STAT 639 Project: Group 11

Vasu Verma (933004913) | Sumeet Nazare (131005188) | Raj Bhavsar (933005059)

Supervised Learning

This project involves a classification problem using a dataset with three variables: x, y, and xnew. We are tasked with training various classifiers to this dataset and estimating the testing error. The goal is to find the best classifier with the smallest testing error.

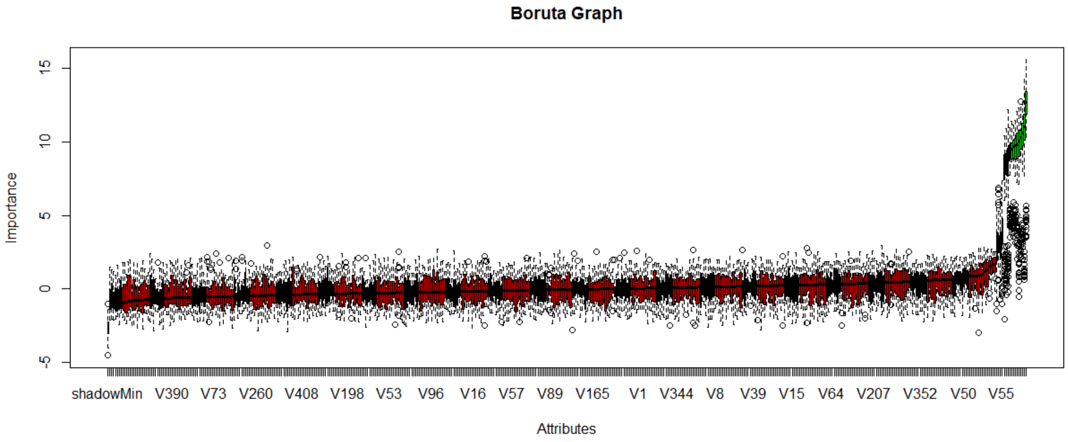
We used several classifiers, including some not covered in the course, to train the dataset. We explained the methods used to tune the parameters of each classifier and estimate the testing error. Readable R code was provided, and we made sure to explain any complex code in plain English. The methods used for classification have been mentioned with the test error estimates in the table below. 'Boruta + XgBoost' (Our Best Method) has been explained briefly.

Methods	Test Error Estimate
Logistic Regression	0.500
LDA	0.475
SVM	0.525
Neural Net	0.4375
PCA + Random Forest	0.375
Boruta + XgBoost	0.225

Boruta + XgBoost (Our Best Method): Boruta is a feature selection technique that is used to identify the most important input variables for a classification problem. It works by comparing the importance of each variable to the importance of random shadow variables. In this method, Boruta is used to select the most important input variables before training an Extreme Gradient Boosting (XgBoost) classifier. XgBoost is a tree-based ensemble method that is similar to Random Forest but uses a different optimization algorithm.

Steps Followed:

1. Splitting the dataset into two parts, keeping 80% for training and 20% for estimating test error
2. Applying Boruta Feature Selection to reduce the number of predictors from 500 to 21
3. Using 5-fold Cross-validation, finding the best parameters for the XgBoost Model that gives minimum logloss
4. Training the final model using the best parameters on the whole training dataset
5. Estimating the test error on the separate 20% dataset that we kept aside



Best Parameters Used:

eta: 0.06975552 ; gamma: 0.1671435; subsample: 0.6994722; max_depth: 8

We were able to achieve a test error estimate of 22.5% using Boruta + XgBoost, which was better than all the other models we tested. Boruta was able to provide us with a huge dimensionality reduction. Overall, the methods used in the project include linear models such as Logistic Regression and LDA, ensemble methods such as Random Forest and XgBoost, as well as dimensionality reduction techniques such as PCA and feature selection techniques such as Boruta.

Unsupervised Learning

The unsupervised problem focuses on obtaining the optimal number of clusters for a given high-dimensional dataset that has 1000 rows and 784 columns. The objective of the project is to explore different methods to obtain the optimal number of clusters.

1. Dimensionality Reduction

- Non-zero variance was calculated but no values were eligible to be dropped
- Using PCA, 300+ components could explain ~90% variance of the data
- T-SNE and UMAP were used to reduce the components to two dimensions

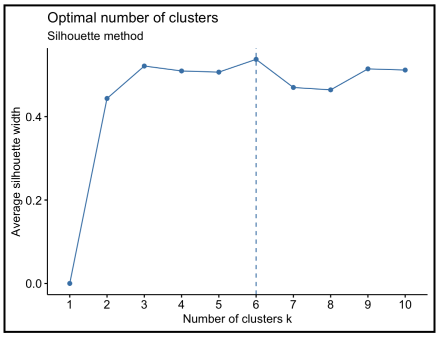
2. Assessing Cluster Tendency

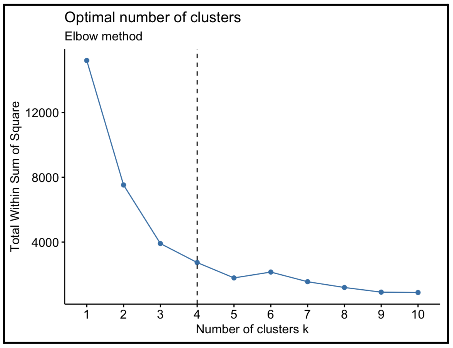
- Hopkins Statistics and Visual Methods were used to assess cluster tendency. For both, UMAP and T-SNE Hopkins had a score > 0.5 but the visual methods displayed clusters

3. Find the Optimal Number of Clusters

- KMeans, PAM, CLARA, and Hierarchical Clustering was executed using the Elbow and Silhouette method. These methods were executed using the fviz_nbcluster() function. The Gap Statistic was another method explored for a few methods which also displayed similar results. But, it was computationally intensive to run all the algorithms
- For DBSCAN, different eps, and min_pts were explored to identify the optimal values
- Model-Based Clustering used a Gaussian finite mixture model fitted by the EM algorithm which focused on selecting the best model which has the highest BIC.
- Fuzzy and Hierarchical K-Means clustering algorithms used Average Silhouette Width to identify the optimal clusters. The methods used clusters from sizes 4 to 9 to identify the Average Silhouette Width and the best results can be observed in the tables below.

Algorithm	**Cluster Selection Method**	**T-SNE Clusters**	**UMAP Clusters**
K-Means	Elbow	4	4
	Silhouette	5	6
PAM	Elbow	5	4
	Silhouette	4	6
Clara	Elbow	4	4
	Silhouette	6	6
Hierarchical Clustering	Elbow	4	4
	Silhouette	5	6





Algorithm	**T-SNE Clusters**	**Average Silhouette Width**	**UMAP Clusters**	**Average Silhouette Width**
DBSCAN	5	-	5	-
Model-based Clustering	9	-	9	-
Fuzzy	5	0.53	5	0.47
Hierarchical K-Means	5	0.53	5	0.49

References

1. “Xgboost in R: How Does xgb.cv Pass the Optimal Parameters Into xgb.train.” *Stack Overflow*, 28 Jan. 2016, stackoverflow.com/questions/35050846/xgboost-in-r-how-does-xgb-cv-pass-the-optimal-parameters-into-xgb-train.
2. Practical Guide to Cluster Analysis in R by Alboukadel Kassambara
3. Feature Selection in R with the Boruta R Package https://www.datacamp.com/tutorial/feature-selection-R-boruta