Joshua Kim (432001712); Jack Rodoni (531006635); Christian Santosa (227008504)

Supervised

We started by fitting classification models to the complete set of 500 features. Since our data set has more features (p = 500) than observations (n = 400), this ruled out using classification methods logistic regression, linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA), at least for the initial round of fitting classifiers.

The classifiers we ended up fitting, using the whole set of available features we're: Lasso Regression, Ridge Regression, Elastic Net Regression, Naive Bayes, Support Vector Machine (SVM) with a linear kernel, SVM with a radial kernel, SVM with a polynomial kernel and Random Forest (RF). For each of the above classification models, we split the data into training and test sets. The hyperparameters for the model were then tuned by selecting the set of hyperparameters for which the test error estimated using repeated cross-validation with k=10 folds and 5 repeats was lowest. A final estimate for test error was then reported by using the test set to make predictions.

The smallest estimated test error we obtained from these classifiers came from the random forest model with mtry = 22, with an estimated test error of 0.3535

None of the classifiers we attempted gave us what we considered to be an acceptable classification error rate. We decided, since we had a large number of features in the original data set, it might be a good idea to do feature selection before fitting our models in order to reduce the number of features in our classification models.

The method we ended up using for feature selection was Boruta. Boruta is a feature selection algorithm that makes use of the feature importance metric from random forest (Mean Decrease Accuracy). Essentially, the algorithm makes a copy of your features, then randomizes the observations within each feature to decorrelate the copies with the y-variable. It then appends these copies of the features to your dataset and fits a random forest model to the new data set. It then obtains importance metrics for each of the features (your original features and the randomized copies, called "shadow" features). Then, if we have statistically significant evidence that the importance of one of our original features is less than the importance of the most important shadow feature, the original feature and its shadow copy are removed from the dataset. This process is repeated many times to give us a final set of features.

In conjunction with this feature selection method, we ran SVM models with Linear, Radial and Polynomial kernels, as well as ridge and elastic net models. When using Boruta for feature selection, we used nested-CV to estimate the test error, where the inner CV was used to select features and tune hyperparameters and the outer CV was used to estimate the test error. Of the models we ran the one that gave us the best results was SVM with Radial kernel. The final set of features used in this model was: v2, v11, v50, v77, v164, v218, v222, v228, v230, v341, v350 and v409. The hyperparameters were C=8, sigma = 0.08114262. With this model, we obtained an estimated test error of 9.7561%

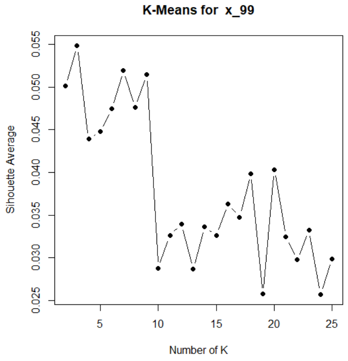
Unsupervised

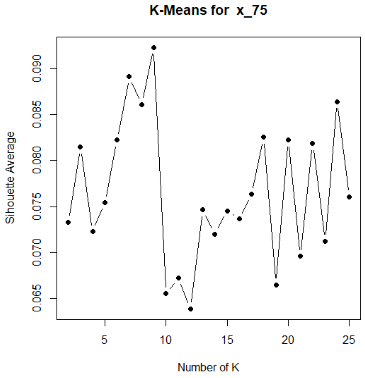
1. Apply PCA to reduce the dimensionality (pre-processing)

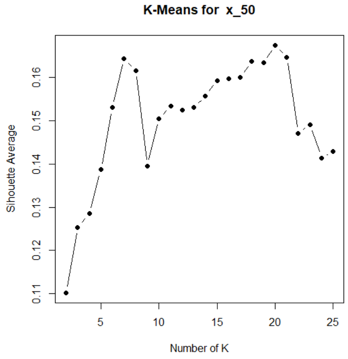
We will be using data sets with 2, 7, 22, 85, 382 principal components, which consist of 14, 33, 50, 75, 99% variation retained. We will examine how the optimal values of k change as the variance proportion changes

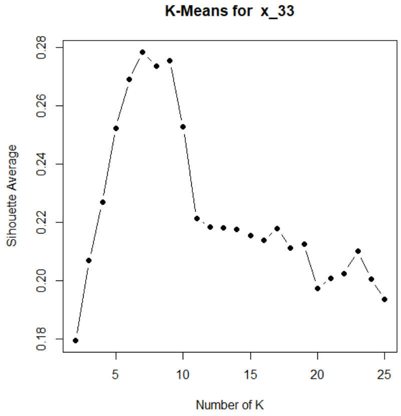
2.1. K-means

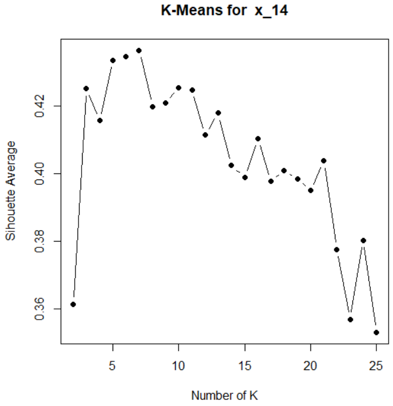
We applied k-means to each data set and used silhouette average to determine k.





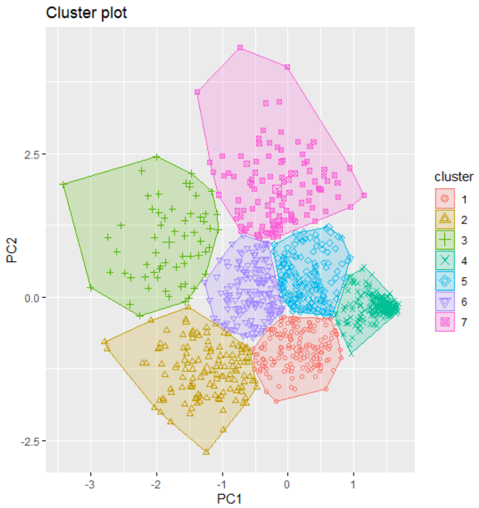






We see that the optimal k differs with respect to the variation proportion used. However, there seems to be a consistent peak around k=7 for all cases. The first two principal components allow us to cluster our data into 7 groups, and the relatively high average silhouette score persists although the “optimal” for each data may change. Thus, it is possible that the data intrinsically has partition of k~7 that may not be fully revealed for a full data set but becomes more clear as we take away the smaller variance, assuming them to be noise. The following is how k=7 looks based on 2 PC’s.

2.2 K-Medoids/PAM



Since k-means assumes that our clusters form a sphere., we also applied K-Medoids/PAM, which loosens up such assumption. We observe a similar but less consistent graphs with silhouette average peaking between k=5~10. Although our result isn’t as stable, it seems to be supporting k being 7 more so than it is undermining it.

2.4. Failed Methods

Hierarchical Clustering and DBSCAN failed to produce any meaningful results.

References:

Boruta:

https://www.machinelearningplus.com/machine-learning/feature-selection/

https://www.listendata.com/2017/05/feature-selection-boruta-package.html

Nested-CV:

https://i.stack.imgur.com/vh1sZ.png

Elastic Net/ Ridge/ Lasso:

http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/153-penalized-regression-essentials-ridge-lasso-elastic-net/#elastic-net

Caret package (R):

https://topepo.github.io/caret/index.html

Clustering:

https://afit-r.github.io/kmeans_clustering