so increased the clustering algorithms accuracy and decreased its overall datasize. Finally, we combined dimesionality reduction and clustering with neural networks. We found that some cases of these hybrid algorithms were noticably more accurate than the baseline algorithms, and performed better than only dimensionality reduction or clustering/clustering with dimensionality reduction.

Overall, the assignment provides in-depth insight into the benefits of using clustering and dimensionality reduction. By applying the algorithms to two different datasets, an analysis on real-world, concrete data can be handily applied and readily observed.