Final Project Report

STAT 639-700

Samuel Burge (UIN: 331008088)

Chad Austgen (UIN: 727007397)

Skylar Liu (UIN: 727007397)

April 19, 2022

Classification Task

For the classification task we used the Naive Bayes classifier, regularized logistic regression (specifically ridge, lasso and elastic net regularization), gradient boosted trees, random forests and support vector machines with radial and polynomial kernels. For model selection and assessment, we utilized 10-fold cross-validation and, when needed, another nested cross-validation to select tuning hyper-parameter using a grid search. Nested cross-validation is a well-known approach to handling both tuning and assessment of models with hyper-parameters, and we opted to employ this approach to avoid potential over-fitting and bias introduced when the same data used to validate the models are also used in parameter tuning (Cawley and Talbot 2010).

Before the analysis we created plot matrices and verified across all predictors that X and Y were not well separated, suggesting logistic regression might be a good model. However, we were not able to fit a logistic regression model outright since $p > n$. We utilized regularization methods to reduce the features in our analysis using lasso, ridge, and elastic net regularization methods. We also opted to use a different boosting algorithm for the classification trees,

XGBoost, compared to the algorithms found in R's gbm package. This was primarily due to the computational resources necessary to perform nested cross-validations, as XGBoost is known for it's impressive predictive performance and computational efficiency (Chen and Guestrin 2016).

The results of our analysis are depicted in the table below. Overall, the tree-based methods had the best performance of all the classifiers and the boosted classification trees had the best generalization performance with a CV error rate of 31%. After re-fitting the final model using the same grid search procedure, our model's tuning parameters were an 80% subsample ratio the training instances, a learning rate (shrinkage penalty) of 0.001 ,and a max interaction depth for each tree of 4. These tuning parameters were selected primarily to help us avoid over-fitting the model given the large amount of flexibility these algorithms can provide if left unchecked, although the interaction depth was deeper than we expected, which could suggest that there are some interactions between the variables in the data.

Table 1: Results from 10-fold cross-validation to assess model generalization performance.

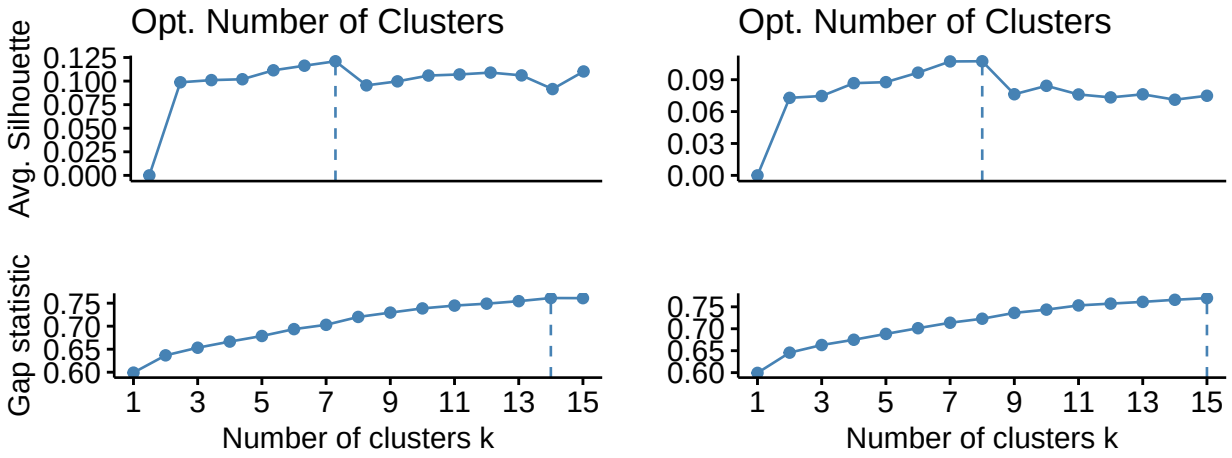
	Training Error	Est. Test Error
Lasso	0.3060	0.3727
Net	0.3060	0.3815
Ridge	0.1786	0.3972
Naive Bayes	1.0000	0.6020
Radial SVM	0.4468	0.4460
Poly. SVM	0.4147	0.4384
Random Forest	NA	0.3125
Boosted Trees	0.0402	0.3082

Clustering Task

For the clustering task, we initially looked at performing principal components analysis (PCA) to reduce the number of dimensions in the data set. The two scree plots below show

that the number of principal components necessary to capture at least 90% of the variation in the data set was about 187, which did not seem beneficial enough to consider for the analysis. Therefore, we decided to retain all the original features in the data set.

