Stat 639 Project

Joshua Kim, Jack Rodoni, Christian Santosa

April 26, 2022

Section 1

1. Supervised Learning
    - Approach
    - Results
    - Summary

2. Unsupervised Learning
    - Approach
    - Results

Supervised Learning: Notes

- We have more features (p = 500) than observations (n = 400) in our training dataset. This leaves us with two options:
  a. Only utilize models that don't require n>p (e.g. we can't use classifiers such as logistic regression, LDA, QDA,... etc)
  b. Perform feature selection to reduce the number of features such that n>p
- We decided to explore both options.

Supervised Learning

Using Full Training Data Set (No Feature Selection)

- We fit a number of different classification models
  - Naive Bayes
  - Lasso Regression
  - Ridge Regression
  - Elastic Net Regression
  - SVM with Linear Kernel
  - SVM with Polynomial Kernel
  - SVM with Radial Kernel
  - Random Forest

Supervised Learning

Using Full Training Data Set (No Feature Selection)

For each of the classification models we fit, we tuned the hyperparameters and estimated test error in the following way*:

1. We split the data into training and test sets (75:25)
2. The hyperparameters were then tuned on the training set using repeated-CV with k=10 folds and 5 repeats
3. We then obtained a final estimate of the test error using the 25% of the data not included in the test set

We employed this method because we found that the estimated test error from CV was consistently lower than our estimated test error from the holdout set. Thus, the holdout set in our scheme acted as a salinity check to make sure we weren't underestimating our test error.

* OOB Error Rate was used as the estimate for test error with random forest

Supervised Learning

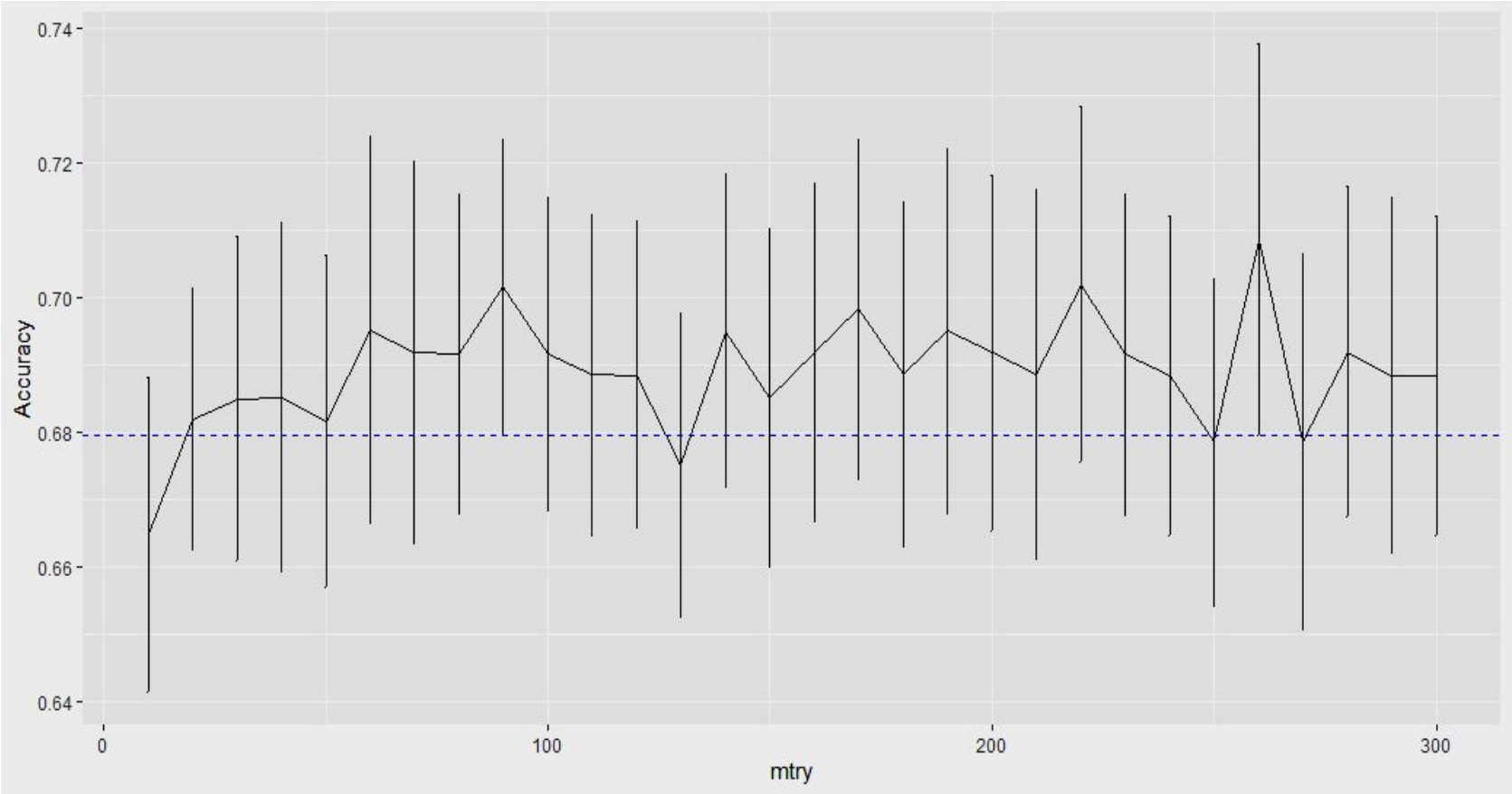
Using Full Training Data Set (No Feature Selection)

