Izaak King

Professor Eleish

Data Science

September 19, 2025

"Lab 3"

Exercise 1

For this lab I trained and evaluated 2 kNN models using the abalone dataset. I used inputs of length, diameter, and height for the first model and inputs of whole weight, shucked weight, viscera weight, shell weight for the second model. Both models used age group as the label or predicted variable. Each model used a baseline k-value of 65, which is the square root of the sample size. For each model, a contingency table, the confusion matrix was created, allowing comparison of performance.

Model 1:

- Inputs: Length, Diameter, Height
- Label: Age Group
- K-value: 65

```
> print(confusionMatrix0)
          Actual
Predicted young adult  old
    young  1018   286   62
    adult   342  1335  625
    old      47   189  272
> print(paste("Amount Correct:", correct))
[1] "Amount Correct: 2625"
> print(paste("Accuracy:", round(accuracy, 3)))
[1] "Accuracy: 0.629"
```

Model 2:

- Inputs: Whole Weight, Shucked Weight, Viscera Weight, Shell Weight
- Label: Age Group
- K-value: 65

```
> print(confusionMatrix1)
          Actual
Predicted young adult  old
    young  1080   281   50
    adult   319  1409  535
    old       8   120  374
> print(paste("Amount Correct:", correct))
[1] "Amount Correct: 2863"
> print(paste("Accuracy:", round(accuracy, 3)))
[1] "Accuracy: 0.686"
```

Thus based on the accuracy of our models, we see that the second model, with inputs of Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight perform better. We will now attempt to find the optimal k-value for this model by using a range of k-values. This will use a for loop to try multiple k-values, calculating the accuracy of each value so we can determine which performs best.

Finding Optimal k-values

- Inputs: Whole Weight, Shucked Weight, Viscera Weight, Shell Weight
- Label: Age Group
- K-value Range: (45, 50, 55, 60, 65, 70, 75, 80, 85)

```
[1] "k-value: 45 Accuracy: 0.685344827586207"
[1] "k-value: 50 Accuracy: 0.685344827586207"
[1] "k-value: 55 Accuracy: 0.685823754789272"
[1] "k-value: 60 Accuracy: 0.685344827586207"
[1] "k-value: 65 Accuracy: 0.685344827586207"
[1] "k-value: 70 Accuracy: 0.687739463601533"
[1] "k-value: 75 Accuracy: 0.686302681992337"
[1] "k-value: 80 Accuracy: 0.685823754789272"
[1] "k-value: 85 Accuracy: 0.685344827586207"
> print(paste("Best performing k-value:", TopkVal))
[1] "Best performing k-value: 70"
```

Based on the accuracy of each model, we conclude the optimal k-value is 70.

Thus, our best performing model for classifying Age Group is trained with inputs of Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight, with a k-value of 70. This gives an accuracy of about 68.77%

Exercise 2

In this exercise, we will train K-Means and PAM models for this dataset, using the same feature subset of Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight. For both K-Means and PAM we will use a for loop to find the best performing model by trying a range of k-values (45, 50, 55, 60, 65, 70, 75, 80, 85), lot a silhouette of the model with the best k-value.

K-Means:

- Input: Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight
- K-values: (45, 50, 55, 60, 65, 70, 75, 80, 85)

