Project 3: Unsupervised Learning and Dimensionality Reduction

Scott Merrill
smerrill7@gatech.edu

I. Introduction

Unlike supervised learning problems where training occurs on labeled observations, a learner in an unsupervised setting is provided only input features. This paper explores two different unsupervised tasks; cluster analysis and dimensionality reduction. Clustering algorithms attempt to group similar observations into distinct classes while dimension reduction algorithms seek to transform high-dimensional data into a simpler representation that retain the important characteristics of the data.

Two common clustering algorithms considered in this analysis are k-means (KM) and Gaussian Mixture Models (GMM). KM is a hard-clustering algorithm in which each observation can belong to a single cluster, while GMM is a soft-clustering algorithm in which an entry can belong to multiple classes. The properties of each of these algorithms will be explored by analyzing two datasets that vary in size, dimensionality, multicollinearity of input features and outliers. The output produced by each of these algorithms will then be used as features to train a Neural Network (NN) and the performance of each will be analyzed. Based on this performance we comment on the appropriateness of using these clustering algorithms as a means of feature engineering.

Four dimensionality reduction algorithms will also be applied to the two datasets; principal component analysis (PCA), independent component analysis (ICA), random projections (RP) and recursive feature elimination (RFE). The clustering algorithms will again be run on the dimensionally reduced datasets and the performance of each will be analyzed. Finally, a NN will be trained using the dimensionality reduced datasets produced by each of PCA, ICA, RP and RFE. We will finally comment on the appropriateness of each dimensionality reduction technique and consider in what settings certain algorithms may be preferred.

II. Datasets

This assignment retains the NFL Scores Dataset and NFL Play-by-Play Dataset used in Assignment 1; we abbreviate these datasets as Dataset 1 and Dataset 2. While both datasets are related to NFL games, each differs in size, dimensionality, balance of positive examples and number of outliers. Dataset 1 contains scores and betting lines of 6,068 games dating back to 1970. In total there are 34 continuous features and the target label is to determine whether the home team will cover the quoted point spread. The output classification of this dataset is almost perfectly balanced with the home team covering the spread 49.62% of the time. In contrast, Dataset 2 contains both continuous and categorical variables which describe the results of every NFL play from 2010-2019 and the target labels are whether a team will run or pass. In total there are 93,471 entries in the dataset and around 40% of them are running plays and 60% are passing plays.

III. K-means (KM) Clustering

KM groups observations based on their proximity to various cluster centers. The algorithm is thus sensitive to the choice of distance function used. For each dataset, both Euclidean distance (L2 norm) and Manhattan distance (L1 norm) were considered. Both metrics were initially tested by running KM with varying numbers of clusters and it was found that Euclidean distance produced significantly lower average distances to the cluster centers for any number of components. Such isn't a surprising result since the L2 norm by definition computes the shortest distance between two points. In contrast, Manhattan distance simply sums the magnitudes of vectors along each dimension and thus doesn't compute the shortest distance between cartesian coordinates. While the L1 norm is less sensitive to irrelevant features and outliers, each of our datasets are high dimensional and the interactions between various features is unknown. Thus, an outlier in one dimension may be counteracted by an outlier in a different dimension. Given our lack of domain knowledge with respect to feature interactions in each dataset, the Euclidean distance metric is solely considered to prevent adding undue bias to the algorithm.

For each dataset, the elbow method was used to determine the optimal number of clusters. Since increasing the number of clusters will always reduce distortion scores, the appropriate number of clusters to select balances between over and underfitting. The elbow method seeks to find kinks in the curve such that adding more clusters show diminishing marginal reductions to the distortion. The second derivative test is used to identify the point which maximizes the curvature and thus where the reduction in distortion isn't worth the added complexity. Figures 1 and 2 plot the distortion scores curves for Dataset 1 and 2 respectively with the appropriate elbow point indicated.

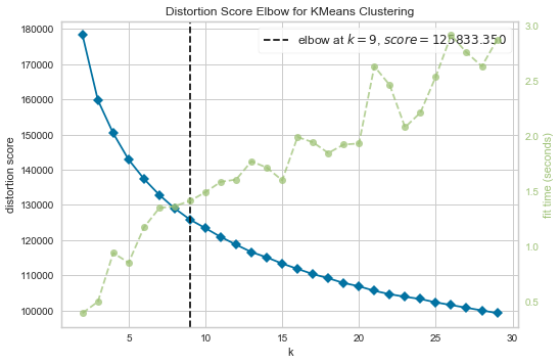


Figure 1

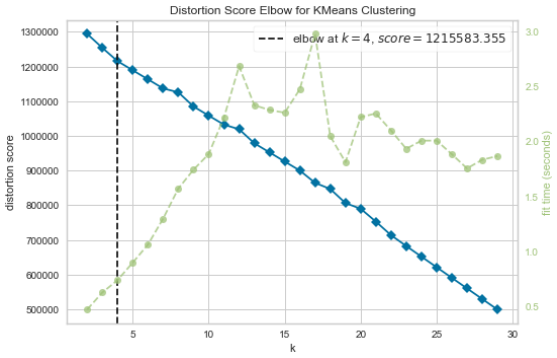


Figure 2

The optimal number of clusters were found to be 9 and 4 for Dataset 1 and 2. While each dataset contains only two output classes, clustering doesn't necessarily try to classify the examples correctly. Rather it attempts to identify sets of observations that are most similar. Ideally, however, these groupings would be correlated with the true output labels.

The silhouette score is used to evaluate the appropriateness of the cluster classes for each dataset. The calculation for the silhouette coefficient is shown in (1) where cohesion is the distance of a point to another in the same cluster and separation is the distance of a point in one cluster to a point in another cluster. Thus, larger values indicate more differentiated clusters, values near zero indicate small margins between decision boundaries and negative values indicate potential misclassification.

$$Silhouette\ Coefficient = \frac{cohesion - separation}{max(cohesion, separation)} \quad (1)$$

The silhouette plots for the Dataset 1 and 2 are shown in Figures 3 and 4 respectively. The average silhouette coefficients of 0.0873 and 0.060 are small for both datasets indicating the clusters identified are close in distance. The silhouette score for the Dataset 1 is almost 50% larger than the silhouette score of the Dataset 2 indicating the clusters groupings are more distinguished. While perhaps more clear decision boundaries, the silhouette score doesn't penalize for complexity; with more clusters, the silhouette score monotonically increases. Other measures to evaluate the degree of similarity between clusters are therefore necessary. Figures 5 and 6 show the parallel coordinates plot for each dataset. From Figure 5 we notice a general correlation amongst the different groupings; datapoints labeled 0 for example all appear to have low values for home overall win %, average home points scored and home win streak and above average values for away team win % away team points scored and away team win streak. Each of the other clusters exhibit similar correlations between features indicating the clustering labels in Dataset 1 are appropriately grouping observations with similar characteristics. Figure 6 shows the parallel coordinates plot for Dataset 2 and displays some correlation between features and clusters; however, the trend is much less convincing. With fewer clusters less similarities amongst features and clustering labels is expected as more compromises will have to be made to group dissimilar observations.