Classifier	Estimated Misclassification Rate
Naive Bayes	0.4141
Lasso Regression	0.4141
Ridge Regression	0.4040
Elastic Net Regression	0.3737
SVM (Linear Kernel)	0.4848
SVM (Polynomial Kernel)	0.4444
SVM (Radial Kernel)	0.4040
Random Forest	**0.3275**

Hence, when using the full dataset, Random Forest was clearly the best classifier.

Supervised Learning

(Tuning mtry for Random Forest)



Supervised learning

(Using Boruta Algorithm to Perform Feature Selection)

In order to perform feature selection we used the Boruta algorithm.

The Boruta Algorithm works in the following way:

- Creates “shadow” features that are copies of your original features, but randomized
- Appends the shadow features to the dataset
- Fits a RF model to the dataset to obtain variable importance measures (the default in R is Mean Decrease Accuracy)
- If we have statistically significant evidence that the importance of one of our features is less than the importance of the most important shadow feature, the feature is removed from the dataset.
- Repeat this process many times (100 is the default in R)

In conjunction with using Boruta for feature selection we fit the following models:

- SVM with Linear, Radial and Polynomial Kernels
- Ridge Regression
- Elastic Net Regression

Supervised learning

(Using Boruta Algorithm to Perform Feature Selection)

For each of the classification models we fit, we performed feature selection, tuned the hyperparameters and estimated test error using nested-CV

1. Split the data into k-folds (k = 10)
2. Perform feature selection on the k-1 (9) folds
3. Split the k-1 (9) folds into k*-folds (k*=10)
4. Tune hyperparameters on the k* folds
5. Estimate test error using the original fold not used in feature selection

Supervised learning

(Using Boruta Algorithm to Perform Feature Selection)

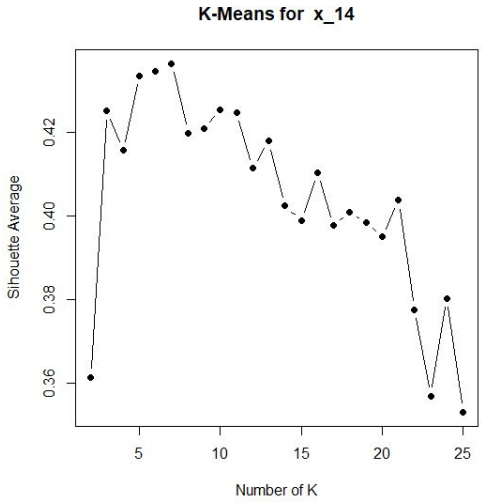
Classifier	Estimated Test Error
SVM (Linear)	0.21875
SVM (Polynomial)	0.1219512
SVM (Radial)	**0.097561**
Ridge Regression	0.2307692
Elastic Net	0.2790698

