CS/STAT 6301.0U2 – R for Data Scientists

Midterm Project (*Clustering and Regression*)

MEMBERS:

Abhinav Kumar Labhishetty	axl160730
Himadhar Duvuru	hxd161130
Teja Krishna Talluri	tkt160230

PART-1

Purpose: Part(i) of this project studied the application of K-Means and H-Clustering

Dataset(s):

For part(i) the dataset can be found at: https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping .

Data Set Description: This data set contains training and testing data from a remote sensing study which mapped different forest types based on their spectral characteristics at visible-to-near infrared wavelengths, using ASTER satellite imagery. The output (forest type map) can be used to identify and/or quantify the ecosystem services (e.g. carbon storage, erosion protection)

Goal:

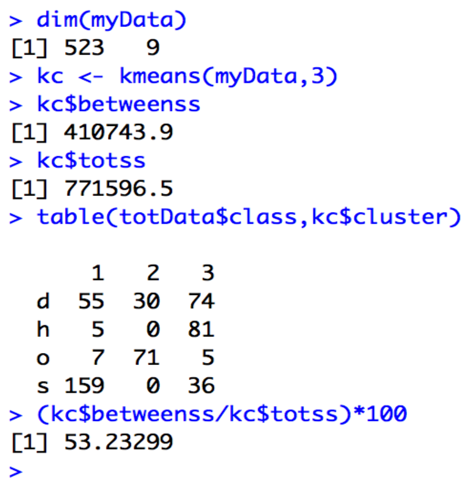
To do a cluster analysis on this dataset, and then see how closely your clusters align with the forest types.

Data Cleaning:

1. We have two data sets Test Data and Training Data each of these consists of 28 columns in each.
2. Combining both the Data sets we get 523 rows and 28 variables.
3. As per the given Data only the first 9 variables (b1….b9) are used to determine the type of forest, So getting rid of all the other unnecessary variables we are left with a data set of 523 records, 10 variable (first variable is the Forest Class variable)

Approach-1:

1. First, we applied k-means clustering on the given data with 3 and plotted the resulting cluster with the Class variable in our data set to validate how accurate is the clustering.



```
> dim(myData)
[1] 523   9
> kc <- kmeans(myData,3)
> kc$betweenss
[1] 410743.9
> kc$totss
[1] 771596.5
> table(totData$class,kc$cluster)

      1   2   3
  d  55  30  74
  h   5   0  81
  o   7  71   5
  s 159   0  36
> (kc$betweenss/kc$totss)*100
[1] 53.23299
>
```

2. As we can see above it only gave an accuracy of 53.24% which is very poor clustering and we cannot afford up to 50% sum of square in the clustering.

3. K-Mean analysis for 4 clusters as initial point gives an improved result but not up to the mark. Below shows the accuracy for the k-mean 4 clustering (63%).

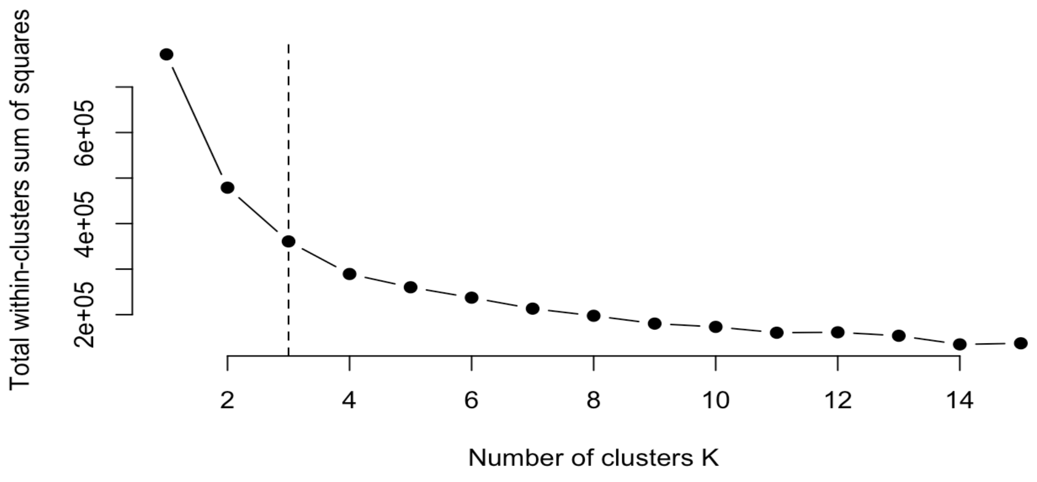
```
> kc <- kmeans(myData,4)
> kc$betweenss
[1] 482287.7
> kc$totss
[1] 771596.5
> table(totData$class,kc$cluster)

      1   2   3   4
  d  44   6   2 107
  h   5   0  81   0
  o   5  54   1  23
  s 161   0  32   2
> (kc$betweenss/kc$totss)*100
[1] 62.50517
```