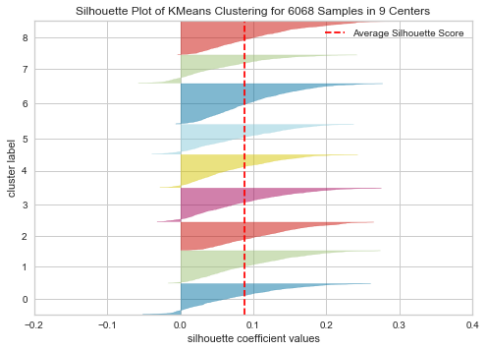


Figure 3

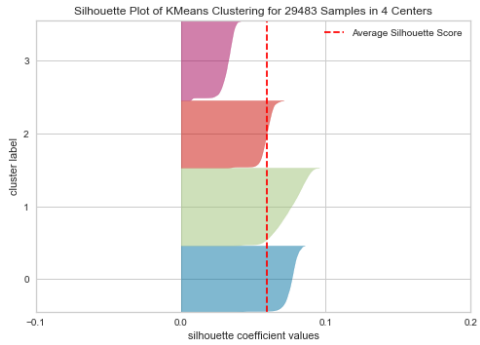


Figure 4

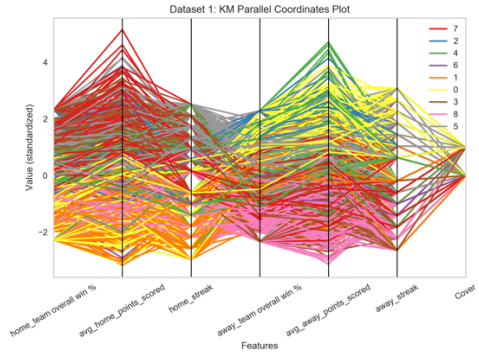


Figure 5

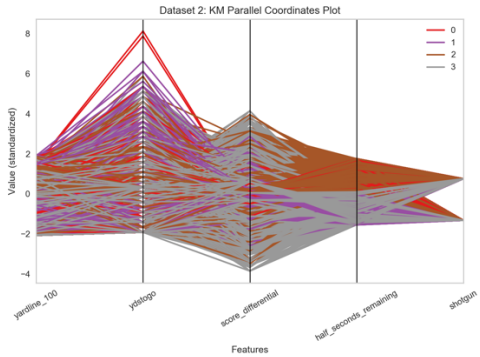


Figure 6

We finally consider external measures that utilize the true labels to evaluate the clustering performance. Figures 7 and 8 show the distribution of the true labels within each cluster for Datasets 1 and 2. While many clusters in Dataset 1 have an equal distribution of labels, 55.1% of the observations in cluster 5 correspond to observations in which the home team covered spread. Given the dataset is balanced and cluster 5 contains almost 650 observations, this correlation appears quite significant. In addition, the clustering distribution far exceeds the accuracy required on Vegas bets to break-even. Recall from Assignment 1, that an accuracy of 52.4% is necessary since these bets typically pay -110 (meaning a $110 bet wins $100). Moreover, the uneven distribution of the true labels in cluster 5 is significant in the scope of the problem. While cluster 5 appears significant, with 8 total clusters, a single cluster with an uneven distribution was inevitable; most of the clusters are in fact roughly evenly distributed. Overall, KM performed poorly on Dataset 1, producing a low silhouette score and clusters that aren't highly correlated with true output labels. The performance may be attributed to high multicollinearity between features. With features that are very closely related, KM is implicitly assigning a lot of weight to the same feature. Moreover, methods like PCA may be effective to summarize the correlation between features into a single component, enabling KM to produce better groupings.

The count of run and pass observations in each cluster in Dataset 2 is shown Figure 8. At first glance, the observations in cluster 1 appear to be highly predictive of a pass play with over 63% of the observations in this cluster corresponding with passes. Cluster 1's distribution, however, is similar to the underlying dataset (which contains 60% pass plays) and thus isn't significantly related to the true output labels. Furthermore, no cluster appears significantly correlated with the output labels. The KM groupings of Dataset 2 is quite poor overall; the margin between decision boundaries is small, the features don't appear correlated with output clusters and the distributions of the output labels in each cluster aren't highly correlated with the true labels. This poor performance can likely be attributed to the irrelevance of most of the 46 features in the dataset. KM suffers from the inductive bias of weighting each of these features equally and with many of these features being irrelevant such resulted in poorly defined clusters. Moreover, reducing the dimensionality of the dataset may eliminate irrelevant features and thus improve cluster classifications.

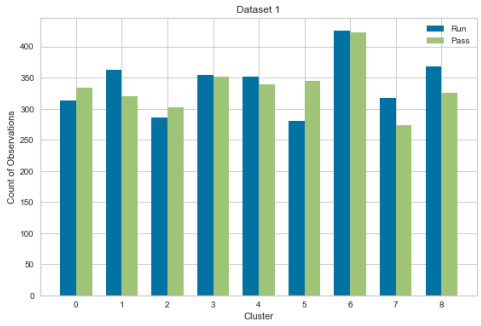


Figure 7

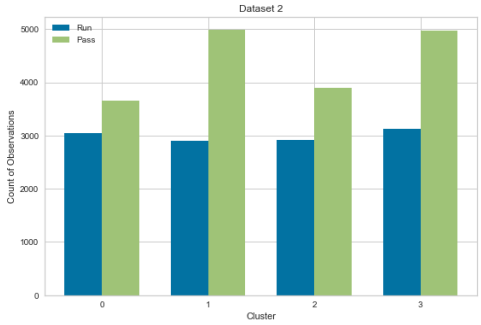


Figure 8

IV. GMM Clustering

GMM is a soft clustering algorithm whereby an observation can be classified into multiple clusters. GMM's use expectation maximization (EM) to fit several multi-dimensional Gaussian distributions to the dataset such that the likelihood of each observation belonging to a particular cluster is maximized. For each dataset, we use the Bayesian Information Criterion (BIC) to select the optimal number of Gaussian distributions or clusters. The BIC attempts to maximize the log likelihood of the data while penalizing complex representations. Figures 9 and 10 show the BIC with respect to varying number of clusters; using the elbow method, the optimal number of clusters for Dataset 1 and 2 are 5 and 7 respectively. These clusters produced silhouette scores of 0.014 and 0.076. The silhouette score for Dataset 1 is lower than that identified using KM, however this may be due to the fact that fewer clusters were used. Similarly, the silhouette coefficient in Dataset 2 is larger than that found using KM, but the metric may be biased as more clusters were used. The low silhouette scores for both Datasets suggest clusters are narrowly separated.

The performance of GMM on each dataset is next evaluated visually using the parallel coordinates plot. Figure 11 and 12 show the parallel coordinates plot for Dataset 1 and 2 respectively. Both figures show little correlation between the clustering labels and specific feature values, which is another indication of poorly defined clusters. Further, from a visual perspective, the clusters appear more ambiguously defined than those clusters found using KM. This makes sense since GMM results in more complex decision boundaries than the spherical decision margins produced by KM. These complex decision boundaries result in GMM implicitly applying non-uniform weights to each feature, resulting in less intuitive interpretations.

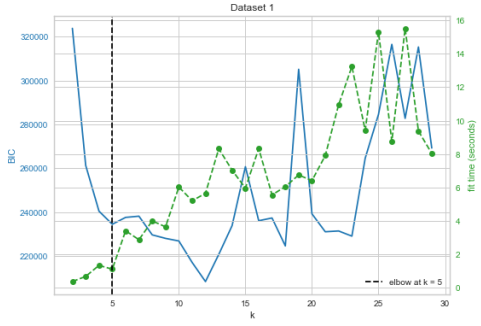


Figure 9

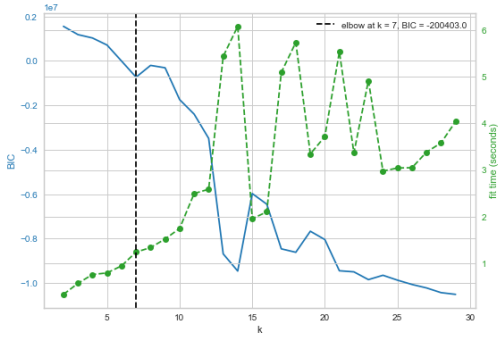


Figure 10

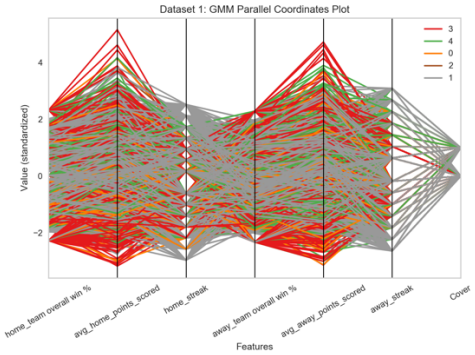


Figure 11

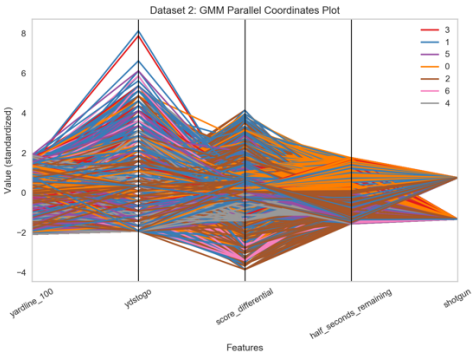


Figure 12

Figures 13 and 14 show the distribution of the true labels within each cluster for Datasets 1 and 2. From Figure 13 we see many more observations falling in cluster 3 than any other cluster. In addition, cluster 2 contains very few labels, indicating one of the Gaussian distributions converged to group a very small subset of the data. The grouping of cluster 2, however appears to be highly correlated with the output labels with 65.5% of the observations in cluster 2 resulting in teams failing to cover the spread. Each of the remaining clusters, however, don't appear highly correlated with the true output labels. Similar to Dataset 1, we note that clusters 4 and 6 in Dataset 2 contain few observations compared to the other clusters. The observations in these clusters may be statistical outliers that are all similar along various dimensions. In addition, observations in cluster 1 appear to be highly correlated with a pass play; 78.7% of the observations grouped in cluster 1 were passing plays, which is quite significant given that the dataset contains a 60/40 split of pass plays to run plays. Many of the remain clusters in Dataset 2 show similar, but less extreme relations with the true output labels.

In contrast to KM, GMM produced clusters with a much wider range of observations in each cluster. This likely occurs due to the close proximity between many of the observations. With observations that are close in distance or even overlapping, the spherical decision boundaries that minimize in cluster loss is biased toward creating more evenly distributed clusters. GMM, however, allows for more complex decision boundaries that doesn't suffer from this bias. While these complex decision boundaries are perhaps more difficult to interpret, the clusters produced by GMM appear to show a higher correlation with the true output labels than KM. Moreover, the choice of clustering algorithm is a natural tradeoff between interpretability and accuracy; if interpretability is important KM is preferred and if accuracy is important GMM may be preferred.

