K-MEANS, HIERARCHICAL CLUSTERING AND DECISION TREE CLASSIFICATION MODELING

A thorough comparison of wheat seed data

By:
Yolanda Lewis

Scope

This report will focus on K-means cluster analysis and decision tree classification models. A wheat seed dataset will be utilized for the analysis. K-means clustering will be performed varying the number of clusters from 3 through 6 and reported. Hierarchical clustering will also be performed and compared to the K-means outcome. Lastly, a decision tree classification model will be utilized to label the varieties of wheat within the data.

Dataset

The dataset consists of 7 variables representing different size measurements for three varieties of wheat seeds. The following variables are present:
1. Area
2. Perimeter
3. Compactness
4. Kernel Length
5. Kernel Width
6. Asymmetry coefficient
7. Kernel Groove Length

Data Cleaning

After importing the seeds dataset, missing values where noted in the following attributes:

3. compactness C = 4*pi*A/P^2
5. width of kernel
7. length of kernel groove
8. The class label attribute

The missing values appeared to be a format error where the values are placed in incorrect columns due to formatting, so a manual approach was utilized for filling in the missing values. This was feasible because the dataset is not very large, and the missing values appeared to be already in the dataset, but misplaced due to a possible formatting issue.

Variables were named based on the attribute information provided. In this report, the class labels 1, 2 , and 3 represent the three varieties of wheat: Kama, Rosa, and Canadian. R and SPSS where used for the analysis in this report.

Data Analysis

Before performing K-means a summary of each attribute in the dataset (central tendency and the mean value of each attribute) was taken to get an understanding of the distribution of the data before normalization. See figure 1.

Figure 1. Five Number Summary

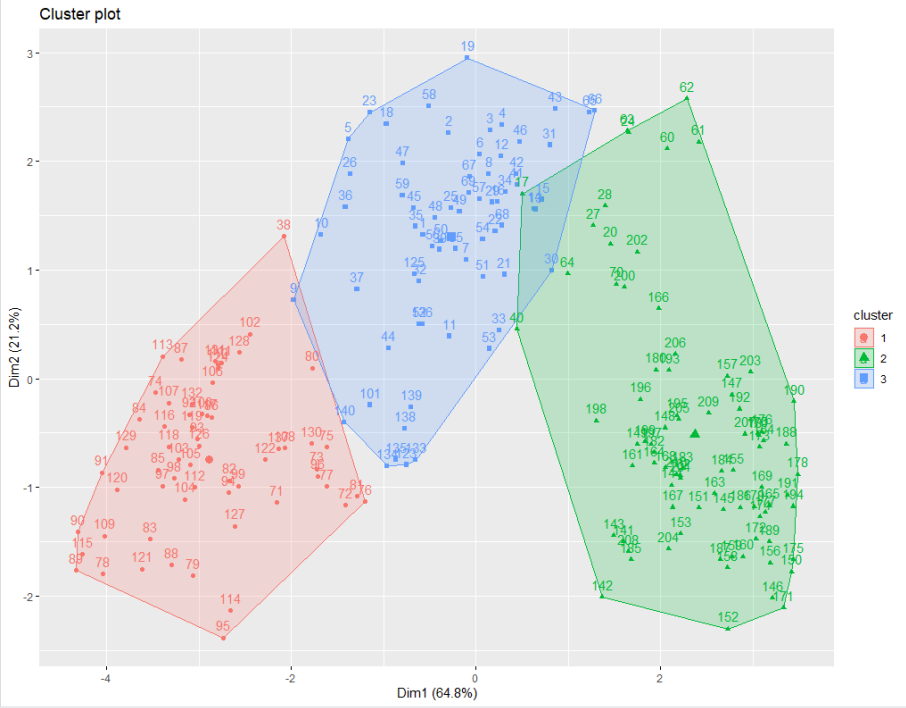
```
> #Get a summary of the Data
> summary(seeds_dataset)
      Area          Perimeter       Compactness     Kernel_Length
 Min.   :10.59   Min.   :12.41   Min.   :0.8081   Min.   :4.899
 1st Qu.:12.27   1st Qu.:13.45   1st Qu.:0.8569   1st Qu.:5.262
 Median :14.36   Median :14.32   Median :0.8734   Median :5.524
 Mean   :14.85   Mean   :14.56   Mean   :0.8710   Mean   :5.629
 3rd Qu.:17.30   3rd Qu.:15.71   3rd Qu.:0.8878   3rd Qu.:5.980
 Max.   :21.18   Max.   :17.25   Max.   :0.9183   Max.   :6.675
  Kernel_Width   Asymmetry_Coefficient kernel_Groove_Length  Class_label
 Min.   :2.630   Min.   :0.7651        Min.   :4.519        Min.   :1
 1st Qu.:2.944   1st Qu.:2.5615        1st Qu.:5.045        1st Qu.:1
 Median :3.237   Median :3.5990        Median :5.223        Median :2
 Mean   :3.259   Mean   :3.7002        Mean   :5.408        Mean   :2
 3rd Qu.:3.562   3rd Qu.:4.7687        3rd Qu.:5.877        3rd Qu.:3
 Max.   :4.033   Max.   :8.4560        Max.   :6.550        Max.   :3
>
```

The cluster centers are calculated using the mean of value of all the points in the cluster. The similarity measure used was the Euclidean distance. The Euclidean distance will allow the high similarity of variables in each cluster and the similarity between clusters to be low.

I will begin by reporting results for when **K=3.** (Please note this is for the data prior to normalization, for comparison, k-means performed on normalized data for the selected K will be reported later in the report).

Figure 2. K=3

```
> fviz_nbclust(seeds_dataset[,1:7],kmeans,method = "wss")
> str(three_clusters)
List of 9
 $ cluster     : int [1:210] 3 3 3 3 3 3 3 3 3 3 ...
 $ centers     : num [1:3, 1:7] 18.7 12 14.8 16.3 13.3 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:3] "1" "2" "3"
  .. ..$ : chr [1:7] "Area" "Perimeter" "Compactness" "Kernel_Length" ...
 $ totss       : num 2720
 $ withinss    : num [1:3] 184 238 167
 $ tot.withinss: num 589
 $ betweenss   : num 2131
 $ size        : int [1:3] 61 82 67
 $ iter        : int 3
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
> print(three_clusters)
K-means clustering with 3 clusters of sizes 61, 82, 67

Cluster means:
      Area Perimeter Compactness Kernel_Length Kernel_Width Asymmetry_Coefficient kernel_Groove_Length
1 18.72180  16.29738   0.8850869      6.208934     3.722672              3.603590             6.066098
2 11.98866  13.28439   0.8527366      5.227427     2.880085              4.583927             5.074244
3 14.81910  14.53716   0.8805224      5.591015     3.299358              2.706585             5.217537

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 2 3 3 3 3 2 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [54] 3 3 3 3 3 3 3 2 2 2 2 2 3 3 3 3 3 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[107] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 1 1 1 1 1 1 1 1 3 3 3 3 1 3 3 3 2 2 2 2 2 2 2 2 2
[160] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 184.1086 237.8538 166.8196
 (between_SS / total_SS =  78.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
>
```



As seen above the final cluster centers for each variable are as follows:

```
> three_clusters$centers
      Area Perimeter Compactness Kernel_Length Kernel_Width Asymmetry_Coefficient kernel_Groove_Length
1 18.72180  16.29738   0.8850869      6.208934     3.722672              3.603590             6.066098
2 11.98866  13.28439   0.8527366      5.227427     2.880085              4.583927             5.074244
3 14.81910  14.53716   0.8805224      5.591015     3.299358              2.706585             5.217537
```

Number of elements in each cluster

```
> three_clusters$size
[1] 61 82 67
> 
```

Cluster 1 = 61, Cluster 2= 82, Cluster 3= 67

The class distribution within each cluster

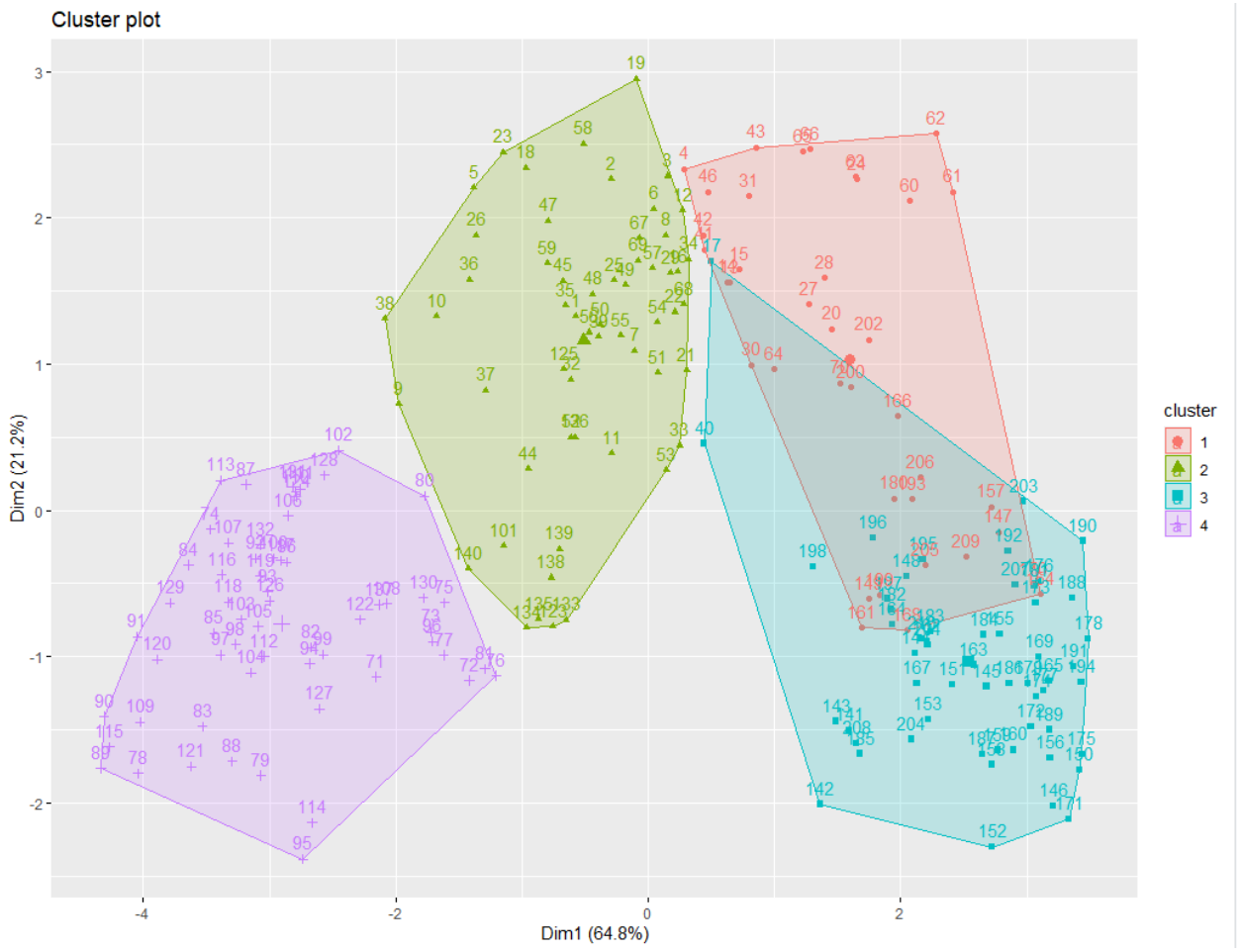
```
> table(seeds_dataset$Class_label,three_clusters$cluster)

    1  2  3
1   1 12 57
2  60  0 10
3   0 70  0
```

