STAT 639: Data Mining & Analysis **April 23, 2020**

Group 3 Final Project

Jordan Ankersen (827003348), FNU Rhythm (829002396), Phani Arvind Vadali (830000167), Namratha Bhat (827005643)

Supervised Learning

The supervised learning task involves classifying 400 observations (n) with 500 predictors(p) each into two classes. Since $p > n$, there might exist features that are not truly associated with the response and including such features when applying a classifier will lead to reduction in quality of fitted model and hence, an increased test error.[1] The current study proposes utilizing a variable selection method called Lasso regularization to pick the most influential variables before fitting a classifier on the dataset comprising of these variables.

Lasso regularization was done to find the λ such that the least estimated Misclassification Error (MSE) is obtained with 10-fold Cross Validation. The Reduced Dataset (RD) was constructed using the 16 variables whose coefficients did not go to zero when using the above λ_{min}. The performance of various classifiers was tested against the Full dataset (FD) i.e., all 500 features and 400 observations and the RD i.e., 16 features and 400 observations. Six classifiers were tuned and tested on the datasets.

Linear Discriminant Analysis (LDA): When the `lda` function in the `MASS` library was fit for FD the console output a warning that some variables may be collinear. LDA was then fit on RD which led to an estimated testing error of 26.34% in contrast to FD which had an estimated testing error of 41.54%.

Logistic Regression (LR): Logistic Regression was done using the `glm` function on the RD and FD. The z values converged to 0 when LR was fit on FD. The results are in Table 2.

Decision Trees: The decision tree was tuned using the `prune.misclass` function in the `tree` library. The pruning parameter was decided based on the generated graph between MSE and size of the tree. 10 fold cross validation was performed on the RD to get a testing

error of 27.06% and 25.97 % on the FD.

Support Vector Machines (SVM): SVM was tuned using the `tune.svm` function in the `e1071` library in R. Different kernels, gamma values (for radial and sigmoid kernels) and degree (for polynomial kernel) were tested. The cost was simultaneously varied from 0.001 to 100. The best parameters for each cost for each kernel were found and then compared.

Boosting: The `caret` library in R was used to tune parameters in Boosting. The parameters tuned were total number of trees (`n.trees`), the learning rate (`shrinkage`) and the number of new nodes made at a split point (`interaction.depth`). The tuned parameters for each of the two datasets are shown in Table 1.

Parameters	***RD***	***FD***
`n.trees`	450	460
`shrinkage`	0.001	0.01
`interaction.depth`	4	5
MSE	23.51%	28.68%

Table 1: Best parameters for Boosting

Random Forest: The only parameter which was tuned in Random Forest was the the number of variables sampled at each split i.e., the `mtry` values. This was done using 10-fold Cross Validation, and the optimal value of `mtry` for the least MSE is 7 for the RD and 50 for FD.

Summary: Table 2 provides a summary of the performance of all the classifiers. The $MSEs$ were estimated by doing 10-fold Cross Validation using the optimal parameters obtained after tuning. It can be seen that RD has lower $MSEs$ than FD which means that some of the variables are not as influential to the class than the others and only add to noise. Random Forest classifier when fit to the Reduced Dataset of only 16 of the initial 500 variables gives the least estimated MSE of 19.7% (i.e., an accuracy of 80%). Therfore, Random Forest should be used to classify the testing data.

Classifier	RD	FD
Linear Discriminant Analysis	23.64%	27.52%
Logistic Regression	26.04%	53.47%
Support Vector Machine	25.54%	35.37%
Decision Trees	25.97%	27.06%
Random Forest	**19.7%**	28.16%
Boosting	23.51%	28.68%

Table 2: MSRs of various classifiers after tuning parameters

Index	Description
Silhouette	Uses the difference between the average distance to the points in the same and closest cluster to create a measure of cohesion and separation of clusters. k^* maximizes the average silhouette width.[2] **K-means Results:** best $k^* = 2$ for $y90$ and $y95$
Gap Statistic	Compares the within-cluster dissimilarity to the expected dissimilarity under a null distribution. The smallest k resulting in a local maximum is optimal. **K-means Results:** $k^* = 12$ for $y90$, $k^* = 23$ for $y95$
NbClust	A package in R containing 30 indices for determining k^*. 26 of these indices were used. (Note: Silhouette method is one of the 26, so results reported out of 25). **K-means Results:** 6 of 25 indices selected $k^* = 2$ for $y90$ and $y95$

Table 3: Description of the indices used for selection of k^* and the results obtained for K-means clustering.