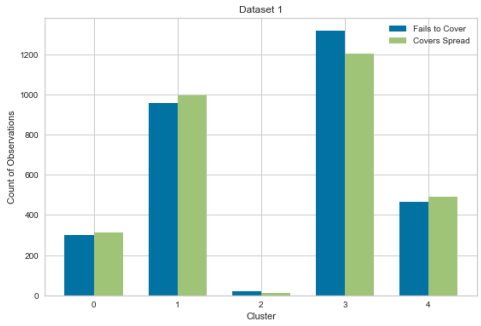


Figure 13

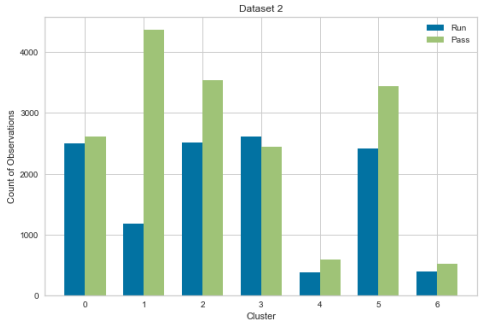


Figure 14

V. PCA

To improve the performance of KM and GMM, PCA was used to reduce the dimensionality of the dataset. Since PCA attempts to identify the best reconstruction and thus minimizing L2 error, the optimal number of components is determined by using the elbow method on the reconstruction error curves. Reconstruction error calculates the difference in original observations after transforming these observations into a lower dimensional space and then decoding and transforming the variables back to the original space. Figures 15 and 16 show the reconstruction error curves for Datasets 1 and 2. Since neither chart produces a clear elbow, we will choose the number of principal components that explains 90% of the total variance in each dataset (or equivalently reduces reconstruction error to 10%). 12 and 36 components are necessary for Dataset 1 and 2 respectively to reduce error to this level. The large percentage of variability explained by the first few principal components and exponential decline in reconstruction error seen in the Dataset 1 occurs due to highly correlated features; home team 4-week average points scored, and home team 5-week average points scored are two separate variables in Dataset 1. These two variables are clearly correlated and can likely be summarized by a single principal component. In contrast, Dataset 2 contains few strongly correlated variables. As a result, the reconstruction error curve decreases more linearly indicating each added component explains a similar amount of variance.

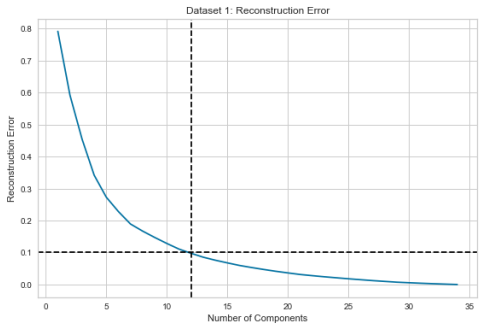


Figure 15

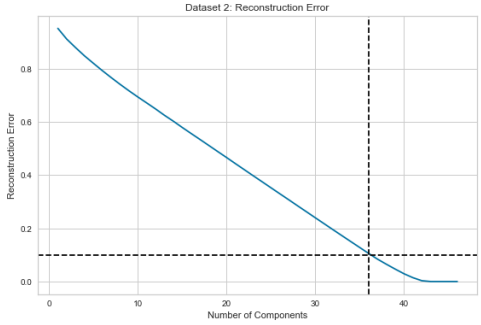


Figure 16

The first 3 principal components explain 55% of the variance in Dataset 1 but only 12% of the variance in Dataset 2. Figure 17 and 18 show a visualization of the first 3 principal components of Dataset 1 and 2 to visualize how they relate to the true labels. Dataset 1 doesn't appear separable in an intuitive way from these 3 components while Dataset 2 appears distinguishable to a reasonable degree of accuracy with 4 clusters. Furthermore, given that Dataset 2 appears more separable with features that explain much less of the data's variability may be an indication of more predicative features or the existence of more noise in Dataset 1.

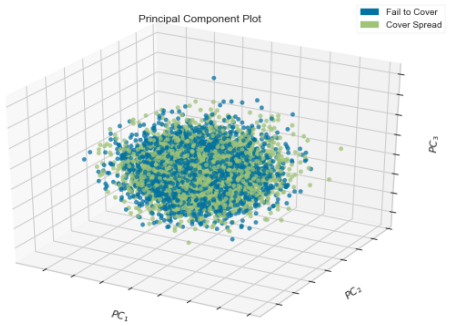


Figure 17

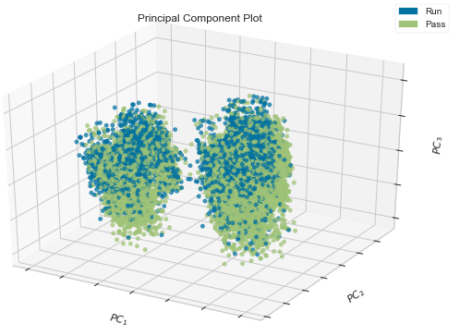


Figure 18

KM was run on each of the dimensionally reduced datasets. The elbow method was used to identify the optimal number of clusters for Datasets 1 and 2 and these were found to be 5 and 32 respectively. The silhouette scores for these optimal groupings were 0.116 and 0.46 which are both improvements over the scores found when KM was applied to the entire dataset. The increase in silhouette coefficient of Dataset 2 was expected as many more clusters were added. In contrast, Dataset 1 used fewer total clusters and still saw its silhouette score increase, indicating the clusters found are more distinguishable and separated by larger margins. To analyze the groupings visually, Figure 19 shows the parallel coordinates plot for Dataset 1. Each clustering label appears significantly dependent on the first 3 principal components. Observations in cluster 2 for example, have above average values for PC0 and much lower values for PC2. Moreover, the improvement in separability of the clusters in Dataset 1 is confirmed visually.

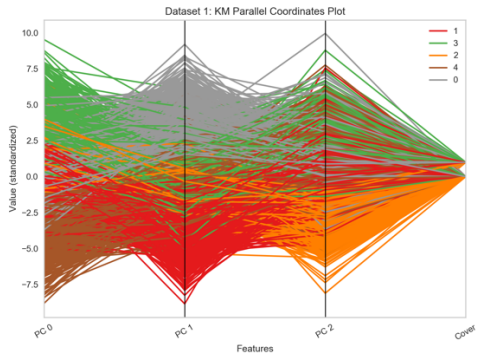


Figure 19

Finally, to evaluate cluster performance using an external measure, Figure 20 shows the distributions of true output labels in each cluster. None of these clusters are strongly related to the true output labels. Cluster 4 appears to be most related as 53.3% of the observations in this cluster resulted in teams failing to cover. Since similar observations don't correspond to similar output labels, such is another indication that Dataset 1 contains a lot of noise and outliers.

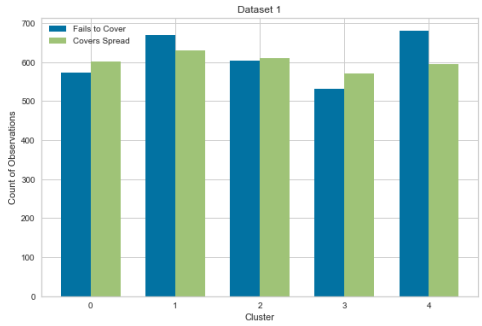


Figure 20

GMM was next applied to the dimensionally reduced datasets and using the elbow method the optimal number of clusters was found to 3 and 29 for Datasets 1 and 2 respectively. The 3 clusters identified on Dataset 1 resulted in a silhouette score of 0.025, indicating the clusters found are quite close to each other and almost non-differentiable. The score is also noticeably smaller than the 0.116 found with KM. The parallel coordinates plot for Dataset 1 is shown in Figure 21 and confirms the indistinguishability of the clusters as there's no clear correlation between PC values and cluster groupings.

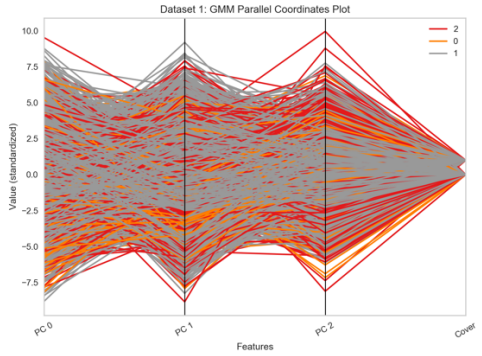


Figure 21

Figure 22 shows the cluster classifications and their association with the true output labels. Each cluster appears to have an even weighting of positive and negative labels and is perhaps more evenly distributed than the clusters produced by KM. Further, clusters 1 and 2 have significantly more observations than cluster 0 indicating that one of the Gaussian distributions converged to classify a small set of potential outliers. The cluster, however, has maximum entropy and provides no information about the true output label.

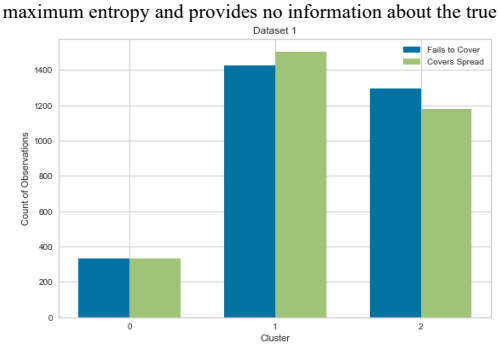


Figure 22

Overall, when applied to the dimensionally reduced datasets, KM produced clusters that were more distinctly separated and correlated with the true output labels than GMM. Such is an interesting result as with a dataset with many outliers, intuiting suggests soft clustering algorithms would outperform. The poor performance by the GMM may indicate that the model converged to a poor local optimum – one in which one of the three Gaussian's converged on a small subset of data that provides no information. Running the algorithm again with different initializations parameters may result in a different local optimum with more favorable properties.