4. To determine which order of clustering provides the better accuracy for the given data we have plotted the BetweeSS-TotalSS ratio for all clusters ranging from (0-15).
   The code for graph is

```
k.max <- 15
wss <- sapply(1:k.max,
              function(k){kmeans(myData, k, nstart=10 )$tot.withinss})
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
abline(v = 3, lty =2)
```

5. The result of the E-Blow graph is as below. It clearly shows that the clustering should be done between 3 or 4. (yet not completely accurate).
6. The plot between no: of clusters and Total with in cluster sum of squares is:



Approach-2:

Next, we have done h-clustering to the dataset, we used Euclidian distances amongst the variables to plot a h-cluster Dendrogram. We analyzed the Dendrogram behavior for four different kinds of Linkages as shown below (Complete, Average, Single, Ward D2).

In case of Complete and Average, the dendrogram shows outliers which is not good for clustering the data points. When cutree is performed these two plots for classifying the data points in to clusters most of the data points fall under one single observation(classification). Which is mainly because of the outliers.

In case of the Single linkage the plot is completely skewed in on direction making it even worse for clustering problems.

However, The Euclidian distances plotted in WardD2 linkage results in more accurate dendrogram along with the cutree dividing the data points more reasonably that the other Linkages. The table shown below gives out a clear cut differentiation for Forest types D, H, S using this linkage yet some part of the O type forests seems to be merging with the D type forest.

Analysis and Observation:

i. The data plotted is the observation of 3 different characteristics of these forests for three days which gives repetition of observation.
ii. It would be better to analyze each of these parameters individually and determine how much they influence the clustering.
iii. The Data set we have is just for 3 days which is very little observation to come to a conclusion for classifying the Forests in to 4 types. Lot more observations can bring a more accurate result for both h-clustering and k-means as well.
iv. The dendrogram clearly shows that this data is not completely sufficient to cluster these forest data in to 4 types.

The above listed points could also be a reason for the k-means (between SS- total SS) ratio to be very low. An improved data set should be able to get us a (between SS- total SS) ratio somewhere between 75- 80 %.

The Elbow graph clearly shows the data points can be classified in to 3 categories very finely whereas have a 4 Forest Types.

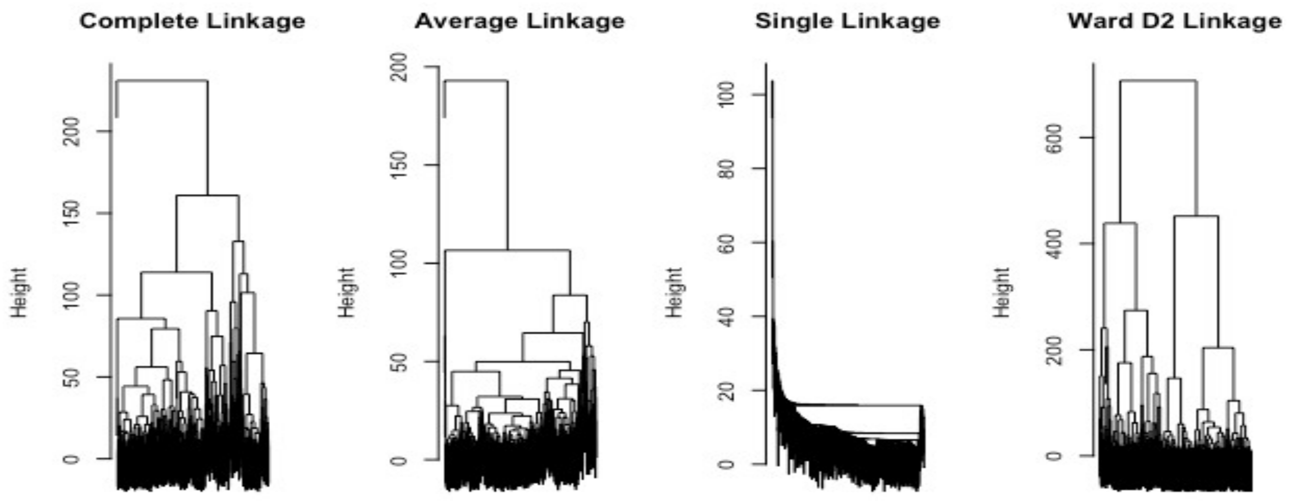
Conclusion:

So we can conclude that with the given data it is not possible to classify the give forest data in to 4 kinds of forests. This could be for multiple reasons mentioned above or there is a very little difference between type D and type O forests resulting these data points to group up (overlap) in both the methods analyzed above. These forest types could be analyzed for few more parameters apart from the 3 observed in the give data set to distinguish these two forests types more accurately.

Summary:

- With large number of variables k-means clustering is computationally faster H-clustering, provided k is small.

- Values of k will affect the outcome
- K means will start with random center of cluster each time we run the model, it will result in different outputs lacking consistency.
- H-clustering does not need number of outputs to be specified.



```
> table(cutree(myData.complete,4), as.factor(testData$class))

      d   h   o   s
  1  24  61  12   4
  2 134  25  38 191
  3   1   0  32   0
  4   0   0   1   0
> table(cutree(myData.average,4), as.factor(testData$class))

      d   h   o   s
  1 155  86  40 195
  2   4   0  39   0
  3   0   0   3   0
  4   0   0   1   0
> table(cutree(myData.single,4), as.factor(testData$class))

      d   h   o   s
  1 159  86  80 195
  2   0   0   1   0
  3   0   0   1   0
  4   0   0   1   0
> table(cutree(myData.wardD2,4), as.factor(testData$class))

      d   h   o   s
  1   6   0  39   0
  2   2  81   0  29
  3  23   5   5 164
  4 128   0  39   2
```

PART-2

Purpose: Use Ridge, Lasso, and PCR to build models and CV to test them.

Dataset(s):

For part(ii)
The dataset can be found at: https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized .

Data Set Description: Communities in the US. Data combines socio-economic data from the '90 Census, law enforcement data from the 1990 Law Enforcement Management and Admin Stats survey, and crime data from the 1995 FBI UCR.

Goal:

To build a model that will predict the per capita crime rate in terms of the other variables.

Given:

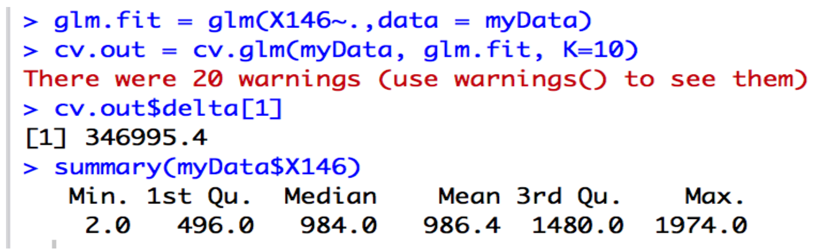
The per capita violent crimes variable was calculated using population and the sum of crime variables considered violent crimes in the United States: murder, rape, robbery, and assault. There was apparently some controversy in some states concerning the counting of rapes. These resulted in missing values for rape, which resulted in missing values for per capita violent crime. Many of these omitted communities were from the mid-western USA (Minnesota, Illinois, and Michigan have many of these).

Data Cleaning:

1. The given data has 147 columns, with 125 predictive, 4 non-predictive, 18 potential goal.
2. Per capita Crime rate is one of these 18 potential goals so we will have to get ride of all the unnecessary 17 potential goals.
3. Most of the Data is Character based which is unclassed and converted to a data frame.
4. Analyzing the data frame, we find about 1872 NULL values in main columns ranging from X104 -X146 which needs to be dropped.
5. After losing all the NA values and converting data set to numeric we will be left with a data frame of dimension 1993 rows and 104 columns.