Unsupervised Learning

The goal of the unsupervised learning task was to determine the optimal number of clusters for a large, unlabeled dataset consisting of 1000 observations of 784 variables (let k denote a general number of clusters, k^* denote the optimal number). Various methods for determining k^* were performed for $k : [2 : 50]$. Clustering methods evaluated include K-means clustering, hierarchical clustering, d clustering and Gaussian mixture models (GMM).

Principal Component Analysis (PCA): Due the size of the dataset, PCA was conducted to reduce the dataset to a set of representative variables. The cumulative proportion of the variance explained was used to determine the number of principal components (PCs) used. To ensure that the number of PCs included was not influencing the results, two datasets, $y90$ and $y95$, that explained 90% and 95% of the variance in the data were created. The $y90$ dataset contained 187 PCs, and the $y95$ dataset contained 250 PCs.

Clustering Methods: Hierarchical clustering using complete linkage was conducted using the `hclust` function, K-means clustering was conducted using the `kmeans` function, and Clustering using GMM was conducted using the `Mclust` package. Finally, density based clustering was conducted using the `DBSCAN` package, and the value of ϵ was determined by identifying the elbow in a distribution plot of the 4 nearest neighbors. Upon observation of the results, it appeared that hierarchical clustering and the GMM were highly influenced by outliers. The results of density-based clustering were ambiguous for the clusters with similar density. For these reasos=n, we focus remaining discussions on results obtained from K-means clustering.

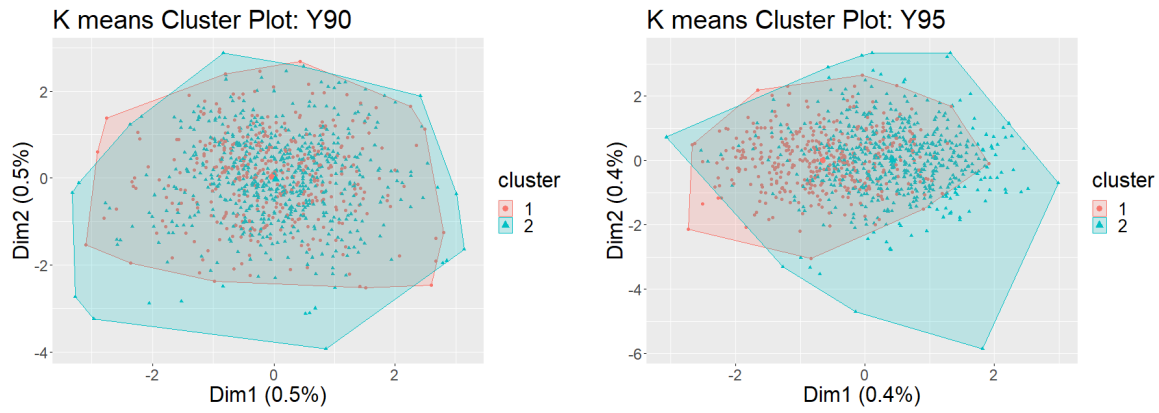

Figure 1: The optimal partition of the data into 2 proposed by NbClust. Both data sets have clusters with 395 and 605 members.

Determining k^*: Because the determination of an optimal k remains an unsolved problem, multiple methods for choosing k^* have been proposed. The methods that were used in this exercise are shown in Table 3 along with their results. AIC, BIC, and WSS were also considered, but a clear k^* was not identified by these indices[2]. The silhouette method and `NbClust` returned $k^* = 2$ for the $y90$ and $y95$ datasets, suggesting 2 as the optimal number of clusters. Additionally, `NbClust` determined the best partition of the data (Figure 1).

Conclusions: Based on the results presented, we propose that best partition of the data can be found using $k^* = 2$ and K-means clustering. When observing Figure 1, however, it is clear that the clusters overlap substantially. This is likely due to the fact that each principal component accounts for less than 0.5% of the variance in the data. For this reason, viewing the data in terms of only 2 principal components may not allow visualization of differences existing between the clusters. Moreover, this indicates that reduction of the size of the dataset using linear PCA may not be the best method. An alternative to PCA may include a non-linear method such as t-distributed stochastic neighbor embedding (t-SNE).

References

[1] Gareth James, Daniela Watson, T. H. R. T. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer, New York.

[2] Trevor Hastie, Robert Tibshirani, J. F. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2 edition.