VI. ICA

While PCA attempts to find mutually orthogonal components that maximize variance, ICA attempts to extract features as to maximize independence among extracted features while at the same time maximizing the mutual information between the original and transformed features. Said more simply, ICA attempts to find independent features while retaining the general properties of the underlying features.

Due to the symmetry of the distribution and the fact that the joint distribution of two Gaussian variables is also Gaussian, ICA breaks down when variables follow a normal distribution. We will thus attempt to select the number of independent components as to minimize their "Gaussianality." We use Fisher Kurtosis to measure normality in the transformed variables. Gaussian distributions have a Fisher Kurtosis of 0 and thus the further a variable's Fisher Kurtosis is from 0, the less normal the variable. For each dataset, the kurtosis of the transformed components was averaged over 5 seeds and the

optimal number of components is chosen such a that the average absolute value of Fisher Kurtosis is maximized. Figures 23 and 24 show the average kurtosis with respect to varying number of components. The number of components resulting in the least normality in independent components was found to be 29 and 30 for Datasets 1 and 2.

To further reduce the dimensionality of each dataset, only independent components whose individual kurtosis was larger than 1 in absolute value were maintained. For the Dataset 2, this was all of the independent components, however this technique reduced Dataset 1 to only 5 components. Figure 25 shows a chart of the least Gaussian components in Dataset 1.

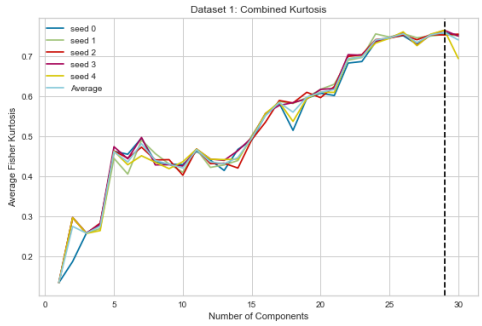


Figure 23

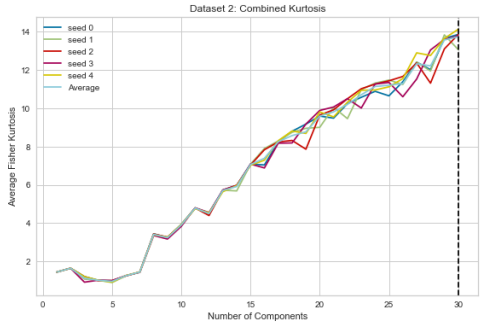


Figure 24

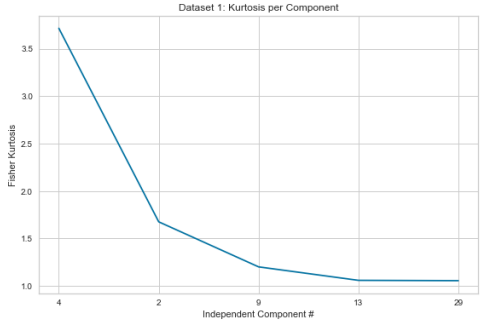


Figure 25

KM was next run on the dimensionally reduced datasets. The elbow method was used to identify the optimal number of clusters and they were found to be 3 and 7 for Datasets 1 and 2. The silhouette scores for these datasets were 0.22 and 0.08. Both of which indicate improvements in terms of separability over the performance of KM on the full datasets, however Dataset 2 did so at the expense of more clusters. The more differentiated clusters produced by Dataset 1 can be visually observed in the parallel components plot in Figure 26. The clustering labels are highly correlated with the independent components, thus providing visual confirmation of increased segregation of the clusters.

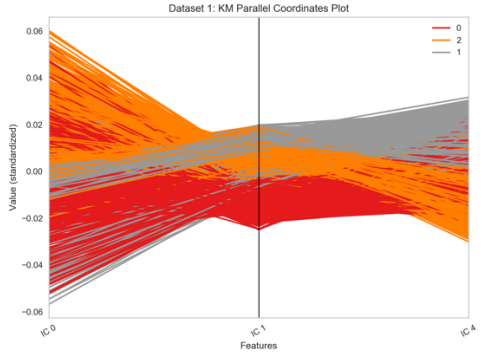


Figure 26

GMM was next run on both dimensionally reduced datasets. Using the elbow method, the optimum number of clusters for Datasets 1 and 2 were found to be 3 and 7. The silhouette scores from the resulting clusters was 0.22 and 0.14. Moreover, both the number of clusters and silhouette coefficients in Dataset 1 were consistent when using both KM and GMM. This may indicate the dimensionality reductions from ICA efficiently separated observations such that they could be clustered in a more consistent and efficient manner.

The parallel coordinates plot of the resulting clusters produced by GMM on the dimensionally reduced dataset is shown in Figure 27. While each cluster appears correlated with the first two independent components, the clusters found using GMM appear more abstract than those found using KM. Further, the correlation of each cluster to the last independent component appears ambiguous. This is consistent with our previous findings about GMM; the algorithm produces more complex decision boundaries that are harder to interpret than KM.

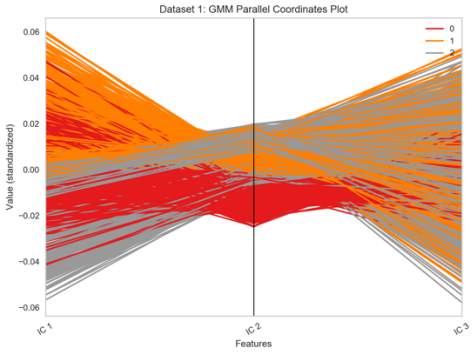


Figure 27

Figure 28 and 29 show the distributions of the clusters produced by KM and GMM for Dataset 1. As can be seen, neither algorithm produced clusters that were significantly correlated with the ground of truth labels. Thus, while the independent components produce larger margins – as indicated by larger silhouette scores – that visually appear to differentiate the data, the groupings found appear unrelated to the ground of truth labels. This again suggests a noisy dataset and underlying features that are poor indicators of the target labels.

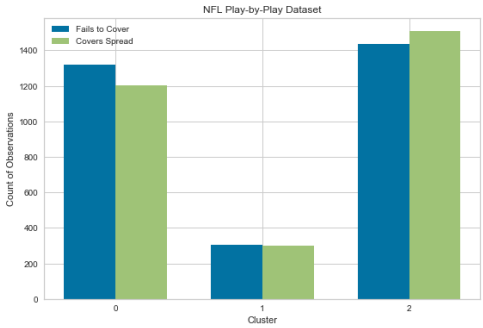


Figure 28

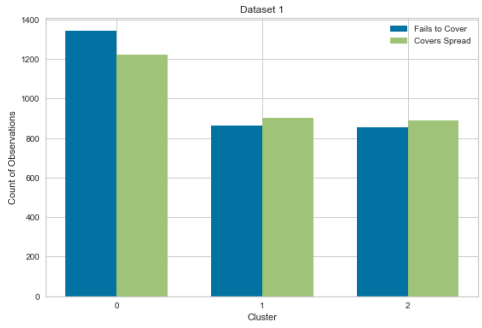


Figure 29

Overall, ICA significantly improved each clustering algorithm on Dataset 1; the identified clusters identified exhibited smaller intra-cluster distances and larger inter-cluster distances. The technique, however, when applied to Dataset 2 resulted in little dimensionality reduction and only marginal improvements in observation groupings with KM and GMM. One potential reason for ICA's poor performance on Dataset 2 is due to the fact that most of the features are uncorrelated. Since ICA attempts to maximize feature independence, little dimensionality reduction will be observed if features are already unrelated. In contrast Dataset 1 contains features that are highly correlated and as such the dataset could be reduced to only 5 independent components. This dimensionality reduction enabled for improved performance and consistency among both KM and GMM.

VII. RP

RP is similar to PCA, however instead of projecting data onto the vector that maintains maximum variance, RP projects data onto randomly generated vectors. To determine the optimal number of random projections we will use the elbow method on the reconstruction loss curve. Figure 30 and 31 show the reconstruction loss curves for Datasets 1 and 2 respectively. Since each curve lacks a clear elbow, we will set a target reconstruction loss threshold of 0.4 to determine the optimal number of projections. That is, we will select the number of random projections such that our reconstruction error is less than 0.4. With this threshold, the optimal number of dimensions for Dataset 1 and 2 were found to be 20 and 27 respectively.