The rows above represent the classes in the dataset and how many of the classes are in each cluster.
The columns represent the clusters and how many of each class (1-3) are in each cluster.
Values for class distribution are shown in the table above.

Next the K-means result for K=4 is reported.

Figure 3. K=4

```
> str(four_clusters)
List of 9
 $ cluster     : int [1:210] 2 2 2 1 2 2 2 2 2 2 ...
 $ centers     : num [1:4, 1:7] 12.6 15.2 11.9 18.7 13.5 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:4] "1" "2" "3" "4"
  .. ..$ : chr [1:7] "Area" "Perimeter" "Compactness" "Kernel_Length" ...
 $ totss       : num 2720
 $ withinss    : num [1:4] 60 122 112 180
 $ tot.withinss: num 474
 $ betweenss   : num 2246
 $ size        : int [1:4] 38 56 56 60
 $ iter        : int 2
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
> print(four_clusters)
K-means clustering with 4 clusters of sizes 38, 56, 56, 60

Cluster means:
      Area Perimeter Compactness Kernel_Length Kernel_Width Asymmetry_Coefficient kernel_Groove_Length
1 12.58842  13.50000   0.8671184       5.256447     2.994553              2.823713             4.960368
2 15.15071  14.69821   0.8809000       5.645036     3.339946              2.771662             5.288911
3 11.89696  13.26054   0.8490429       5.235929     2.858518              5.315446             5.115607
4 18.74917  16.31267   0.8847067       6.215217     3.723333              3.614383             6.075800

Clustering vector:
  [1] 2 2 2 1 2 2 2 2 2 2 2 2 1 1 1 2 3 2 2 1 2 2 2 1 2 2 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 3 1 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 2 2 2 1 4 4 4 4 4 4
 [76] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 2 4 2 2 2 3 3 3 3 3 3 1 3 1 3
[151] 3 3 3 1 3 3 1 3 3 3 3 1 3 3 3 3 1 3 1 3 1 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 1 1 3 1 3 3 1 1 3 3 1 3
Within cluster sum of squares by cluster:
[1]  60.03513 121.74853 112.33646 179.59511
 (between_SS / total_SS =  82.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"         "ifault"
> |
```



Final cluster centers

As seen above the final cluster centers for each variable are as follows:

```
Cluster means:
      Area Perimeter Compactness Kernel_Length Kernel_Width Asymmetry_Coefficient kernel_Groove_Length
1 12.58842  13.50000   0.8671184      5.256447     2.994553              2.823713             4.960368
2 15.15071  14.69821   0.8809000      5.645036     3.339946              2.771662             5.288911
3 11.89696  13.26054   0.8490429      5.235929     2.858518              5.315446             5.115607
4 18.74917  16.31267   0.8847067      6.215217     3.723333              3.614383             6.075800
```

Number of elements in each cluster

```
 $ size        : int [1:4] 38 56 56 60
```

Cluster 1 = 38, Cluster 2= 56, Cluster 3= 56, Cluster 4 = 60

The class distribution within each cluster

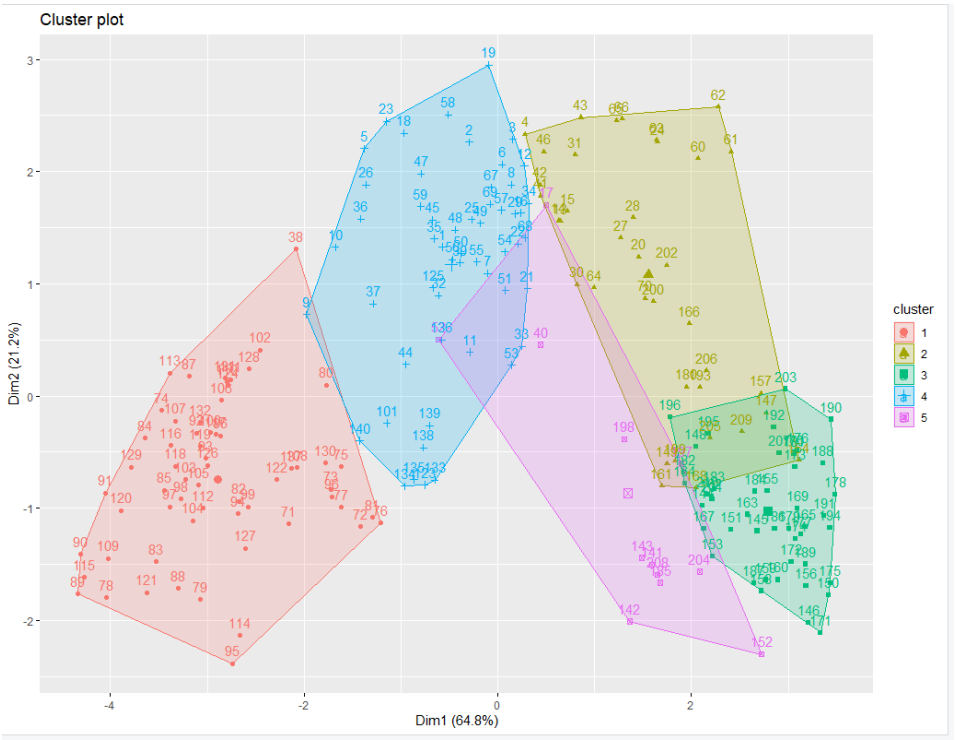
```
> table(seeds_dataset$Class_label,four_clusters$cluster)

    1  2  3  4
1  22 46  2  0
2   0 10  0 60
3  16  0 54  0
>
```

The rows represent the classes in the dataset and how many of the classes are in each cluster.
The columns represent the clusters and how many of each class (1-3) are in each cluster.
Values for class distribution are shown in the table above.

Next the K-means result for K=4 is reported.

Figure 4. K= 5

```
> str(five_clusters)
List of 9
 $ cluster     : int [1:210] 4 4 4 2 4 4 4 4 4 4 ...
 $ centers     : num [1:5, 1:7] 18.7 12.6 11.6 15.1 13.4 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:5] "1" "2" "3" "4" ...
  .. ..$ : chr [1:7] "Area" "Perimeter" "Compactness" "Kernel_Length" ...
 $ totss       : num 2720
 $ withinss    : num [1:5] 184.1 57.3 51.2 108.6 28.7
 $ tot.withinss: num 430
 $ betweenss   : num 2290
 $ size        : int [1:5] 61 37 46 54 12
 $ iter        : int 3
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
> print(five_clusters)
K-means clustering with 5 clusters of sizes 61, 37, 46, 54, 12

Cluster means:
      Area Perimeter Compactness Kernel_Length Kernel_Width Asymmetry_Coefficient kernel_Groove_Length
1 18.72180  16.29738   0.8850869      6.208934     3.722672              3.603590             6.066098
2 12.62486  13.51351   0.8679622      5.260946     3.002108              2.804868             4.956919
3 11.57630  13.13283   0.8430457      5.206978     2.797022              5.018087             5.107543
4 15.10333  14.68167   0.8801889      5.641037     3.331852              2.716002             5.288130
5 13.39500  13.86583   0.8745417      5.371250     3.130250              6.328917             5.145917

Clustering vector:
  [1] 4 4 4 2 4 4 4 4 4 4 4 4 2 2 2 4 5 4 4 2 4 4 4 2 4 4 2 2 4 2 2 4 4 4 4 4 4 1 4 5 2 2 2 4 4 2 4 4 4 4 4 5 4 4 4 4 4 4 4
 [60] 2 2 2 2 2 2 2 4 4 4 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[119] 1 1 1 1 4 1 4 1 1 1 1 1 1 1 4 4 4 4 1 4 4 4 5 5 5 3 3 3 2 3 2 3 3 5 3 2 2 3 3 2 3 3 3 2 3 3 3 2 3 2 3 3 3 3 3 3 3 3 3
[178] 3 3 2 3 3 3 3 5 3 3 3 3 3 3 2 3 3 3 5 5 2 2 3 2 3 5 2 2 3 5 2 3

Within cluster sum of squares by cluster:
[1] 184.10858  57.28519  51.23419 108.59438  28.72497
 (between_SS / total_SS =  84.2 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
>
```