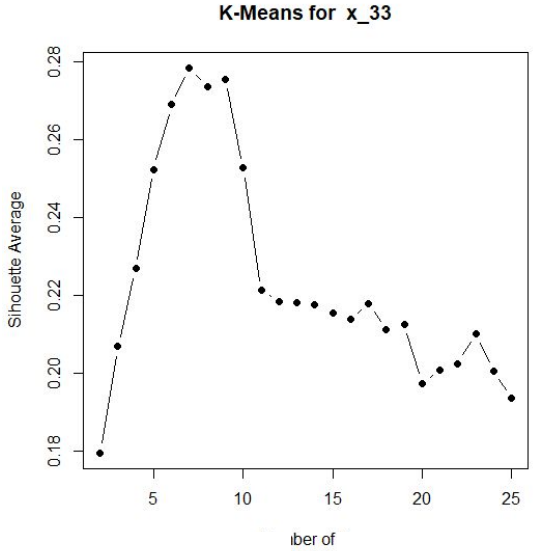
Features Selected: v2, v11, v50, v77, v164, v218, v222, v228, v230, v341, v350, v409
HyperParameters: C = 8, Sigma = 0.08114262

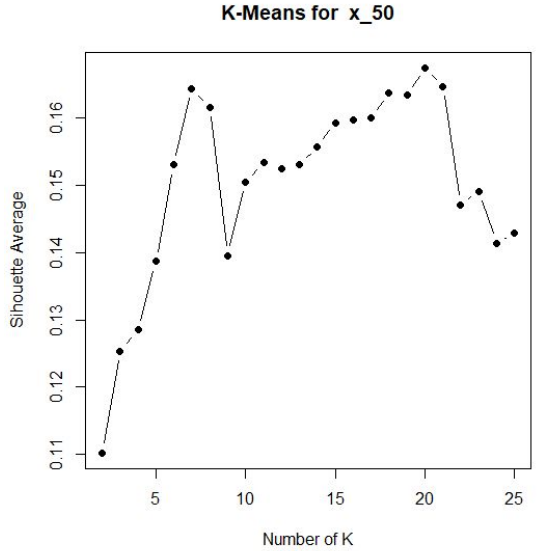
Unsupervised Learning

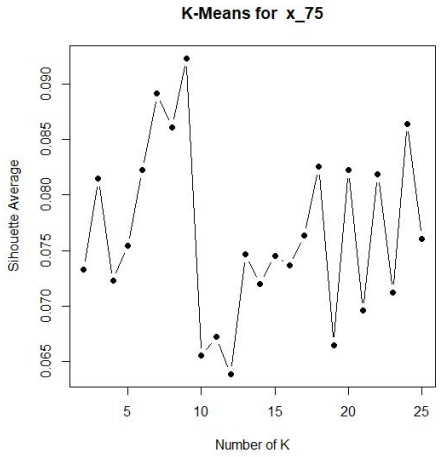
1. Pre-processing (PCA)
    - Dimensionality reduction
    - Test multiple sets with varying number of principal components
        - 14%, 33%, 50%, 75%, 99% variation proportion used
2. Clustering Methods
    - K-means
    - K-medoids/PAM
    - Hierarchical clustering
    - DBSCAN
3. Performance metric
    - Silhouette average

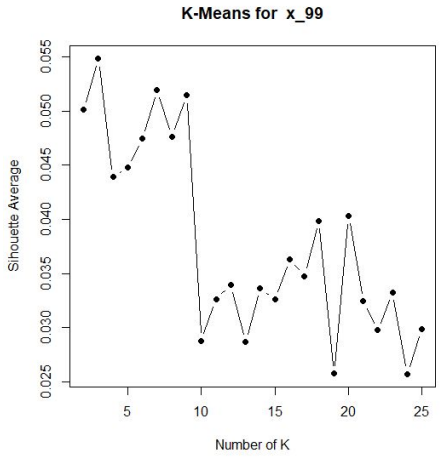
K-Means


K-Means for x_14
Silhouette Average
0.36
0.38
0.40
0.42
5
10
15
20
25
Number of K

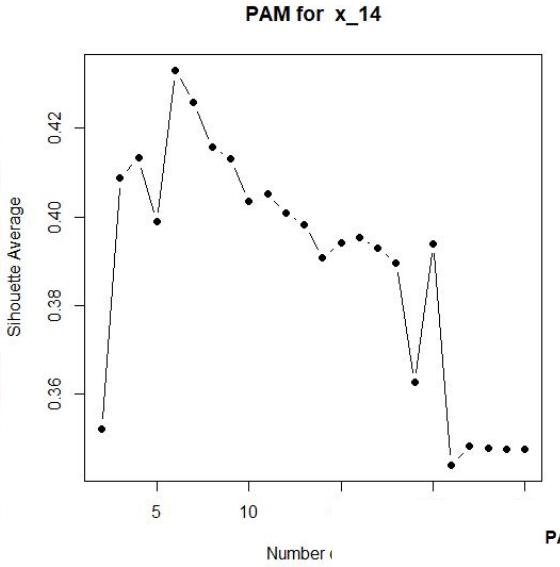

K-Means for x_33
Silhouette Average
0.18
0.20
0.22
0.24
0.26
0.28
5
10
15
20
25
ıber of