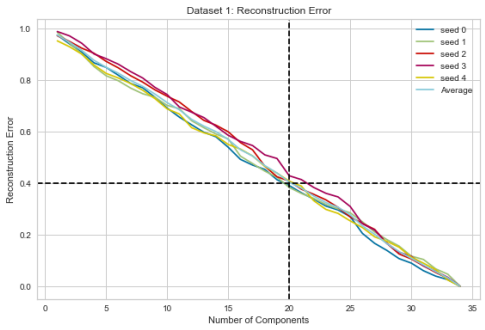


Figure 30

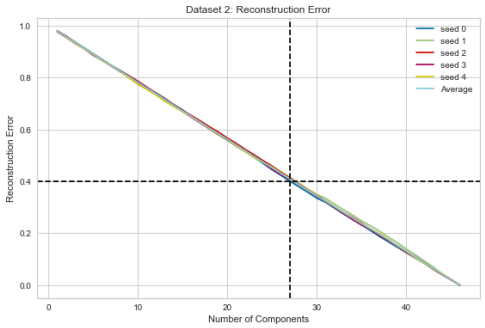


Figure 31

KM was next run on the dimensionally reduced datasets. The optimal number of clusters using KM were 7 and 14 for Datasets 1 and 2. These groupings produced silhouette scores of 0.105 and 0.27 which are both improvements over the coefficients found when KM was run without dimensionality reduction. The silhouette score for Dataset 2, however has many more clusters and an improved score should be expected. In contrast, the silhouette coefficients improved on Dataset 1 with fewer clusters indicating more appropriate and segregated groupings were identified. Figure 32 shows a visualization of the identified clusters for Dataset 1. The observations appear to be significantly correlated with the random components. Cluster 6 for example shows significant positive correlations with RC1 and RC2 and a significant negative relation with RC4. Similar trends are noted with each of the other 6 clusters. Further, these correlations appear to be stronger than when KM was run on Dataset 1 without dimensionality reduction.

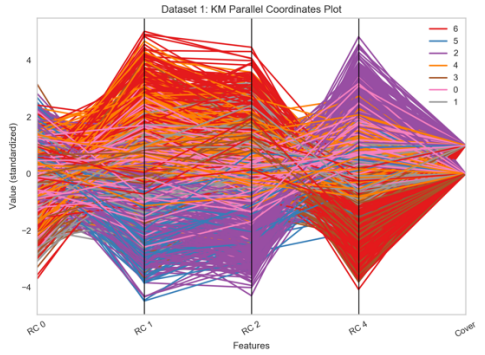


Figure 32

When GMM was run on the dimensionally reduced datasets, the resulting number of clusters and silhouette scores were found to be 2 and 0.05 for the Dataset 1 and 7 and 0.11 for Dataset 2. Thus, GMM when run on Dataset 1 produced the number of output labels consistent with the true number of labels. These groupings however were quite poor as indicated by the low silhouette coefficient. The close and potentially indistinguishable decision boundaries can be noted visually from the parallel components plot in Figure 33; no clear correlation exists between clusters and random components.

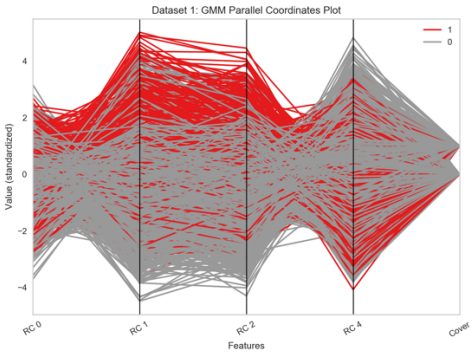


Figure 33

Figures 34 and 35 show the distribution of positive and negative examples in each cluster for KM and GMM respectively. While GMM produced the appropriate number of clusters, the identified groupings aren't materially related to the ground of truth labels. In contrast a few of the clusters produced from KM show promising correlations to the true labels. Over 55% of the observations in cluster 4 for example correspond to observations in which the home team failed to cover. Given the underlying dataset is balanced and there are nearly 900 observations in cluster 4, these results may be significant; more data, however, is needed to further validate the association between cluster 4 and the true data labels.

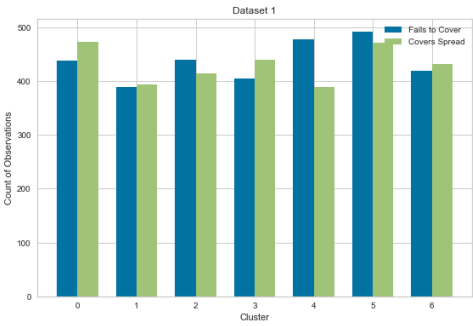


Figure 34

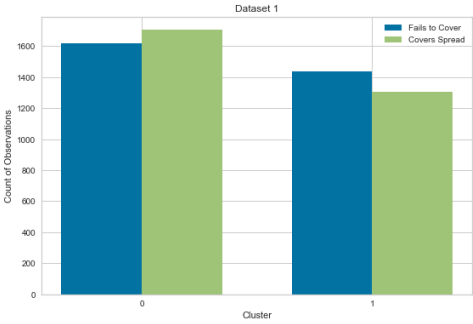


Figure 35

When KM and GMM were applied to the RP reduced dataset, the resulting clusters were comparable in terms of separation to those found when KM and GMM were applied after reducing the dataset using PCA and ICA. The most blatant benefit of RP, however, is apparent when analyzing the computational speed of the algorithm. Figure 36 shows a comparison of the training time of RP relative to PCA and ICA on Dataset 2. The speed of RP dominates both PCA and ICA and these improvements appear linear in the number of components. While PCA must calculate the orthogonal projection that maximize variance and ICA makes calculations to maximize independence and mutual information, RP projects data onto randomly generated vectors and thus is much more computationally efficient. Moreover, given the comparability of results and the computational benefits, RP is likely preferred when time is a significant constraint.

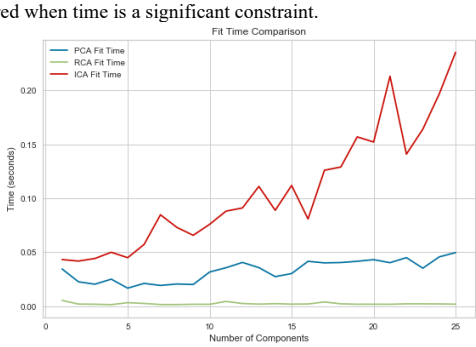


Figure 36

VIII. RFE

RFE was next used as a dimensionality reduction technique on both datasets. RFE requires a supervised learning classifier to rank the features in terms of importance. The algorithm recursively eliminates the least important features until an optimal number of features is found. For each dataset, a decision tree was used to determine the best features; the optimum hyperparameters for these decision trees on each dataset were found in Assignment 1 and were maintained in this assignment.

When ranking the features on Dataset 1, the most important feature was found to be the away teams win percentage. Figure 37 shows the frequencies of the true output labels with respect to away teams win percentage. With larger values in this variable, the determination of whether the home team will cover is quite ambiguous. However, when the away team's win percentage falls below 0.3, the home team fails to cover more often than not. Such might indicate that when the road team is poor, the Vegas point spreads are quoted too large. The significant correlation seen in lower values of the away team win percentage is likely the reason, the feature was found most important. The worst two features for the dataset were found to be home wins and away wins, which likely occurred due to the fact that the information provided by these variables is already captured by the home team win percentage and away team win percentage features.

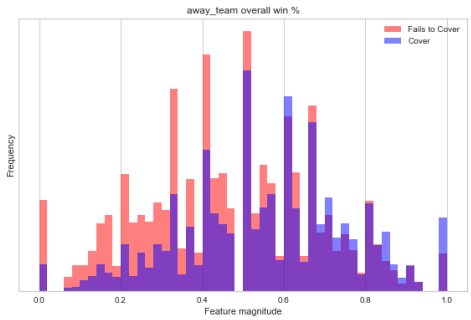


Figure 37

To determine the optimal number of features for each dataset, the cross-validation score of the decision tree trained on different feature set sizes was calculated and the highest score was selected. For each dataset, the optimal number of features was found to be 4. Figures 38 and 39 show the cross-validation scores of each dataset with respect to varying number of features. The validation scores in both datasets initially increase before peaking at 4, indicating the addition of more variables caused the decision tree to overfit and thus reduced generalization accuracy.

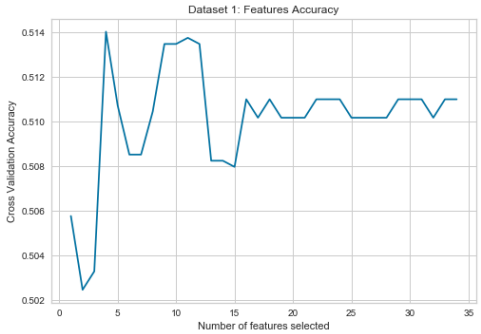


Figure 38

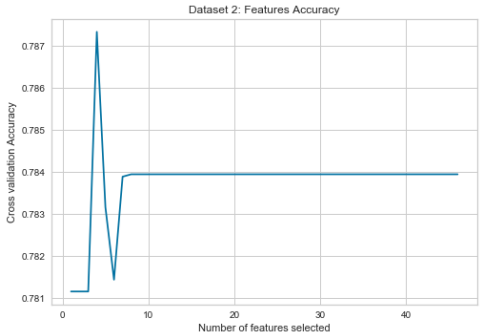


Figure 39

KM was next applied to each of the dimensionally reduced datasets. Using the elbow method, the optimal number of clusters for Datasets 1 and 2 were found to be 2 and 11 respectively. These values resulted in silhouette coefficients of 0.32 and 0.43. Each of these scores are significantly larger than those found when KM was run without dimensionality reduction and indicates more distinct groupings. These vast improvement of the silhouette scores may be explained by the inductive bias of KM to weight each feature equally. With very high dimensional data and many irrelevant features, this bias prevented the identification of good clusters. Moreover, using 4 of the most relevant features avoids this bias as it gives no weighting to irrelevant features.