Final cluster centers

As seen above the final cluster centers for each variable are as follows:

```
Cluster means:
      Area Perimeter Compactness Kernel_Length Kernel_Width Asymmetry_Coefficient kernel_Groove_Length
1 18.72180  16.29738   0.8850869      6.208934     3.722672              3.603590             6.066098
2 12.62486  13.51351   0.8679622      5.260946     3.002108              2.804868             4.956919
3 11.57630  13.13283   0.8430457      5.206978     2.797022              5.018087             5.107543
4 15.10333  14.68167   0.8801889      5.641037     3.331852              2.716002             5.288130
5 13.39500  13.86583   0.8745417      5.371250     3.130250              6.328917             5.145917
```

Number of elements in each cluster

```
K-means clustering with 5 clusters of sizes 61, 37, 46, 54, 12
```

Cluster 1 = 61, Cluster 2= 37, Cluster 3= 46, Cluster 4= 54, Cluster 5= 12

The class distribution within each cluster

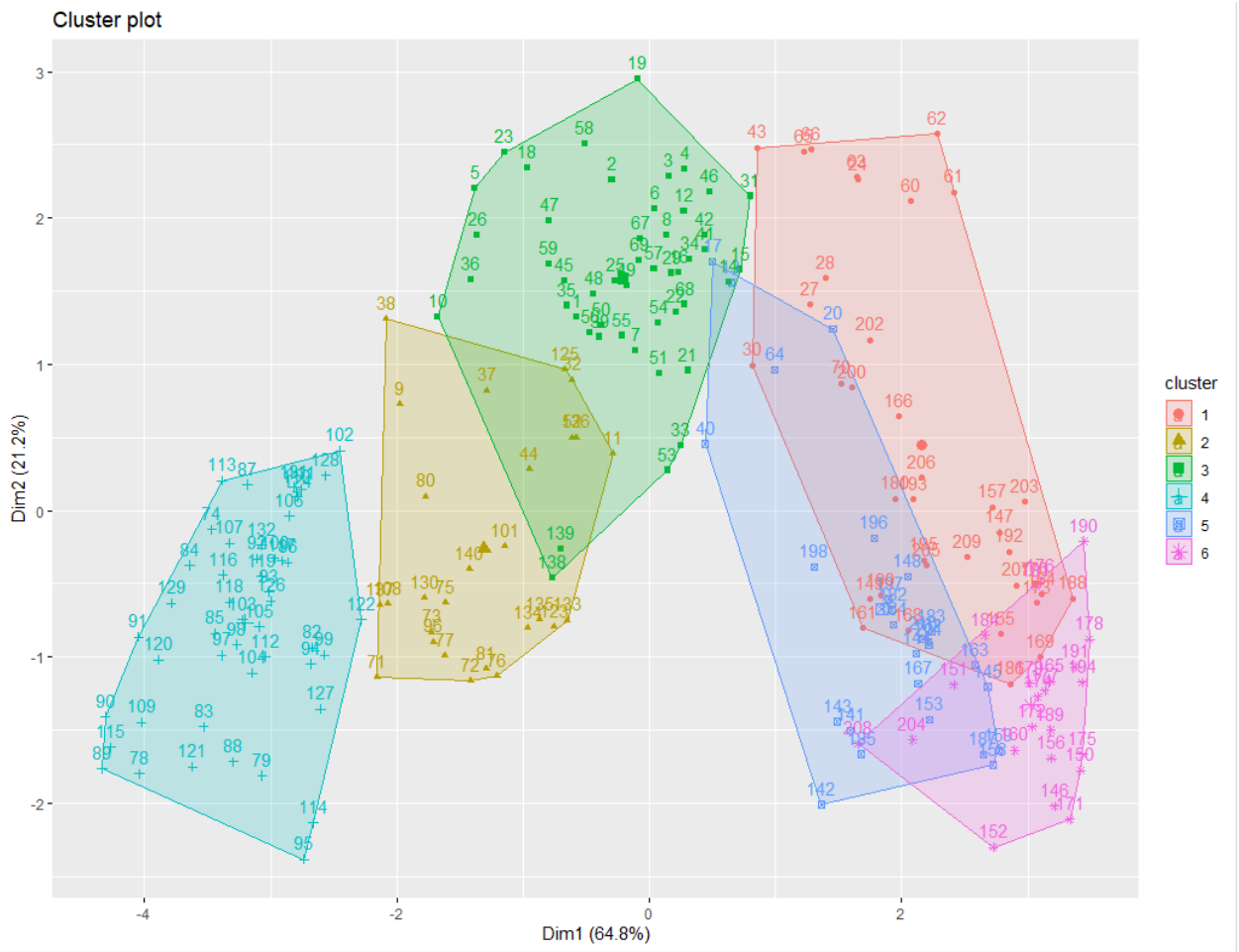
```
> table(seeds_dataset$Class_label,five_clusters$cluster)

    1  2  3  4  5
1   1 22  0 44  3
2  60  0  0 10  0
3   0 15 46  0  9
```

The rows represent the classes in the dataset and how many of the classes are in each cluster.
The columns represent the clusters and how many of each class (1-3) are in each cluster.
Values for class distribution are shown in the table above.

Lastly is when K= 6

Figure 5. K= 6

```
> print(six_clusters)
K-means clustering with 6 clusters of sizes 37, 27, 48, 48, 27, 23

Cluster means:
      Area Perimeter Compactness Kernel_Length Kernel_Width Asymmetry_Coefficient
1 12.03811  13.26865   0.8585865      5.195595     2.900270              3.159622
2 16.47852  15.34778   0.8787963      5.868074     3.476148              4.064778
3 14.67875  14.47000   0.8805979      5.573042     3.282771              2.329275
4 19.15104  16.46917   0.8870896      6.268854     3.772937              3.460417
5 12.68704  13.61926   0.8589519      5.327704     2.984000              5.054889
6 11.35957  13.01391   0.8423391      5.176435     2.778217              5.913043
  kernel_Groove_Length
1             4.979946
2             5.696185
3             5.160313
4             6.127250
5             5.140704
6             5.088609

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 2 3 2 3 5 3 3 3 5 3 3 5 3 3 3 1 3 3 1 1 3 1 3 2 3 3 3 2 2 3 5 3 3 1 2 3 3 3
 [49] 3 3 3 2 3 3 3 3 3 3 3 1 1 1 1 5 1 1 3 3 3 1 2 2 2 4 2 2 2 4 4 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 2
 [97] 4 4 4 4 2 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 2 4 4 4 4 2 4 4 2 2 2 2 2 3 3 2 5 5 5 5
[145] 5 6 1 5 1 6 6 6 5 1 1 6 1 5 5 6 1 5 5 5 6 1 5 1 1 1 6 6 1 6 6 6 6 6 6 1 6 5 5 6 5 1 5 1 6 6 6 1
[193] 1 6 1 5 5 5 1 1 5 1 1 6 1 1 1 6 1 5

Within cluster sum of squares by cluster:
[1]  52.73839  41.09715  74.52075 118.21163  30.72904  37.61631
 (between_SS / total_SS =  87.0 %)
```



Final cluster centers

As seen above the final cluster centers for each variable are as follows:

```
Cluster means:
      Area Perimeter Compactness Kernel_Length Kernel_Width Asymmetry_Coefficient
1 12.03811  13.26865   0.8585865      5.195595     2.900270              3.159622
2 16.47852  15.34778   0.8787963      5.868074     3.476148              4.064778
3 14.67875  14.47000   0.8805979      5.573042     3.282771              2.329275
4 19.15104  16.46917   0.8870896      6.268854     3.772937              3.460417
5 12.68704  13.61926   0.8589519      5.327704     2.984000              5.054889
6 11.35957  13.01391   0.8423391      5.176435     2.778217              5.913043
  kernel_Groove_Length
1             4.979946
2             5.696185
3             5.160313
4             6.127250
5             5.140704
6             5.088609
```

Number of elements in each cluster

```
K-means clustering with 6 clusters of sizes 37, 27, 48, 48, 27, 23
```

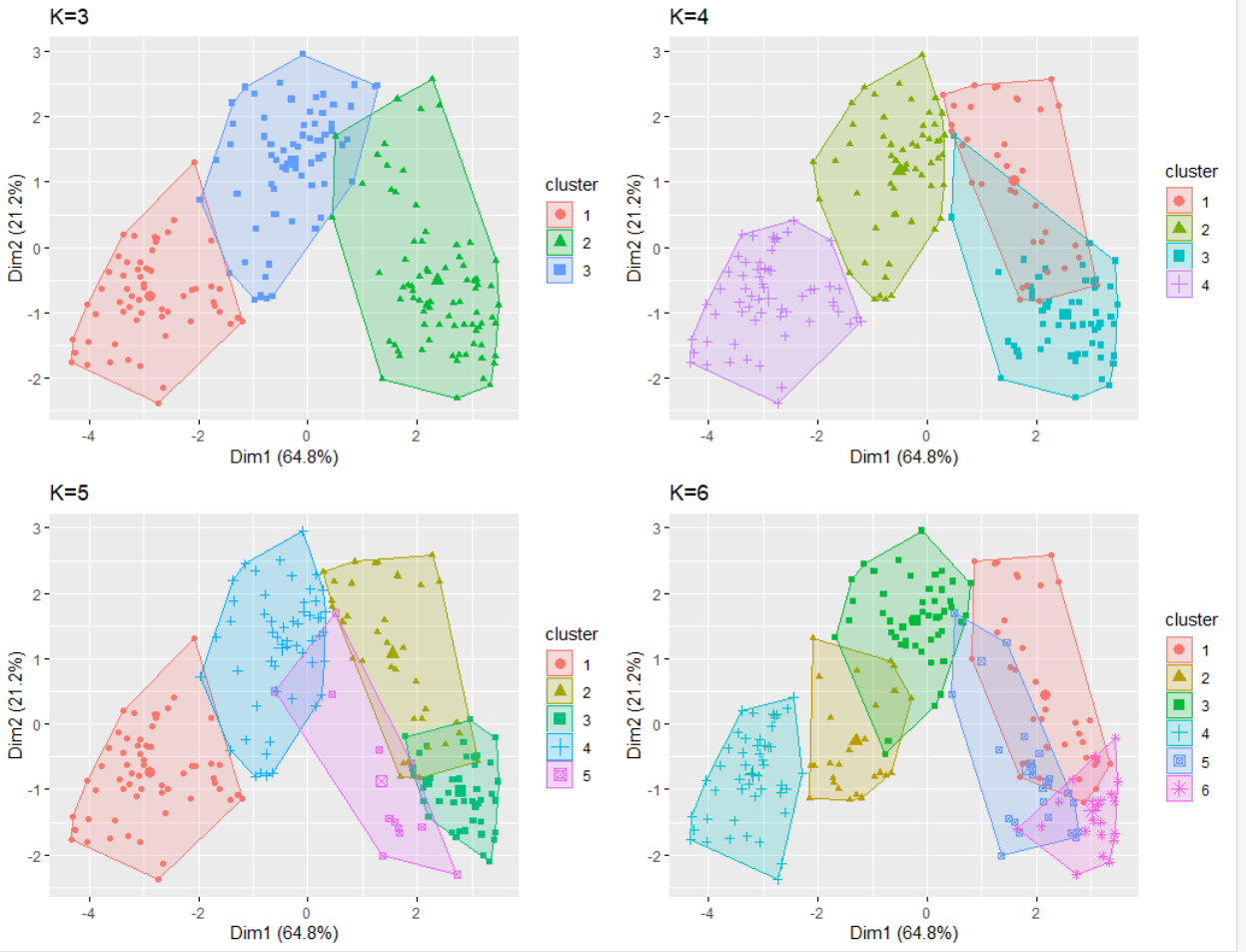
Cluster 1 = 37, Cluster 2= 27, Cluster 3= 48, Cluster 4 = 48, Cluster5 = 27, Cluster 6 = 23