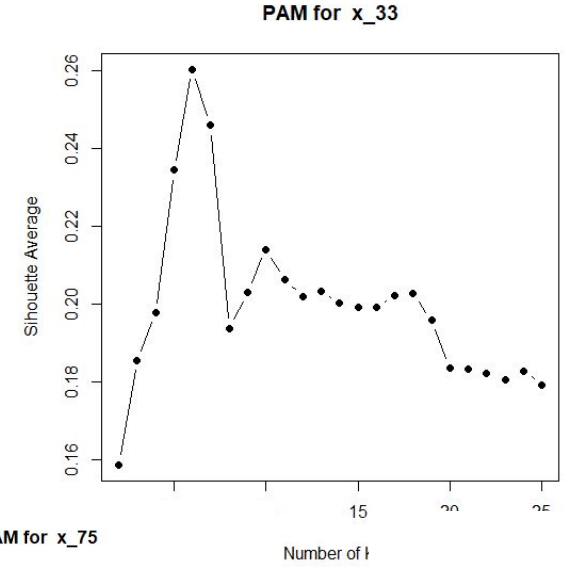

K-Means for x_50
Silhouette Average
0.11
0.12
0.13
0.14
0.15
0.16
5
10
15
20
25
Number of K

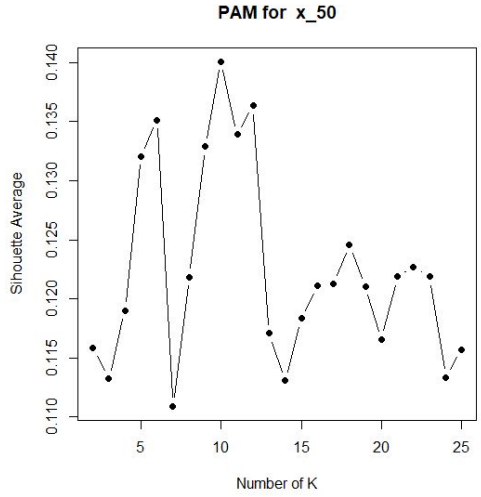

K-Means for x_75
Silhouette Average
0.065
0.070
0.075
0.080
0.085
0.090
5
10
15
20
25
Number of K

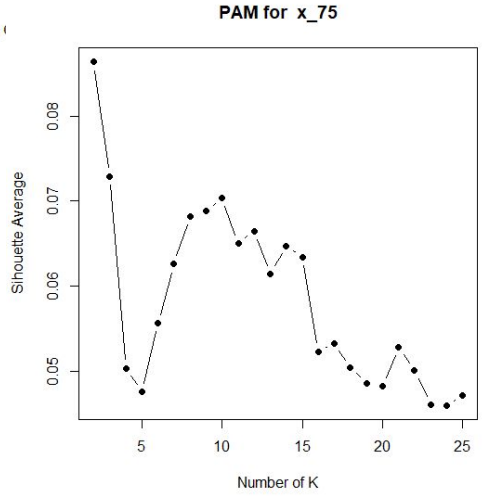

K-Means for x_99
Silhouette Average
0.025
0.030
0.035
0.040
0.045
0.050
0.055
5
10
15
20
25
Number of K

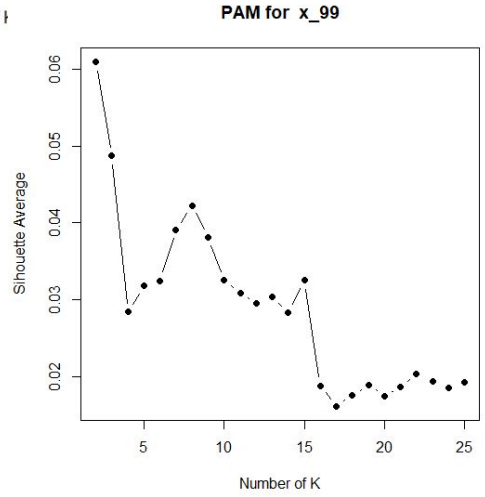
K-Medoids/PAM











K=7