While the silhouette score is almost 0.5 for Dataset 2, the features don't appear significantly related in the parallel coordinates plot as seen in Figure 40. The identified groupings, however, are significantly related to the ground of truth labels in the dataset. Figure 41 shows the frequency of Run and Pass observations in each cluster. As can be seen, over 90% of the observations in cluster 7 are passing observations. In addition, 75.5% of the observations in cluster 3 are run examples which is particularly impressive since most of the observations in the dataset are pass plays (60%). Many of the other clusters exhibit similar correlations with the true labels and thus would likely be good features to incorporate into a supervised learning model.

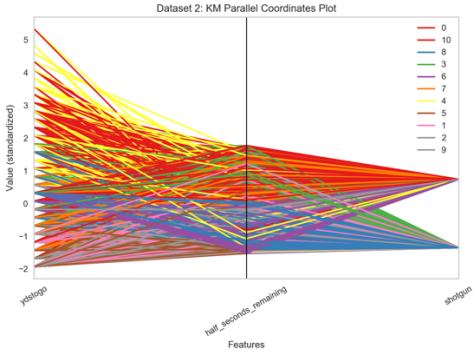


Figure 40

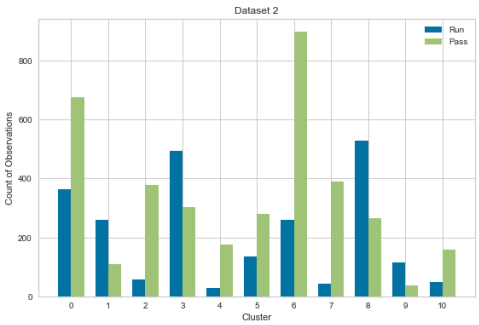


Figure 41

GMM was finally run on each reduced dataset and it was found that the optimal number of clusters was 2 and 4 for Datasets 1 and 2 respectively. The resulting silhouette scores were 0.315 and 0.409 which are significantly larger than those observed when the algorithm was run on without dimensionality reduction. Thus, the clusters identified using GMM on the reduced dataset are much more clearly defined. This can be seen visually in Figure 42 which shows the parallel coordinates plot of Dataset 1; feature values and clustering labels shows significant correlation. The clustering labels, however, don't correlate with a team actually covering the point spread as seen in Figure 43. Since similar observations don't result in similar outcomes, the dataset likely contains considerable outliers.

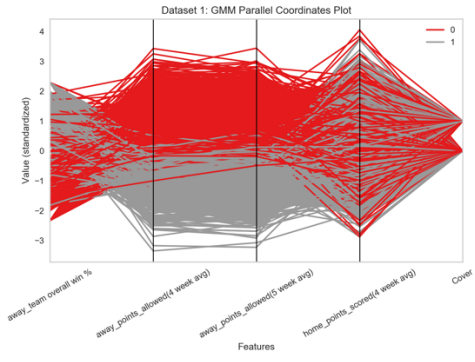


Figure 42

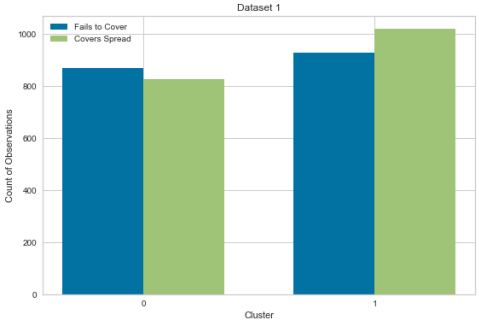


Figure 43

RFE effectively reduced two very high dimensional datasets into smaller ones containing only 4 features. The reduced datasets provided significant benefits when KM and GMM were used to cluster the observations. The clusters identified were separated by wide decision boundaries and were overall more appropriate than the groupings found by running KM and GMM on the datasets reduced using PCA, ICA and RP. RFE, however, suffers from the fact that it requires a supervised learning model – in our case, a decision tree – and thus is inappropriate if the true classes of the data are unknown.

IX. Neural Network (NN) Training

We next look to fit a NN to each of the dimensionality reduction algorithms on Dataset 1. A NN was fit to the dataset in Assignment 1 using all 34 features and it was determined that the best architecture consisted of a single hidden layer of size 3; this same architecture was maintained when training the classifier on the dimensionally reduced datasets. In addition, for each of the dimensionality reduction algorithms, KM and GMM clustering were applied. For KM, the resulting distances to each cluster mean were used as features to train the NN. For GMM, the resulting probabilities of each observation belonging to a particular cluster were used as features. These features were scaled to have a mean of 0 and a unit variance.

A. No Dimensionality Reduction

A NN was first optimized using random search with 100 iterations to find the optimal NN parameters on the full dataset. Figure 44 shows the learning curve for the optimal NN. From the learning curve we note both high variance as indicated by the divergence of the in and out-of-sample accuracies as iterations increase. In addition, the model suffers from high bias with peak accuracy not even reaching 50.5%. As noted in Assignment 1, a NN may not be the most appropriate supervised learning model for Dataset 1 given the limited data (with only 6k entries) and amount of noise present; decision trees and support vector machines were found to produce most accurate results on this dataset.

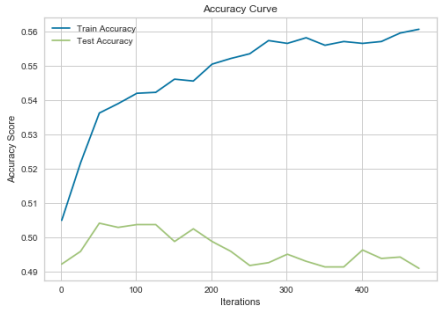


Figure 44

KM was next applied to Dataset 1 and the distance of each observation to the cluster means were used as features to train the NN. Figure 45 shows the test accuracies achieved with varying number of clusters; these test accuracies are averaged over 10 different random seeds. The optimal test accuracy occurs when 9 clusters were used which resulted in an average test accuracy of over 51%. The accuracies appear highly sensitive to the number of clusters as can be seen by the many local optima and minima. However, there appears to be a sweet spot between 9 and 17 clusters where performance is best; any fewer clusters results in underfitting and thus higher bias. More than 17 clusters appear to result in overfitting as average test accuracy appears to degrade and variance increase.

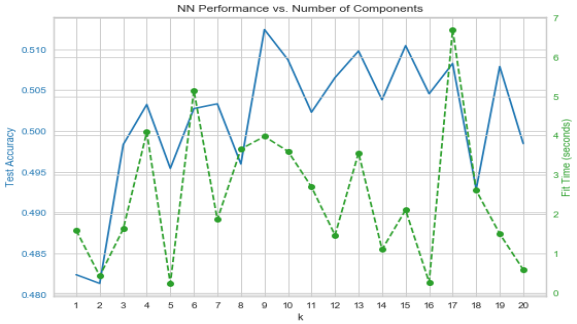


Figure 45

GMM was next applied to Dataset 1 and the probabilities of each observation belonging to a particular cluster were used as features to train the NN. Figure 46 shows the average test accuracies achieved using an optimized NN trained on these probabilities. Peak test accuracy of 51.7% occurs when 11 clusters are used. On average more clusters appear to have better accuracies than fewer clusters. This might be due to the fact that with more clusters, observations are likely more clearly distinguishable and perhaps have better correlations with the true output labels. In turn, this yields more appropriate features and better performance when a NN is trained on these features.