The class distribution within each cluster

```
> table(seeds_dataset$Class_label,six_clusters$cluster)

    1  2  3  4  5  6
  1 12  7 46  0  5  0
  2  0 20  2 48  0  0
  3 25  0  0  0 22 23
>
```

The rows represent the classes in the dataset and how many of the classes are in each cluster.
The columns represent the clusters and how many of each class (1-3) are in each cluster.
Values for class distribution are shown in the table above.

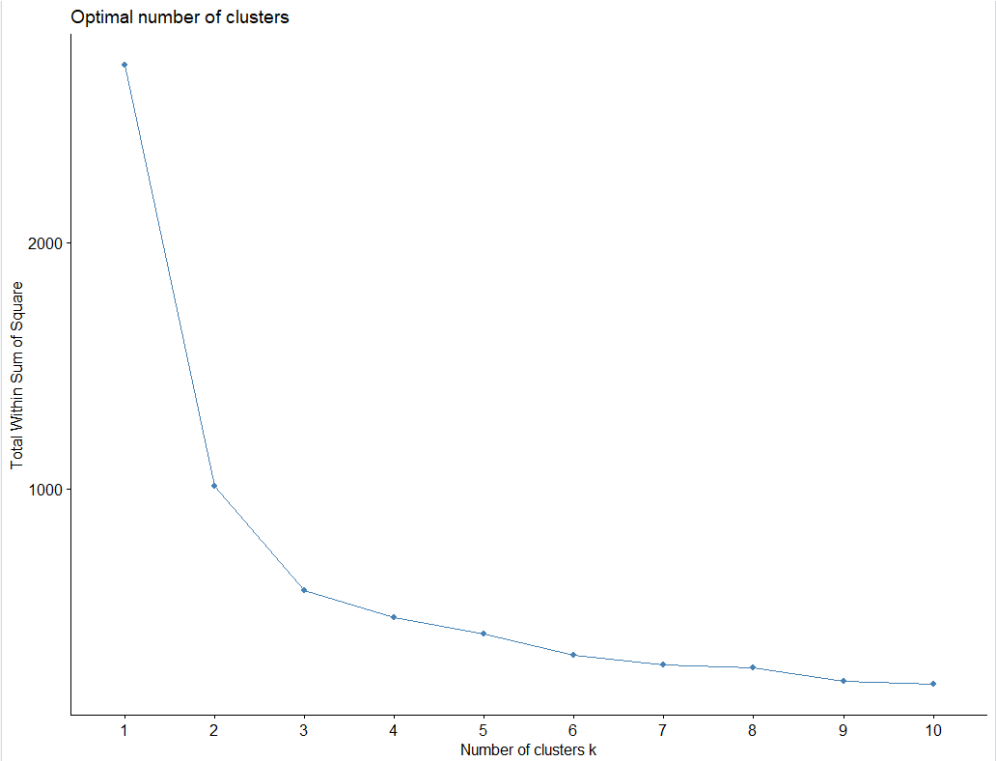
To get a better overall view of the different K means plots see the grids below in figure 6:



In my opinion K= 3 should be selected. This is because the total within-cluster sum of square should be minimized, meaning we want the compactness of the clustering to be as small as possible. K=3 does this. Despite this, to prove this further the elbow method was used to determine the optimal K.. See the graph 1 below:

```
#Determine which K should be selected using the elbow method
set.seed(125)

#Use elbow method to find the optimal K value
fviz_nbclust(seeds_dataset[,1:7],kmeans,method = "wss")
```

Graph 1. Elbow Method



There is a large reduction from K=1 to K=3. In my opinion K=3 shows the optimal number of clusters because it appears to be the bend in the elbow

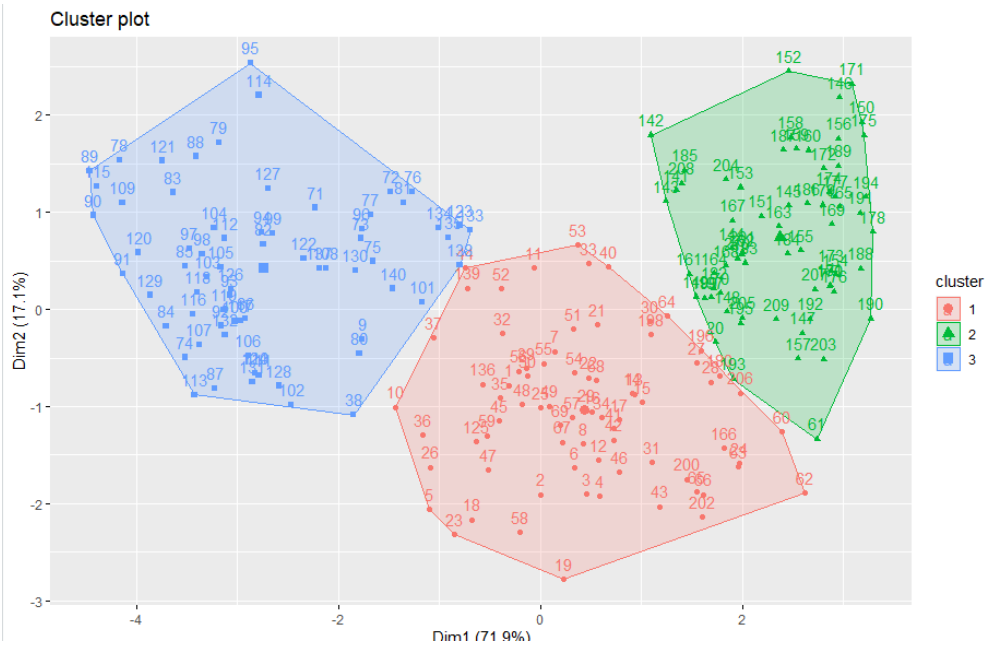
Normalization of Data to improve clustering

The data was normalized using the scale function in R:

```
> #Part IV: normalized the data
> norm_seeds_dataset <- scale(seeds_dataset[,1:7])
> head(norm_seeds_dataset)
           Area    Perimeter  Compactness Kernel_Length Kernel_Width Asymmetry_Coefficient kernel_Groove_Length
[1,]  0.14175904  0.214948819 6.045733e-05    0.30349301    0.1413640           -0.9838010          -0.3826631
[2,]  0.01116136  0.008204153 4.274938e-01   -0.16822270    0.1969616           -1.7839036          -0.9198156
[3,] -0.19160873 -0.359341919 1.438945e+00   -0.76181710    0.2075516           -0.6658882          -1.1863572
[4,] -0.34626388 -0.474200066 1.036904e+00   -0.68733567    0.3187467           -0.9585276          -1.2270506
[5,]  0.44419577  0.329806966 1.371233e+00    0.06650665    0.8032397           -1.5597684          -0.4742231
[6,] -0.16067770 -0.267455401 1.019976e+00   -0.54740087    0.1413640           -0.8235144          -0.9198156
>
```

K-means was performed on the optimal K (K=3) using the normalized dataset to see if the normalization influences the clustering results. A seed was set to ensure each time the code is ran ; the same answer is generated.