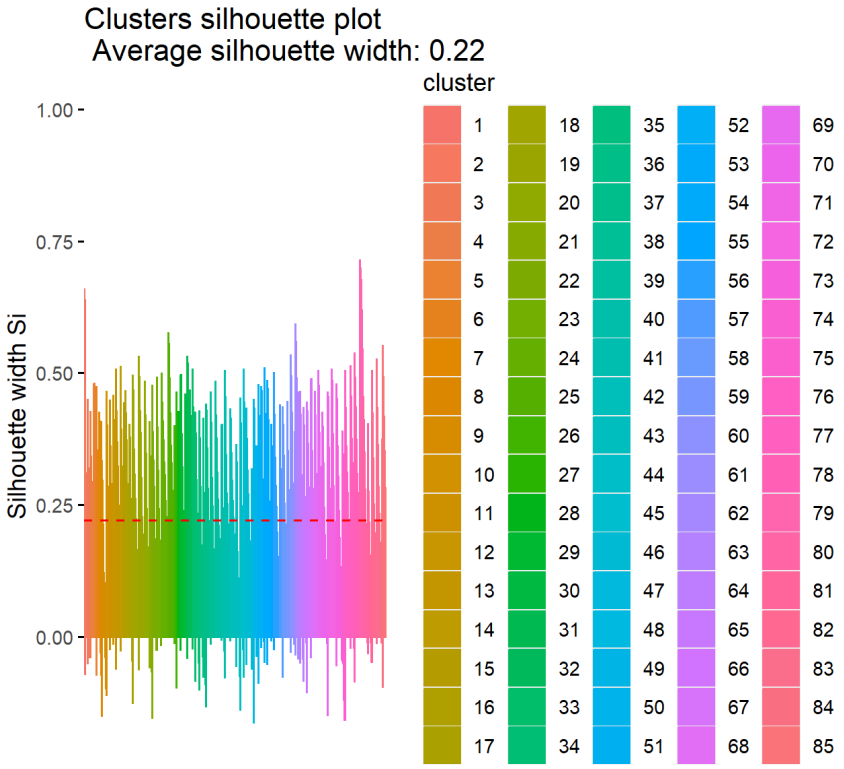
```
[1] "k = 45 WCSS = 425.281391024269 Avg SI = 0.247980522260524"
[1] "k = 50 WCSS = 418.441647972245 Avg SI = 0.233209955386962"
[1] "k = 55 WCSS = 416.372875106768 Avg SI = 0.23754798181949"
[1] "k = 60 WCSS = 366.608979318942 Avg SI = 0.233119778920223"
[1] "k = 65 WCSS = 416.077549026313 Avg SI = 0.220067789580063"
[1] "k = 70 WCSS = 353.465667933341 Avg SI = 0.227789972792494"
[1] "k = 75 WCSS = 322.105562128011 Avg SI = 0.225656550321667"
[1] "k = 80 WCSS = 350.370645501431 Avg SI = 0.230428510066466"
[1] "k = 85 WCSS = 300.657790965041 Avg SI = 0.231090242859395"
```

```
> print(paste("Best k by wcss:", best.k.wcss))
[1] "Best k by wcss: 85"
> print(paste("Best k by silhouette:", best.k.si))
[1] "Best k by silhouette: 50"
```

Based on this data, the model with k-value of 85 results in the smallest Within Cluster Sum of Squares (WCSS) value. Meanwhile k-value of 50 has the largest Silhouette Index (SI). Thus, both models perform the “best” based on each metric, so we will plot both.

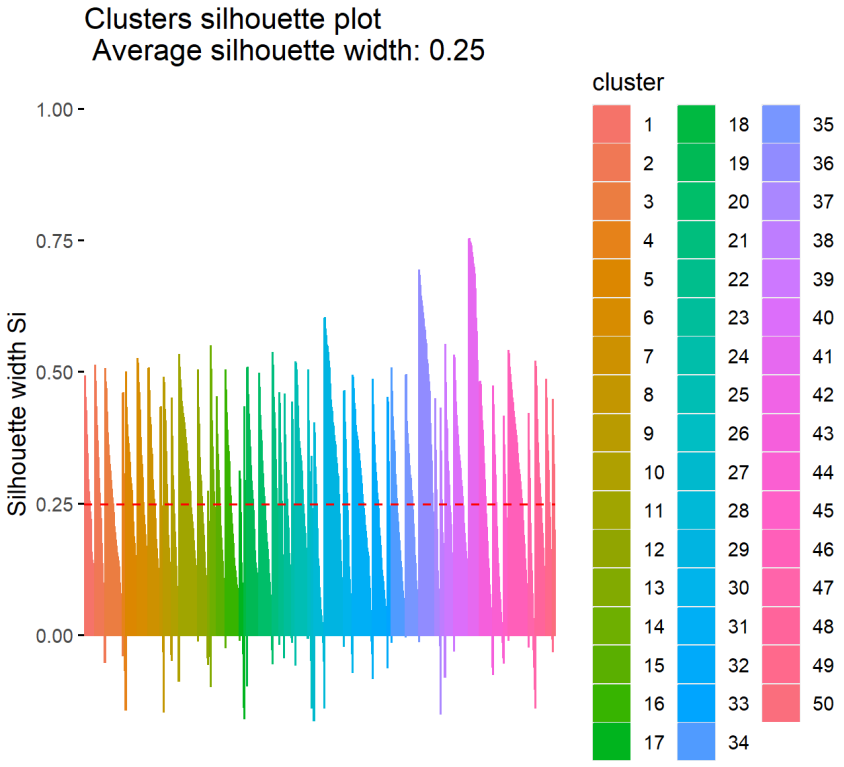
Model 1:

- Input: Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight
- K-value: 85



Model 2:

- Input: Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight
- K-value: 50



PAM Model:

My system struggled to create PAM Models based on the abalone data due to its size. I decided to use CLARA which performs better for clustering large applications. Both are k-medoid based clustering algorithms, with CLARA performing better for our large dataset

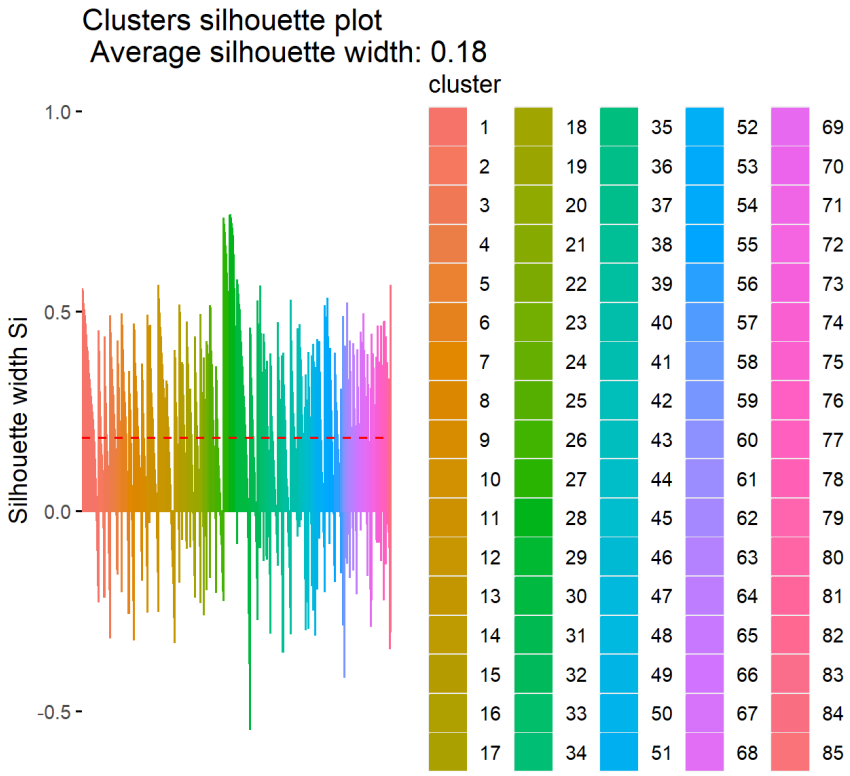
- Input: Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight
- K-values: (45, 50, 55, 60, 65, 70, 75, 80, 85)

```
[1] "k = 45 Sum Diss = 53896.1813034144 Avg SI = 0.201782278336981"
[1] "k = 50 Sum Diss = 47133.6645044488 Avg SI = 0.203128589382404"
[1] "k = 55 Sum Diss = 38618.631974705 Avg SI = 0.196564776127379"
[1] "k = 60 Sum Diss = 37892.4808341446 Avg SI = 0.204934762606081"
[1] "k = 65 Sum Diss = 31023.8325138089 Avg SI = 0.187271009478102"
[1] "k = 70 Sum Diss = 26507.7734400515 Avg SI = 0.190075812942892"
[1] "k = 75 Sum Diss = 25371.6804196645 Avg SI = 0.18210821799371"
[1] "k = 80 Sum Diss = 24458.3342655804 Avg SI = 0.194191365196158"
[1] "k = 85 Sum Diss = 22126.2563684771 Avg SI = 0.184609596380461"
> print(paste("Best k by sum of dissimilarities:", best.k.sumdiss))
[1] "Best k by sum of dissimilarities: 85"
> print(paste("Best k by silhouette:", best.k.si))
[1] "Best k by silhouette: 60"
```

Based on this data, the model with k-value of 85 results in the smallest sum of similarities value. Meanwhile k-value of 60 has the largest Silhouette Index (SI). Thus, both models perform the “best” based on each metric, so we will plot both.

Model 1:

- Input: Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight
- K-value: 85



Model 2:

- Input: Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight
- K-value: 60