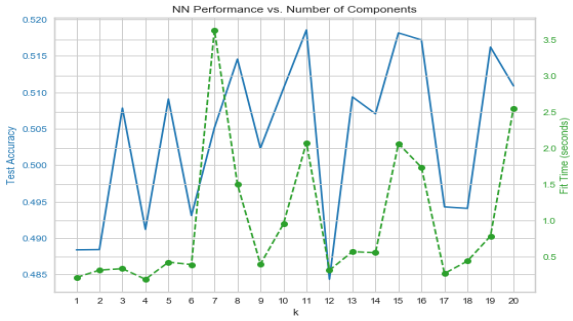


Figure 46

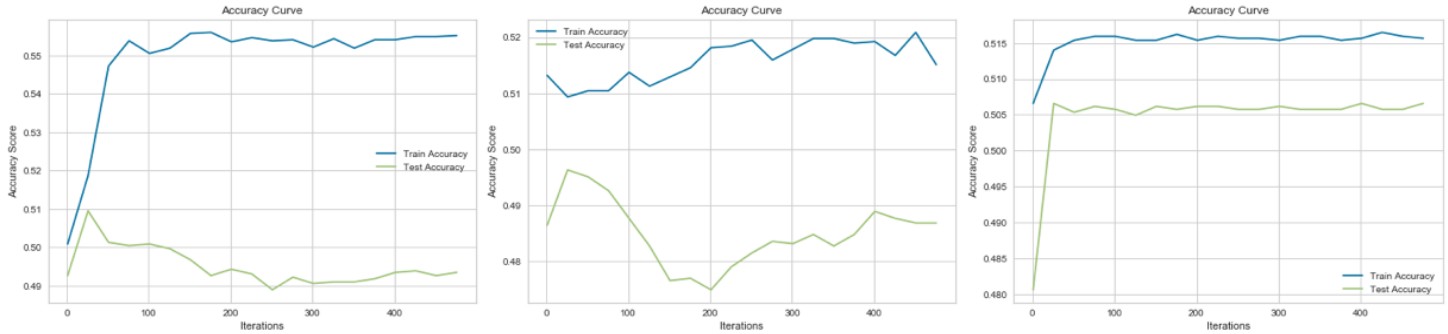
B. PCA

We saw previously that KM and GMM identified 5 and 3 clusters respectively when run on the dimensionally reduced dataset with 12 principal components. Moreover, we trained 3 NN's; the first used the 12 principle components, the second used the distance of each observation to the 5 cluster means from KM and the last was trained on the probabilities of each observation belonging to the 3 clusters identified by GMM. Each NN was optimized using random search with 100 iterations to find the best hyperparameters. The learning curves of these three NN are shown in Figure 47. The NN trained using the PCA appears to exhibit both less variance and less bias than the NN trained using all features. Since the 12 principal components account for 90% of the variance, the improved performance may indicate that the 10% of the variance not captured by these components is simply noise in the dataset. Thus, not fitting to the noise results in both less variance and improved generalization.

When trained using the features from KM, the model appears less variant, but produces significant bias. The reduction in variance can be explained by the simplicity of the model as it contains only 5 features. The increased bias is likely the result of clustering groupings that aren't highly correlated with the true output labels. This is evident by noting the cluster with the lowest entropy is cluster 4 in which 53.3% of the labels in that cluster correspond to teams failing to cover. The remaining clusters have a more equal weighting and provide even less information.

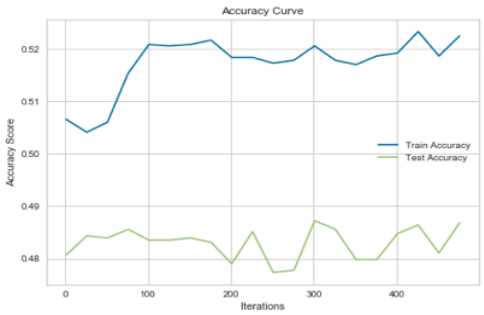
The NN trained using GMM probabilities is shows the least variance out of all the PCA models as indicated by the close proximity of the training and testing accuracy curves; this is as expected since the model is least complex. Surprisingly, the bias appears lowest in the GMM NN which might indicate that the observations in each Gaussian distribution were correlated with the true output labels.

Figure 48 shows the test accuracy of the NN trained with varying numbers of principal components. Random search with 100 iterations was used to identify the best hyperparameters for each set of components and test accuracy is averaged over 10 random seeds. From the graph, 8 principal components maximized the test accuracy which contrasts the 12 principal components used in our previous analysis. The 12 principal components used account for 90.2% of the variance in the dataset while the first 8 principal components account for 85.3%. Moreover, the additional 4 variables that account for only 5% of cumulative variance likely capture a lot of the noise present in the dataset. Thus, their inclusion results in a NN that overfits to the noise and doesn't generalize well. This illustrates the importance of selecting an appropriate variance threshold when determining the appropriate number of principal components for inclusion. While our threshold of 90% was too large, the optimal threshold likely depends on the amount of noise in the dataset. For datasets with fewer outliers, larger thresholds are likely preferred. And, for more stochastic datasets, lower thresholds should be considered.

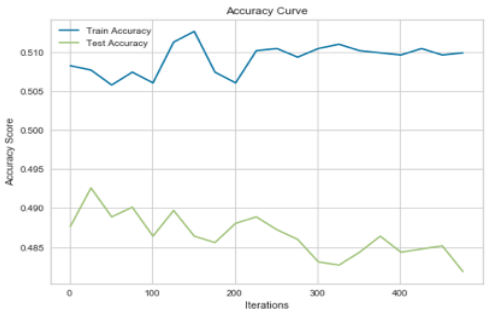


(a) PCA components *(b) KM Distances* *(c) GMM Probabilities*

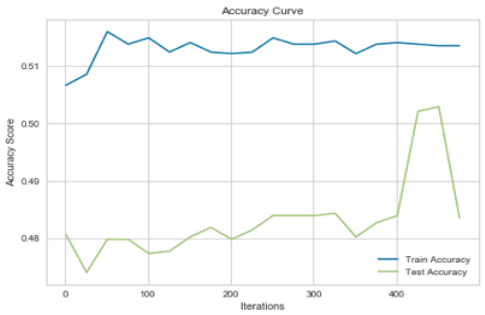
Figure 47



(a) ICA components



(b) KM Distances



(c) GMM Probabilities

Figure 49

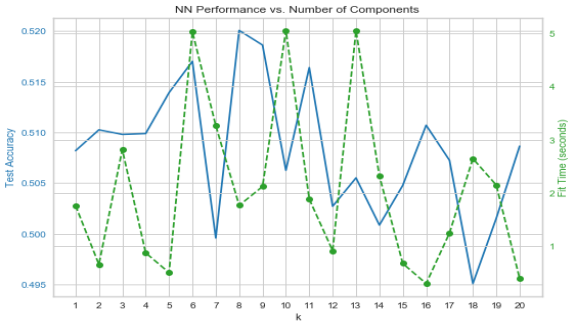


Figure 45

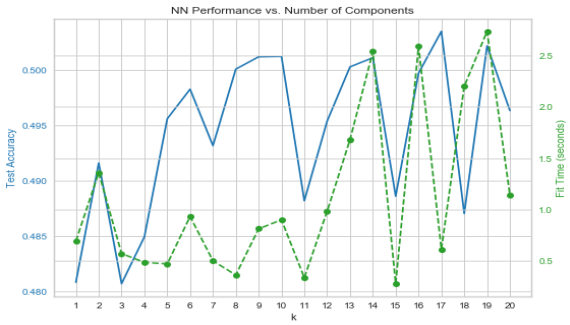


Figure 50

C. *ICA*

Recall that for Dataset 1, 5 independent components were found to be optimal and such resulted in 3 clusters when KM and GMM were applied. A NN was trained on the 5 independent components as well as the output means and probabilities of KM and GMM. Each NN was optimized using 100 iterations of random search and the learning curves for these NN are shown in Figure 49.

The NN trained using the independent components shows less variance than the NN trained using all the features as indicated by less deviation between the training and testing accuracy curves. In addition, the classifier appears to have a similar amount of bias. Moreover, a similar accuracy is achieved using 29 fewer features highlighting the fact that many of Dataset 1's features lack significant predictive power. The NN trained using the distance to the KM cluster centers exhibits a further increase of both bias and variance. This conclusion is consistent with the behavior observed when a NN was trained using the cluster means from PCA and is likely due to low correlation between these clusters and the true output label. The test accuracy of the NN trained using the probabilities from GMM clustering exhibits an increasing trend with more iterations while the training accuracy remains constant. The inconsistent spike in test accuracy after 425 iterations is a testament of the variance remaining in the model. The general upward trend in test accuracy, however, is intriguing and may indicate improved performance with more iterations and data.

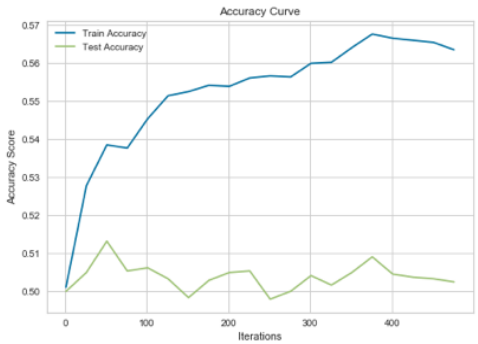
Figure 50 shows the test accuracy achieved by a NN with varying independent components. Random search with 100 iterations was used to identify the best hyperparameters for each set of components and test accuracy is averaged over 10 random seeds. While 5 components appear to produce a local optimum, test accuracy is optimized using 17 independent components. The test accuracy with 5 components, however, has performance only slightly worse than when 17 independent components are used; the increased complexity from adding 12 additional input units may not be worth the marginal improvement in test accuracy.

D. *RP*

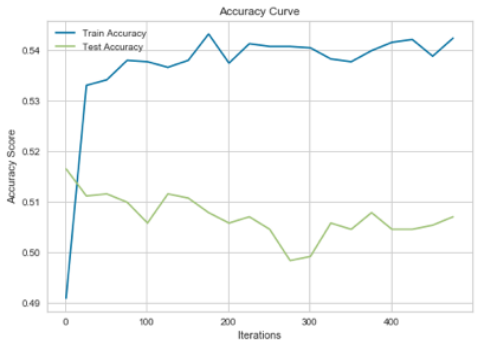
The optimal number of random projections was found to be 20 and when KM and GMM were used to group the resulting observations, 7 and 2 clusters were identified respectively. A NN was trained with the dimensionally reduced dataset as well as with the outputs from KM and GMM. Each NN was optimized using random search and the learning curves for the models are shown in Figure 51.