Figure 7. Normalized Data with K= 3



It appears that normalization influenced the clustering results for when k=3. It appears that the size of the clusters changed from [61,82,67] to [75,66,69]. The with-in-sum of squares(WSS) also changed from the from the following:

Un-normalized data

```
within cluster sum of squares by cluster:
[1] 184.1086 237.8538 166.8196
 (between_SS / total_SS =  78.4 %)
```

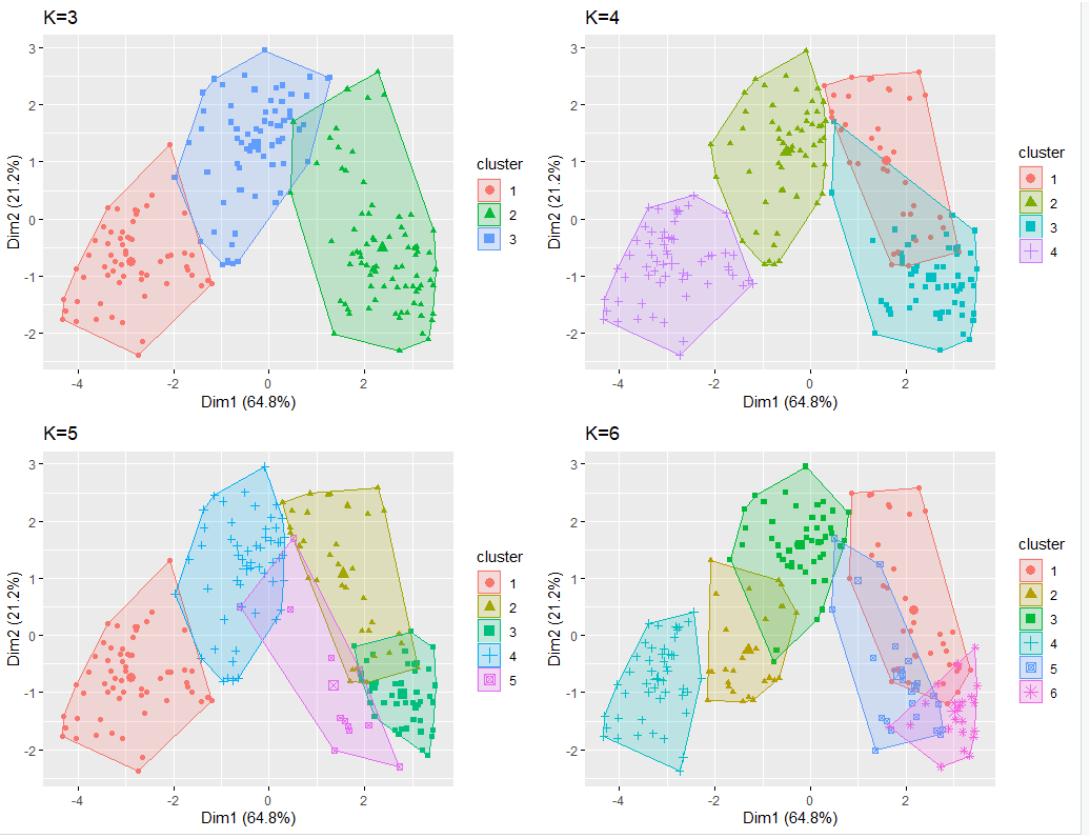
Normalized data

```
within cluster sum of squares by cluster:
[1] 160.7960 118.2489 149.8787
 (between_SS / total_SS =  70.7 %)
```

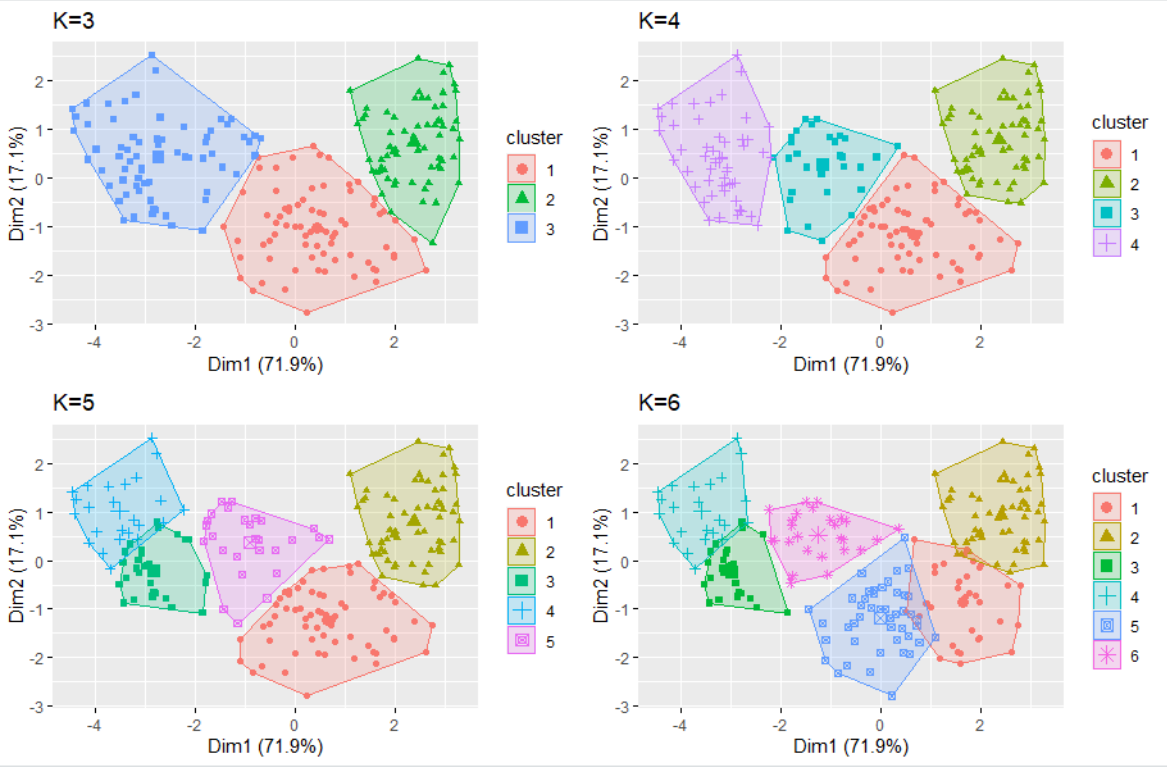
The results changed slightly for k=3 when the data was normalized. The WSS changed which means that the squared distance between each member of the cluster and its centroid decreased. The mean values also changed significantly. So, I would say normalization helps to improve the clustering results

To see a comparison on how the overall data changed for each K when the data was normalized, so I plotted each K to visualize the change and re-performed the elbow method with the normalized data. In my opinion it appears that the total variation within the clusters seems to decrease after K-means for K=4 is performed. See below:

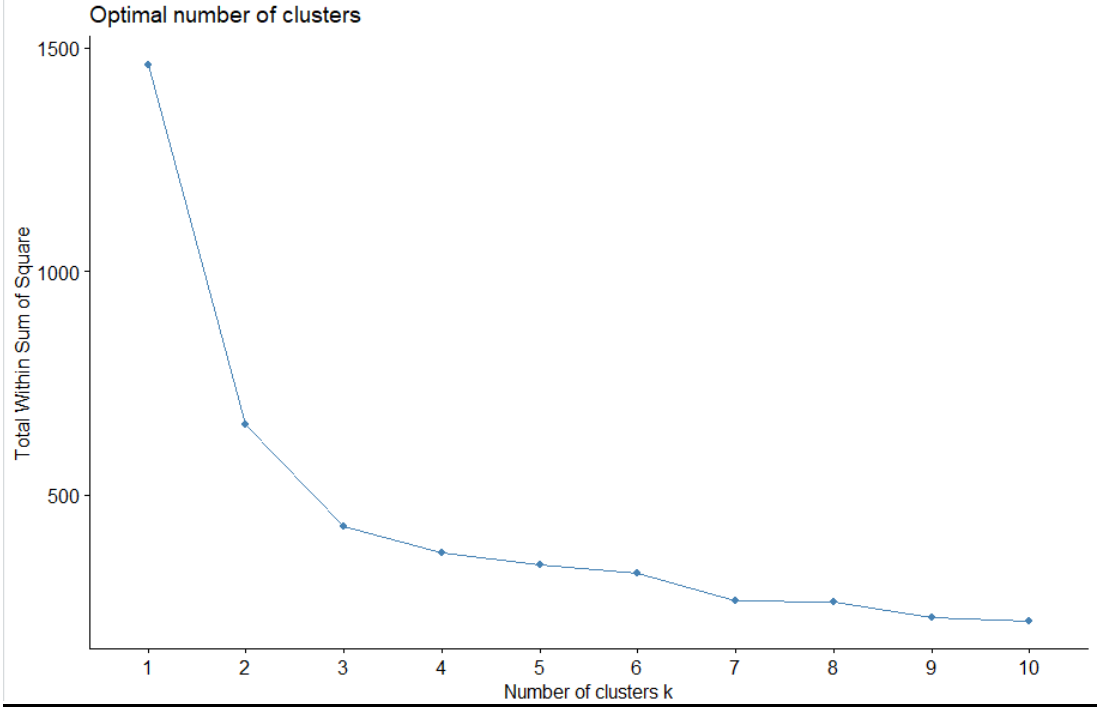
Unnormalized Dataset



Normalized Dataset



Elbow Method with standardized Data

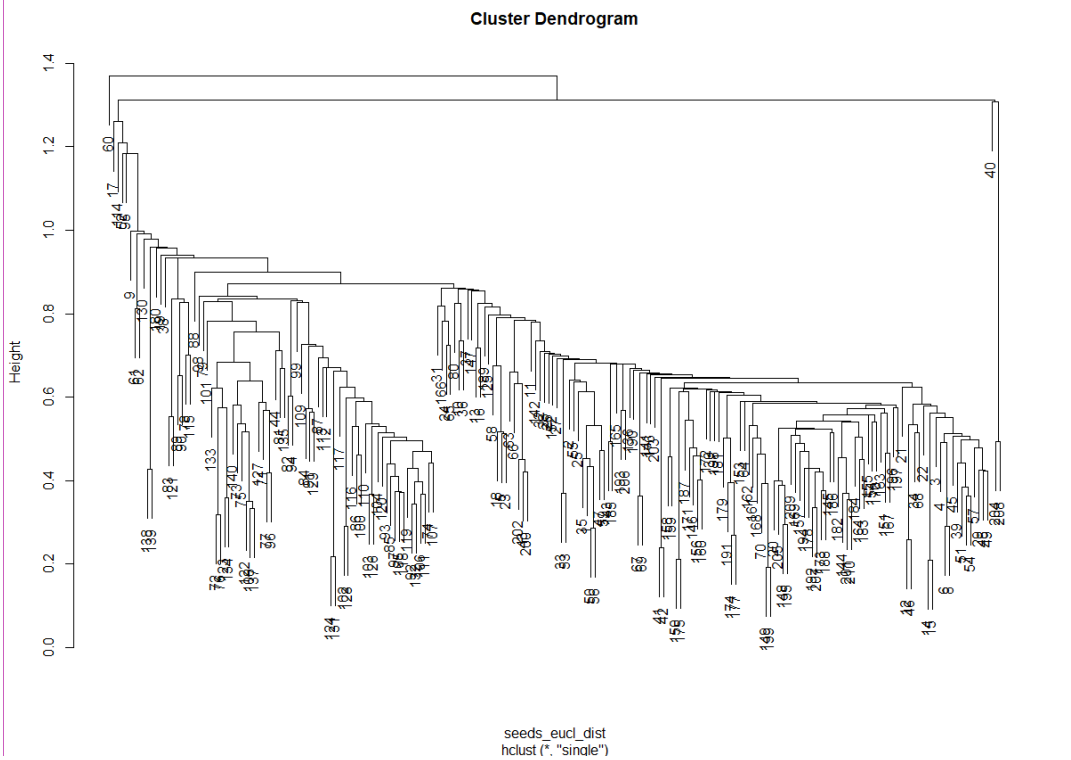


Hierarchical Clustering

Hierarchical clustering was performed to compare the classification algorithm performance. A single linkage algorithm result is reported below.