We opted to use k-means and hierarchical clustering, in part because the high-dimensional data is intractable with DBSCAN and due to computing restraints. Since the given data does not have any contextual basis for selecting the number of clusters K, we used several widely used approaches include the elbow method, the silhouette method (Rousseeuw 1987), and more recently the use of the gap statistic (Tibshirani, Walther, and Hastie 2001). We decided on $K = 7$.



The graphs suggested possibly using 14, but we ultimately decided on 7 which is easier to interpret. The results of the K-means and hierarchical clustering methods are depicted in the bar graphs below. K-Means grouped the data into relatively equal sized clusters, with cluster 5 having slightly more observations. Hierarchical clustering resulted in the majority of our data falling into cluster 2, which was consistent with the dendrogram for complete linkage hierarchical cluster. Because of the difficulty visualzing data in high dimensions in

the data, we plotted bar graphs of the clusters from both methods.

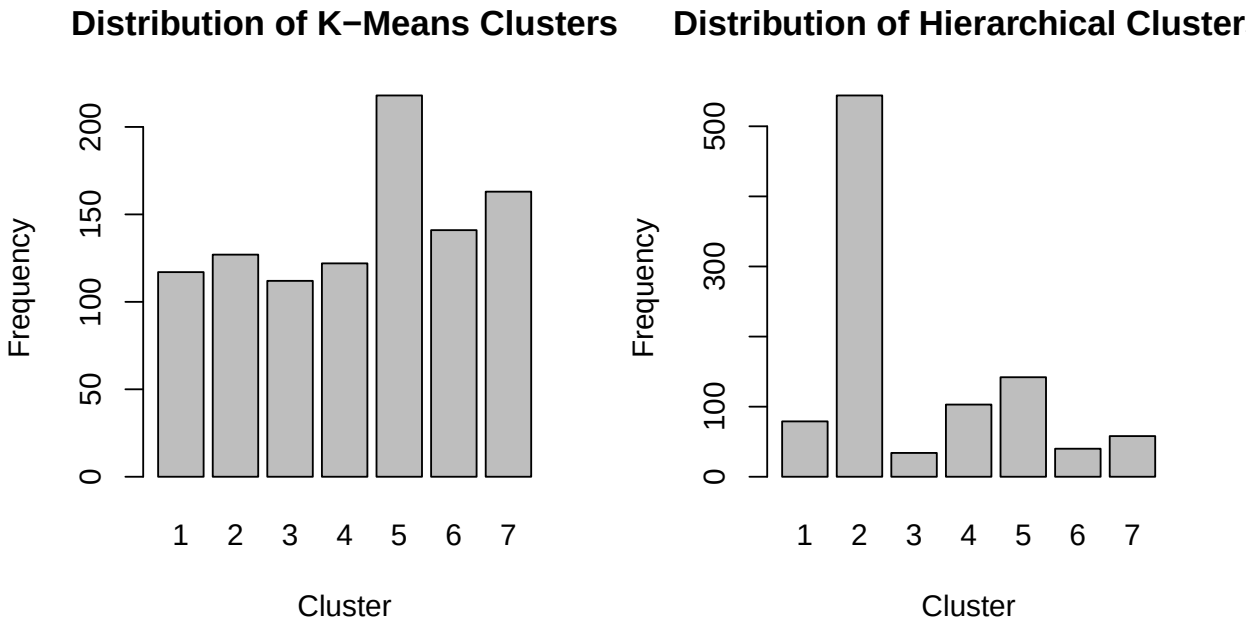


Figure 1: Count of observations in each cluster for both clustering methods.

Cawley, Gavin C., and Nicola L. C. Talbot. 2010. "On over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation." *Journal of Machine Learning Research* 11 (70): 2079–2107. http://jmlr.org/papers/v11/cawley10a.html.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. KDD '16. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785.

Rousseeuw, Peter J. 1987. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics* Volume 20 (https://doi.org/10.1016/0377-0427(87)90125-7): 53–65.

Tibshirani, Robert, Guenther Walther, and Trevor Hastie. 2001. "Estimating the Number of Clusters in a Data Set via the Gap Statistic." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63 (2): 411–23. https://doi.org/https://doi.org/10.1111/1467-9868.00293.