The test accuracy of the NN trained using RP appears to dominate the accuracies of both PCA and ICA. This, however, may simply due to the fact that 20 RP were used compared to only 12 principal components and 5 independent components. The larger features set enables for more complex decision boundaries and thus less bias at the expense of more variance. The NN trained on the outputs from KM show less variance as observed by the close proximity between training and testing accuracy curves; this makes sense as the model contains fewer parameters resulting in a less complex model that is less sensitive to training examples. The NN trained using the outputs from GMM performs incredibly well achieving consistent test accuracies over 51.5%. The model also exhibits little variance which is as expected due to the model's lack complexity. The encouraging performance likely indicates a reasonable correlation between produced probabilities and true output labels.

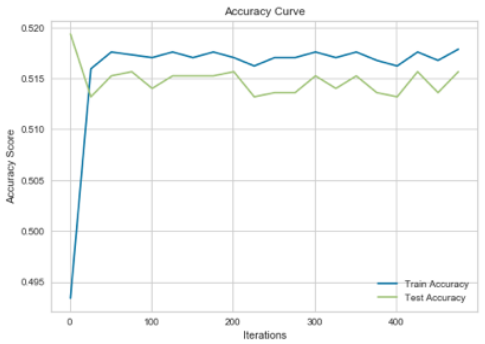
Figure 52 shows the test accuracy of a NN trained with varying number of RP on Dataset 1. Random search with 100 iterations was used to identify the best hyperparameters for each set of components and test accuracy is averaged over 10 random seeds. In our previous analysis, we determined the optimal number of random projections was 20, however the number of components that resulted in the best test accuracy was 15. Recall that we selected the optimal number of random projections by assigning a reconstruction loss cutoff of 40%. Reducing the dataset to 15 components would have required the acceptance of a reconstruction loss of approximately 55%. This might indicate that our selection for our loss cut off may have been inappropriate considering the amount of noise in our dataset. Furthermore, with 5 more random components, our reconstruction loss likely decoded the noise present in the dataset resulting in poor out of sample generalization.



(a) RP components

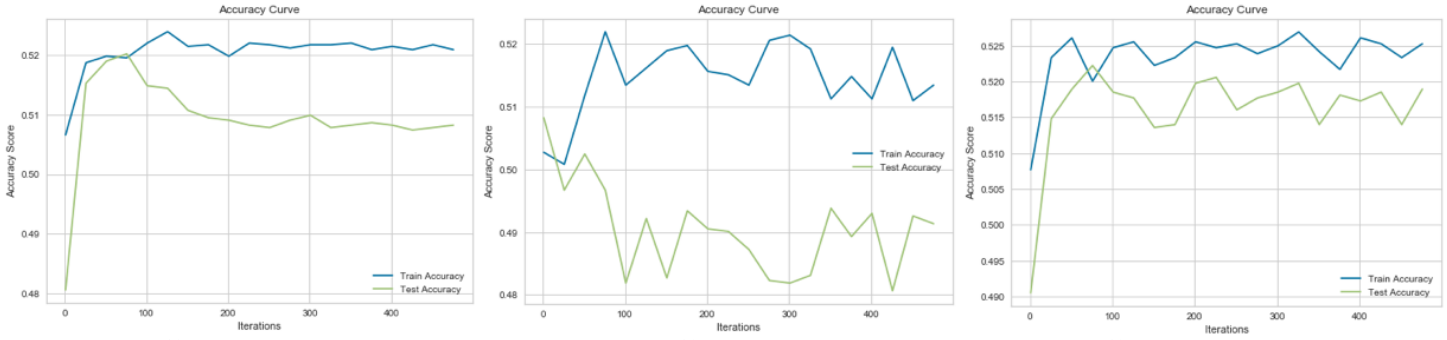


(b) KM Distances



(c) GMM Probabilities

Figure 51



(a) RFE components *(b) KM Distances* *(c) GMM Probabilities*

Figure 53

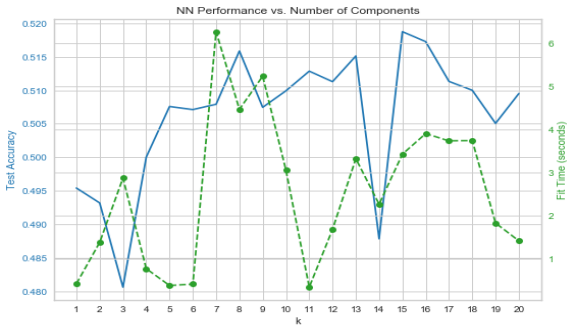


Figure 52

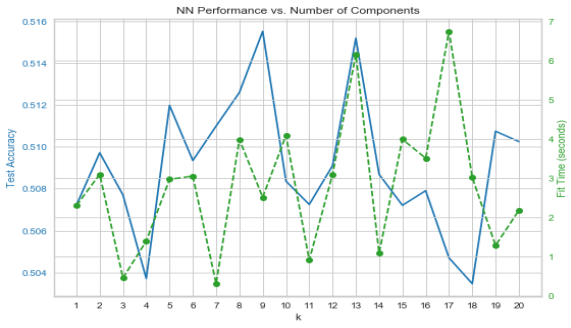


Figure 54

E. RFE

A NN was finally trained using the 4 highest ranking features resulting from RFE as well as on the outputs from KM and GMM when these algorithms were applied to the reduced dataset. Random search was used with 100 iterations to find the optimal hyperparameter for each classifier and the learning curves for each of these models is shown in Figure 53.

The NN trained on the RFE features exhibits low variance and test accuracy that outperforms each of the previous dimension reduction algorithms. This indicates that using only the 4 top ranking features can appropriately capture a lot of the information in the dataset without overfitting to the outliers and noise. Favorable initial weights allow for the NN trained on the outputs from KM to achieve a test accuracy of almost 52%. The models performance, however, is poor and degrades with more iterations which is likely due to the model's simplicity; with an input layer of size 2 and a single hidden layer of size 3, the model lacks enough complexity to generalize well. Thus, improved test accuracy may be realized by the addition of more hidden layers and units in each layer. While the GMM model is also simple –containing an input layer of size 2 and a single hidden layer of size 3 – it performs quite well. Test accuracy hovers around 52% throughout the learning process and approaches the target accuracy of 52.4% (that which is required to break-even on Vegas spread bets). The model also exhibits low variance and thus appears superior to all NN models. The strong performance likely indicates reasonable correlations between the GMM cluster probabilities and the target labels.

Figure 54 shows the test accuracy of a NN trained with varying number of ranked features on Dataset 1. Random search with 100 iterations was used to identify the best hyperparameters for each set of components and test accuracy is averaged over 10 random seeds. Using 4 features results in a local minimum in terms of out-of-sample accuracy, which likely occurs due to underfitting. With an input layer of size 4 and a single hidden layer of size 3, the network likely isn't complex enough to create appropriate decision boundaries capable of properly classifying the examples. Moreover, with only 4 features, more hidden layers or units in each layer would enable for more complexity and likely result in less biased generalization.

X. Conclusion

KM and GMM can both be effective methods to group similar observations on low dimensional datasets. However, neither of these algorithms scale well with increasing dimensionality. KM in particular suffers from the inductive bias of assigning equal weight to all features. Such a preference is less plausible as dimensionality increases. This bias also makes KM particularly sensitive to irrelevant features and noisy data. Dimensionality reduction algorithms can thus improve the performance of these clustering algorithms by removing irrelevant features, thus combating the clustering biases.

The choice of the appropriate clustering algorithm is problem dependent. With noise-free data, GMM may be preferred as the more sophisticated decision boundaries enable for more appropriate classifications. With noisy data, however, KM may be preferred as the complex decision boundaries of GMM result in the model overfitting to outliers. The spherical decision boundaries of KM are less susceptible to overfitting and are also easier to interpret. Thus, if interpretation is important or data is stochastic, KM is likely preferred.

Dimension reduction algorithms simplify the analysis of high-dimensional datasets and combat the curse of dimensionality. Further, using dimensionality reduction techniques to pre-process features prior to training a supervised learning model can whiten the noise in a dataset, thus preventing overfitting and improving generalization. Dimensionality reduction algorithms, however, have several limitations; in particular no dimensionality reduction algorithm can correct for non-predictive underlying features. Further, if features in a dataset are independent PCA and ICA will likely provide little benefit. RFE may be a useful strategy in such situations, but the technique requires an underlying supervised learning model and thus output labels must be known.

References

[1] Bengfort et al., (2019). Yellowbrick: Visualizing the Scikit-Learn Model Selection Process. Journal of Open Source Software, 4(35), 1075, https://doi.org/10.21105/joss.01075

[2] Crabtree, T. (2020). NFL Scores and Betting Data. Retrieved from https://www.kaggle.com/tobycrabtree/nfl-scores-and-betting-data

[3] Horowitz, M. (2018. Detailed NFL Play-by-Play Data 2009-2018. Retrieved from https://www.kaggle.com/maxhorowitz/nflplaybyplay2009to2016