The dendrogram

Single Linkage Dendrogram



The class distribution at the level of the dendrogram where there are only three clusters.

This is the total in each cluster

```
> count(seeds_single_dend,cluster)
# A tibble: 3 x 2
  cluster     n
    <int> <int>
1       1   206
2       2     3
3       3     1
> |
```

This is the class distribution where there are only three clusters.

```
> table(seeds_single_dend$cluster,seeds_dataset$Class_label)

    1  2  3
1  68 70 68
2   1  0  2
3   1  0  0
```

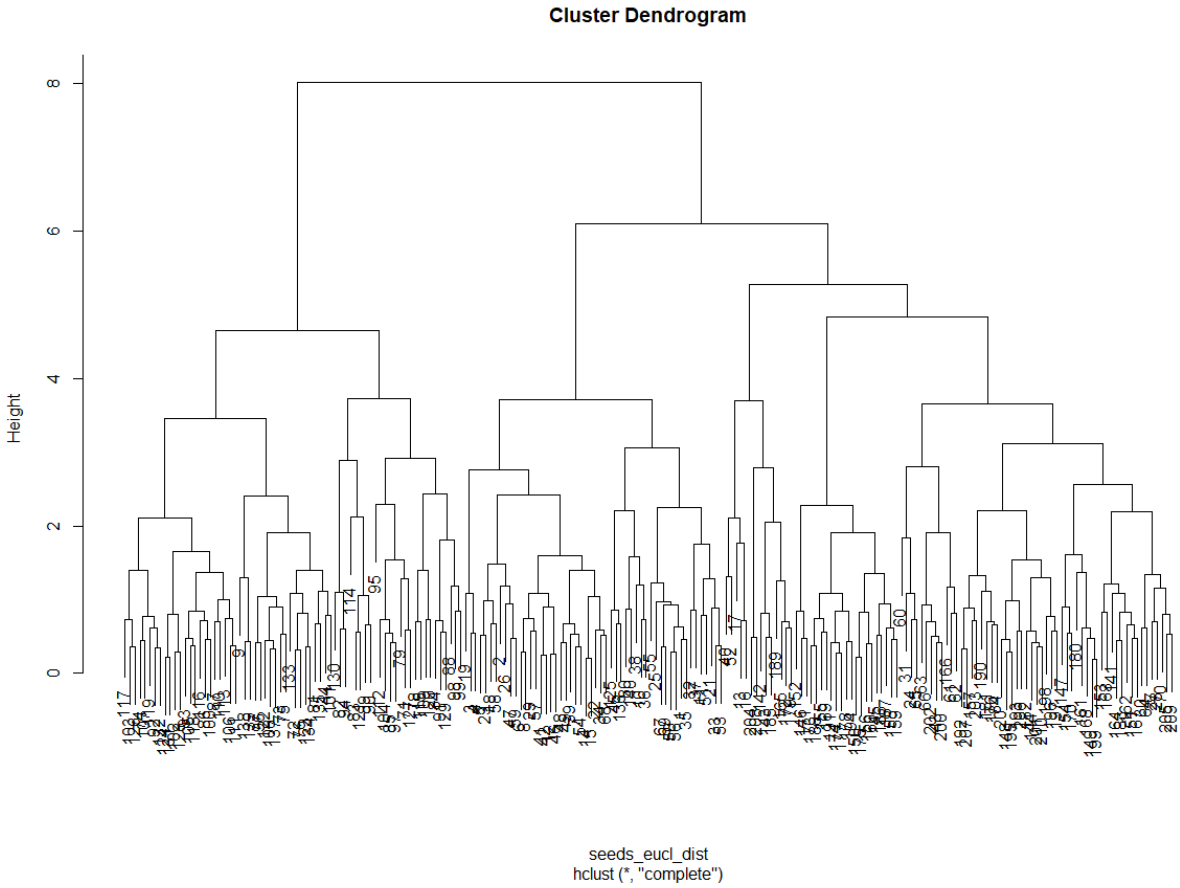
Rows in the table are the clusters, and the columns are the class labels (1-3)

Cluster 1= 68 seeds from class 1, 70 seeds from class 2, and 68 seeds from class 3

Cluster 2= 1 seed from class 1, 0 seeds from class 2, and 2 seeds from class 3
Cluster 3 = 1 seed from class 1, 0 seeds from class 2, and 0 seeds from class 3

Next a complete linkage dendrogram is utilized.

Complete Linkage Dendrogram



The class distribution at level of the dendrogram where there are only three clusters.

```
> count(seeds_complete_dend,cluster)
# A tibble: 3 x 2
  cluster     n
    <int> <int>
1       1    52
2       2    68
3       3    90
> table(seeds_complete_dend$cluster,seeds_dataset$Class_label)

    1  2  3
1 48  4  0
2  2 66  0
3 20  0 70
```

Rows in the table are the clusters, and the columns are the class labels (1-3)
Cluster 1= 48 seeds from class 1, 4 seeds from class 2, and 0 seeds from class 3
Cluster 2= 2 seed from class 1, 66 seeds from class 2, and 0 seeds from class 3
Cluster 3 = 20 seed from class 1, 0 seeds from class 2, and 70 seeds from class 3

Comparison of results with hierarchical clustering and k-means algorithm.

Class distribution using k means when k=3

```
> table(seeds_dataset$Class_label,three_clusters$cluster)

    1  2  3
1   1 12 57
2  60  0 10
3   0 70  0
```

Class distribution using single linkage when K=3

```
> table(seeds_single_dend$cluster,seeds_dataset$Class_label)

    1  2  3
1  68 70 68
2   1  0  2
3   1  0  0
```

Class distribution using complete linkage when K=3

```
> table(seeds_complete_dend$cluster,seeds_dataset$Class_label)

    1  2  3
1  48  4  0
2   2 66  0
3  20  0 70
```

Note: The rows in the tables are the clusters and the columns are the class labels

The class distribution as seen above for the hierarchical clustering verses k-means is not very good. The classes are not well distributed amongst the clusters when using the Euclidean distance as compared to when the k-means algorithm is used for clustering. For example, for single linkage most of the seeds from each class are placed into cluster 1, which is bad for clustering. For complete linkage, the clustering improved but it appears that K-means clustering did a much better job at clustering the data set.

Executive Summary

In this analysis we focused on clustering the dataset to determine the class labels. This means that the different varieties of wheat(In my dataset the different varieties of wheat are represents as 1, 2 , and 3) are placed into clusters. The clusters are formed based on the mean of points assigned to a cluster. Each record is then compared to this mean(or center) to determine if it is close to the clusters center. If it's close, the record is placed into that cluster. If it's not close, it is compared to other clusters formed in the dataset. The goal is to place all records that are most alike into the same cluster.

In this problem, each row represents a record, and each column represents a feature of the record. For example, some features of the data in each row for the dataset was area, perimeter, and length of a kernel. These features help to determine how each record should be labeled. K-means is a clustering algorithm that is used to cluster data using the values of these features. The mean(average) of each feature is

calculated and based on these mean values, a cluster center known as a centroid is formed. When determining the optimal cluster, we want what is known as the total-within-cluster sum or squares to be small. This is simply measuring the compactness of the cluster. The number of clusters is pre-defined by the user and random k clusters are selected from the data to serve as the initial center for the clusters. The remaining objects are assigned to the closest centroid using the Euclidean distance. A new mean is completed each time a new record is added to a cluster. This is done iteratively until all records have been assigned to a cluster.

Hierarchical clustering is done by using dendrograms which use linkage methods to measure the distance between clusters. In our problem we used single linage and complete linkage. Single linkage calculates the minimum distance between the clusters before they are merged. Single linkage is good for detecting high values in a dataset. Complete linkage calculates the maximum distance between the clusters before they are merged. One of main differences between k-means and hierarchical clustering is hierarchical clustering can handle many distance metrics while k-means relies on the Euclidean distance.

The K-means algorithm performed best when clustering the seed data set. The results indicated that the when the data is normalized the clustering result is improved. In my opinion the optimal k is K=4 when the data is normalized. See the elbow method results above. The results when applying hierarchical clustering did not produce clusters in which the similarity between clusters could be well observed. It also appeared that certain features such as kernel width were more dominate in certain clusters such as cluster 3. The total class distribution after using K-means verses hierarchical clustering was as follows:

K-means

```
> table(seeds_dataset$Class_label,three_clusters$cluster)

    1  2  3
1   1 12 57
2  60  0 10
3   0 70  0
```

Single Linkage

```
> table(seeds_dataset$Class_label,three_clusters$cluster)

    1  2  3
1   1 12 57
2  60  0 10
3   0 70  0
```

Complete linkage

```
> table(seeds_complete_dend$cluster,seeds_dataset$Class_label)

    1  2  3
1  48  4  0
2   2 66  0
3  20  0 70
```

To improve the clustering results, checking to see if there are any variables that are similar using PCA would be feasible. This might offer better clustering results. I would also normalize the data right away when using k-means because it avoids producing false results caused by certain features outweighing others.