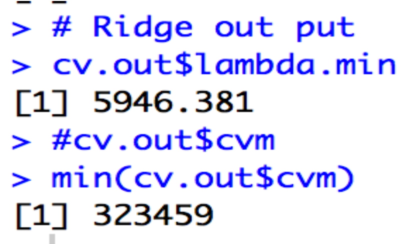
Approach:

1. First, we will build a simple linear regression model and cross validated it with 10-fold CV. We are using k-fold Cross validation to measure the accuracy of the model. Since it is a known fact that k-fold CV provides the most accurate method to validate the model. The mean square error for simple linear regression is obtained as 346995.40(variable).



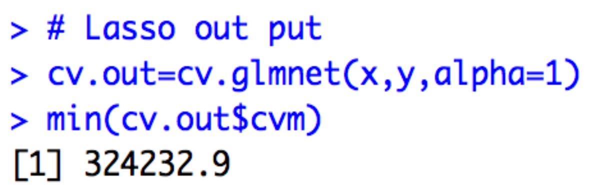
```
> glm.fit = glm(X146~.,data = myData)
> cv.out = cv.glm(myData, glm.fit, K=10)
There were 20 warnings (use warnings() to see them)
> cv.out$delta[1]
[1] 346995.4
> summary(myData$X146)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    2.0   496.0   984.0   986.4  1480.0  1974.0 
```

2. We built a model matrix to perform Ridge and Lasso for the give data. When we applied Ridge on the model the MSE is obtained as 323459. Note we use the built in CV function to select the best lambda, and this will also give the test MSE error estimate for the lambda. We tested a wide range of lambdas to find the best one. Which is given by the lambd.min



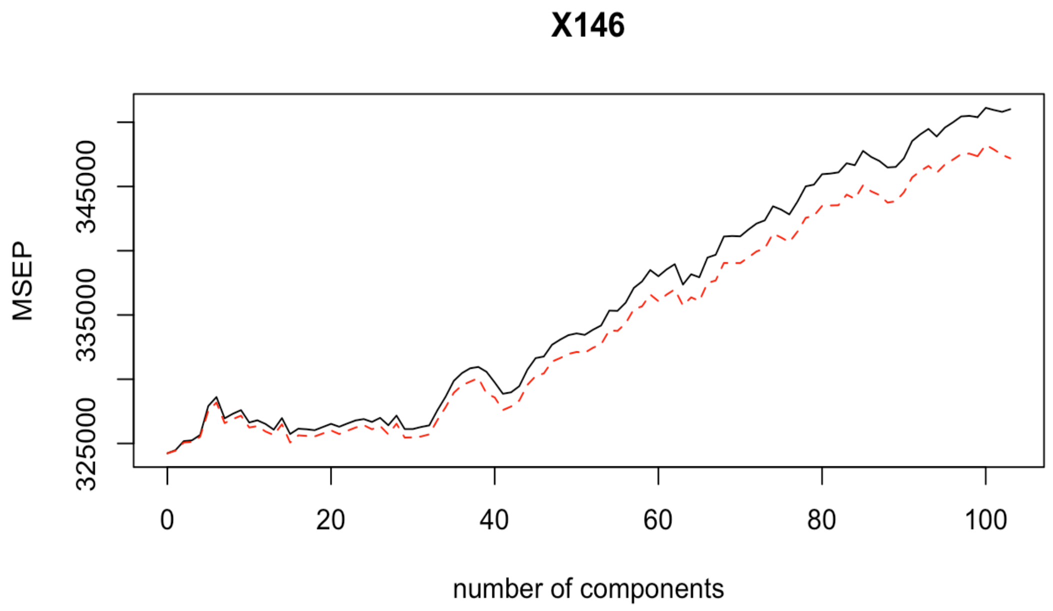
```
> # Ridge out put
> cv.out$lambda.min
[1] 5946.381
> #cv.out$cvm
> min(cv.out$cvm)
[1] 323459
```