Part 2: Decision Tree Classifier

A decision tree classification model for the three different varieties of wheat: Kama, Rosa and Canadian was utilized. Ten-fold cross validation and at least five different configurations are utilized to produce a decision tree classifier. Results are reported for different configurations and one is chosen as being the best among the configurations.

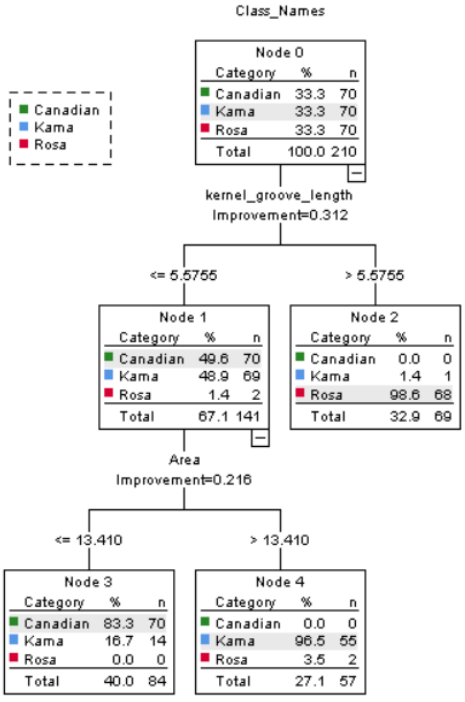
- SPSS was used for part 2 of this report.

The first tree configuration consists of the following:

Depth	Parent Cases (np)	Child Cases (nc)	Accuracy
15	100	50	91.9%

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Class_Names
	Independent Variables	Area, Perimeter, Compactness, Kernel_Length, Kernel_Width, Asymmetry_coefficient, kernel_groove_length
	Validation	Cross Validation
	Maximum Tree Depth	15
	Minimum Cases in Parent Node	100
	Minimum Cases in Child Node	50
Results	Independent Variables Included	kernel_groove_length, Perimeter, Kernel_Length, Area, Kernel_Width, Compactness, Asymmetry_coefficient
	Number of Nodes	5
	Number of Terminal Nodes	3
	Depth	2



Classification

Observed	Predicted: Canadian	Kama	Rosa	Percent Correct
Canadian	70	0	0	100.0%
Kama	14	55	1	78.6%
Rosa	0	2	68	97.1%
Overall Percentage	40.0%	27.1%	32.9%	91.9%

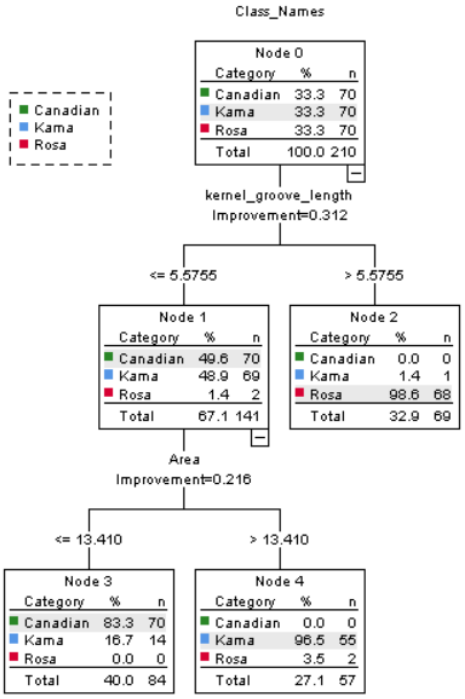
Growing Method: CRT
Dependent Variable: Class_Names

Second tree configuration

Depth	Parent Cases (np)	Child Cases (nc)	Accuracy
15	40	20	91.9%

Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Class_Names
	Independent Variables	Area, Perimeter, Compactness, Kernel_Length, Kernel_Width, Asymmetry_coefficient, kernel_groove_length
	Validation	Cross Validation
	Maximum Tree Depth	15
	Minimum Cases in Parent Node	40
	Minimum Cases in Child Node	20
Results	Independent Variables Included	kernel_groove_length, Perimeter, Kernel_Length, Area, Kernel_Width, Compactness, Asymmetry_coefficient
	Number of Nodes	5
	Number of Terminal Nodes	3
	Depth	2



Classification

Observed	Predicted: Canadi	Double-click to activate	Rosa	Percent Correct
Canadian			0	100.0%
Kama	14	55	1	78.6%
Rosa	0	2	68	97.1%
Overall Percentage	40.0%	27.1%	32.9%	91.9%

Growing Method: CRT
Dependent Variable: Class_Names

Third Tree Configuration

Depth	Parent Cases (np)	Child Cases (nc)	Accuracy
10	10	5	97.1%

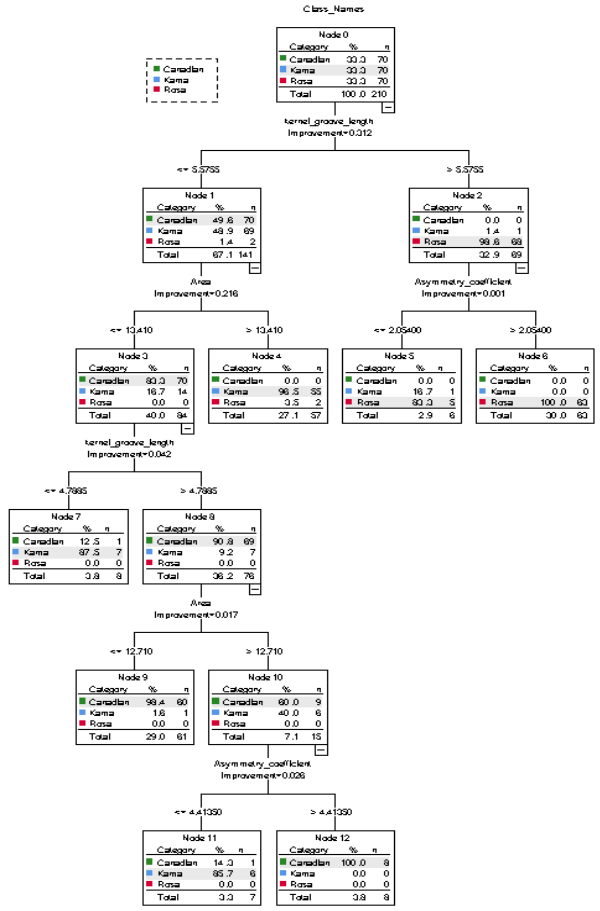
Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Class_Names
	Independent Variables	Area, Perimeter, Compactness, Kernel_Length, Kernel_Width, Asymmetry_coefficient, kernel_groove_length
	Validation	Cross Validation
	Maximum Tree Depth	10
	Minimum Cases in Parent Node	10
	Minimum Cases in Child Node	5
Results	Independent Variables Included	kernel_groove_length, Perimeter, Kernel_Length, Area, Kernel_Width, Compactness, Asymmetry_coefficient
	Number of Nodes	13
	Number of Terminal Nodes	7
	Depth	5

Classification

	Predicted			
Observed	Canadian	Kama	Rosa	Percent Correct
Canadian	68	2	0	97.1%
Kama	1	68	1	97.1%
Rosa	0	2	68	97.1%
Overall Percentage	32.9%	34.3%	32.9%	97.1%

Growing Method: CRT
Dependent Variable: Class_Names



Fourth Tree Configuration

Depth	Parent Cases (np)	Child Cases (nc)	Accuracy
5	4	2	97.1%

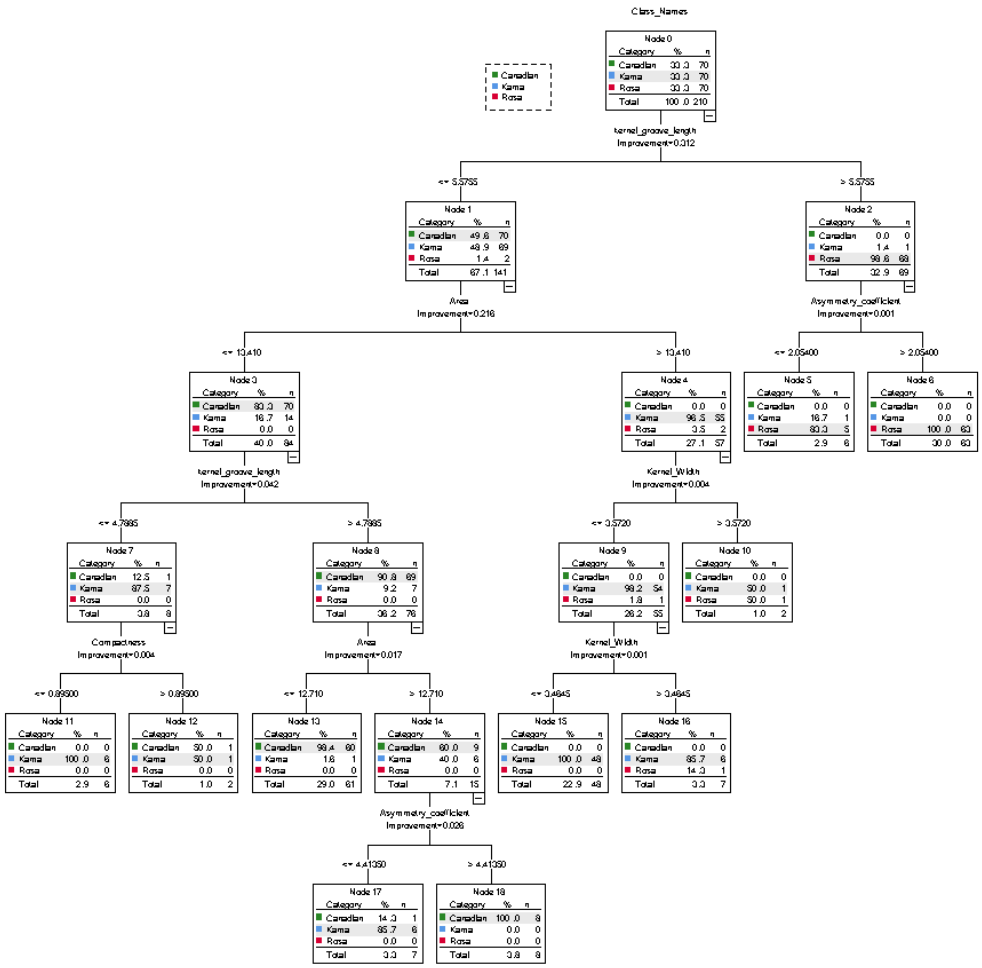
Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Class_Names
	Independent Variables	Area, Perimeter, Compactness, Kernel_Length, Kernel_Width, Asymmetry_coefficient, kernel_groove_length
	Validation	Cross Validation
	Maximum Tree Depth	5
	Minimum Cases in Parent Node	4
	Minimum Cases in Child Node	2
Results	Independent Variables Included	kernel_groove_length, Perimeter, Kernel_Length, Area, Kernel_Width, Compactness, Asymmetry_coefficient
	Number of Nodes	19
	Number of Terminal Nodes	10
	Depth	5