3. When we applied Lasso on the model the MSE is obtained as 324232.9.

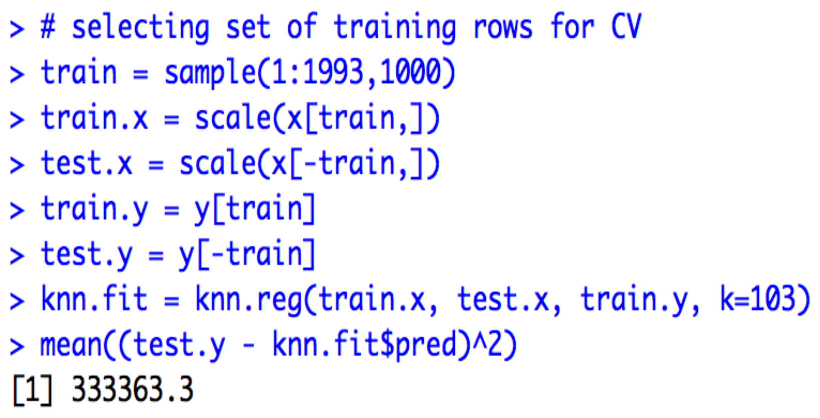


```
> # Lasso out put
> cv.out=cv.glmnet(x,y,alpha=1)
> min(cv.out$cvm)
[1] 324232.9
```

4. Now, let`s try PCR model for this data set and observe CV results for different number of components using the ‘Validation function’. We have a total of 103 components. The MSEP variation for number of components is plotted as below. This basically says that we cannot expect any component reduction for estimating the per capita crime rate for this data set.

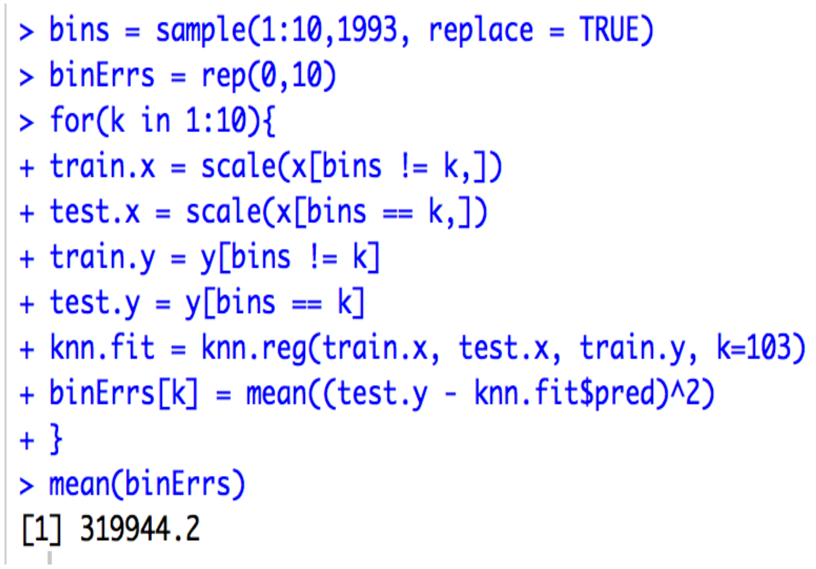


5. Since we have determined that dimension reduction will not work with this data set let`s try a KNN-regression taking a sample of data to train (say 1000 records) the model and calculate the MSE. The MSE obtained = 333363.30



```
> # selecting set of training rows for CV
> train = sample(1:1993,1000)
> train.x = scale(x[train,])
> test.x = scale(x[-train,])
> train.y = y[train]
> test.y = y[-train]
> knn.fit = knn.reg(train.x, test.x, train.y, k=103)
> mean((test.y - knn.fit$pred)^2)
[1] 333363.3
```

6. If we divided the data in to individual bins and try KNN we get a still better result = 319944.20 which is much better MSE.



```
> bins = sample(1:10,1993, replace = TRUE)
> binErrs = rep(0,10)
> for(k in 1:10){
+ train.x = scale(x[bins != k,])
+ test.x = scale(x[bins == k,])
+ train.y = y[bins != k]
+ test.y = y[bins == k]
+ knn.fit = knn.reg(train.x, test.x, train.y, k=103)
+ binErrs[k] = mean((test.y - knn.fit$pred)^2)
+ }
> mean(binErrs)
[1] 319944.2
```

Conclusion:

On observing all the above obtain results we can conclude that **Ridge is better that Lasso** for the given Data set.