Classification

	Predicted			
Observed	Canadian	Kama	Rosa	Percent Correct
Canadian	68	2	0	97.1%
Kama	1	68	1	97.1%
Rosa	0	2	68	97.1%
Overall Percentage	32.9%	34.3%	32.9%	97.1%

Growing Method: CRT
Dependent Variable: Class_Names



Lastly, my fifth tree configuration

Depth	Parent Cases (np)	Child Cases (nc)	Accuracy
5	2	1	99.5%

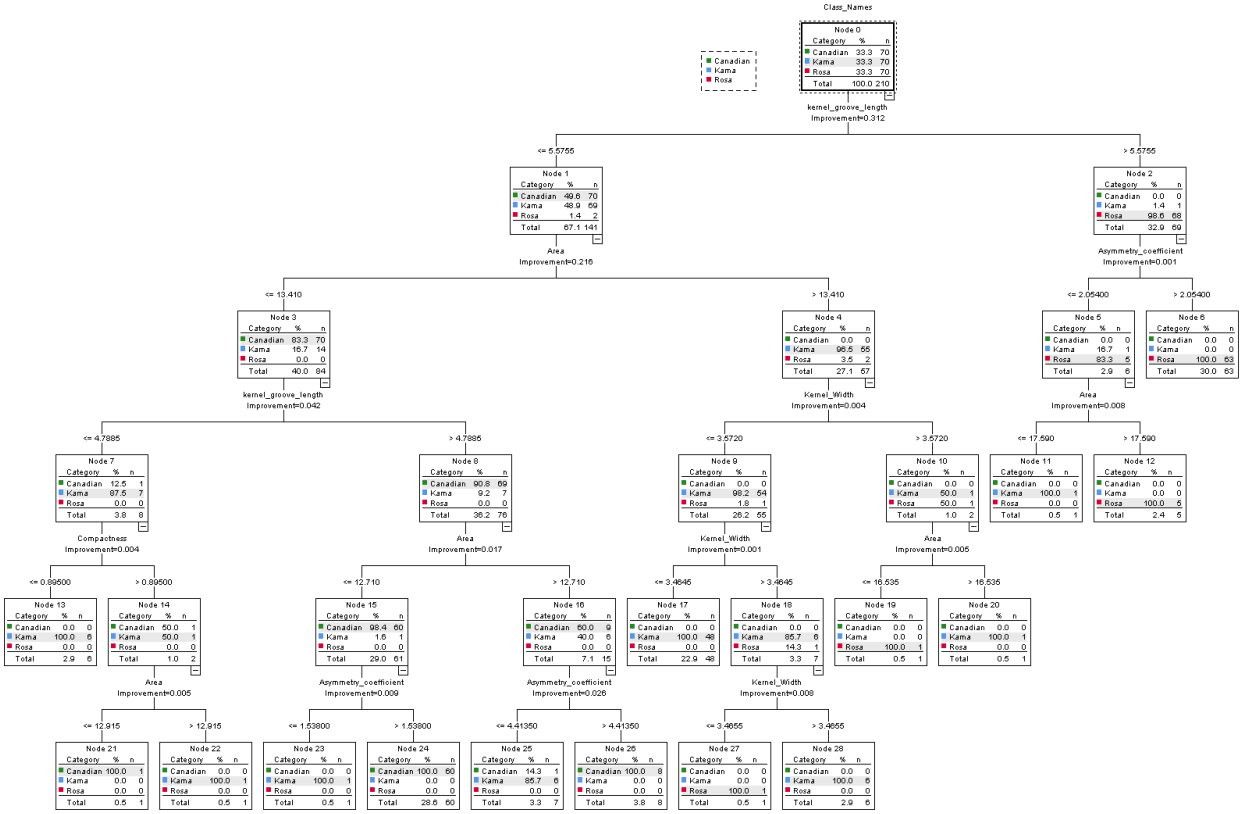
Model Summary

Specifications	Growing Method	CRT
	Dependent Variable	Class_Names
	Independent Variables	Area, Perimeter, Compactness, Kernel_Length, Kernel_Width, Asymmetry_coefficient, kernel_groove_length
	Validation	Cross Validation
	Maximum Tree Depth	5
	Minimum Cases in Parent Node	2
	Minimum Cases in Child Node	1
Results	Independent Variables Included	kernel_groove_length, Perimeter, Kernel_Length, Area, Kernel_Width, Compactness, Asymmetry_coefficient
	Number of Nodes	29
	Number of Terminal Nodes	15
	Depth	5

Classification

	Predicted			
Observed	Canadian	Kama	Rosa	Percent Correct
Canadian	69	1	0	98.6%
Kama	0	70	0	100.0%
Rosa	0	0	70	100.0%
Overall Percentage	32.9%	33.8%	33.3%	99.5%

Growing Method: CRT
Dependent Variable: Class_Names



Comparisons for all five trees are as follows:

	Tree 1	Tree 2	Tree 3	Tree 4	Tree 5
Depth	15	15	10	5	5
Np	100	40	10	4	2
Nc	50	20	5	2	1
Accuracy	91.9%	91.9%	97.1%	97.1%	99.5%

Tree 4 and Tree 5 present the best results for accuracy. When considering accuracy Tree # 5 would be the best amongst the different tree configurations tried. As the complexity of the tree increased so did the accuracy.

Tree 5 classification matrix

Classification

	Predicted			
Observed	Canadian	Kama	Rosa	Percent Correct
Canadian	69	1	0	98.6%
Kama	0	70	0	100.0%
Rosa	0	0	70	100.0%
Overall Percentage	32.9%	33.8%	33.3%	99.5%

Growing Method: CRT
Dependent Variable: Class_Names

For the Canadian wheat 69 are predicted correctly and the other varieties of wheat Kama and Rosa are all predicted correctly. The overall accuracy is 99.5%. This could be exhibiting some overfitting though.

Overall, accuracy may not be the best case in all cases because many times the accuracy depends on the quality of the data and whether the data is symmetric. If these things are not the case, the accuracy may not be the best performance metric to choose. The data might also be overfitting, causing a false result. Precision, recall, F-measure, or specificity might be good options to use when evaluating the performance of a model.

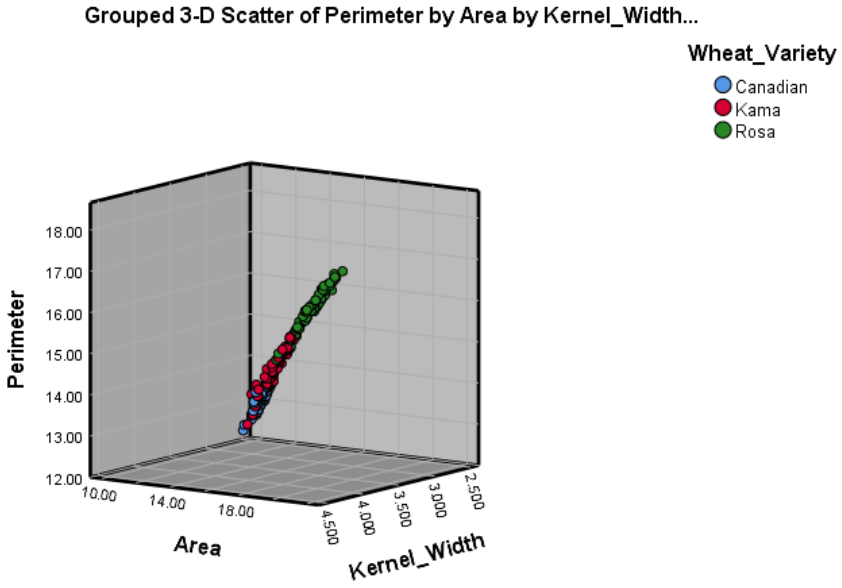
The three most important attributes for classifying the wheat data are the Area, Perimeter, and the Kernel Width.

Independent Variable Importance

Independent Variable	Importance	Normalized Importance
Area	.517	100.0%
Perimeter	.502	97.0%
Kernel_Width	.461	89.2%
Kernel_Length	.412	79.7%
Kernel_Groove_Length	.400	77.4%
Asymmetry_Coefficient	.236	45.7%
Compactness	.189	36.5%

Growing Method: CRT
Dependent Variable: Wheat_Variety

Lastly the three most important variables were visualized in 3D dimensional space to visualize the relationship between the variables mentioned above.



Above is a 3D scatter plot of the Area, Perimeter, and Kernel Width attributes. There is some overlap present for the Canadian and Kama wheat. Rosa appears to be fairly segregated from the other wheat varieties. There also appears to be a linear relationship between the wheat's perimeter and area.

Thank-you ☺