assignment_11_02_PothineniKalyan

PothineniKalyan

2023-05-27

Introduction to machine Learning

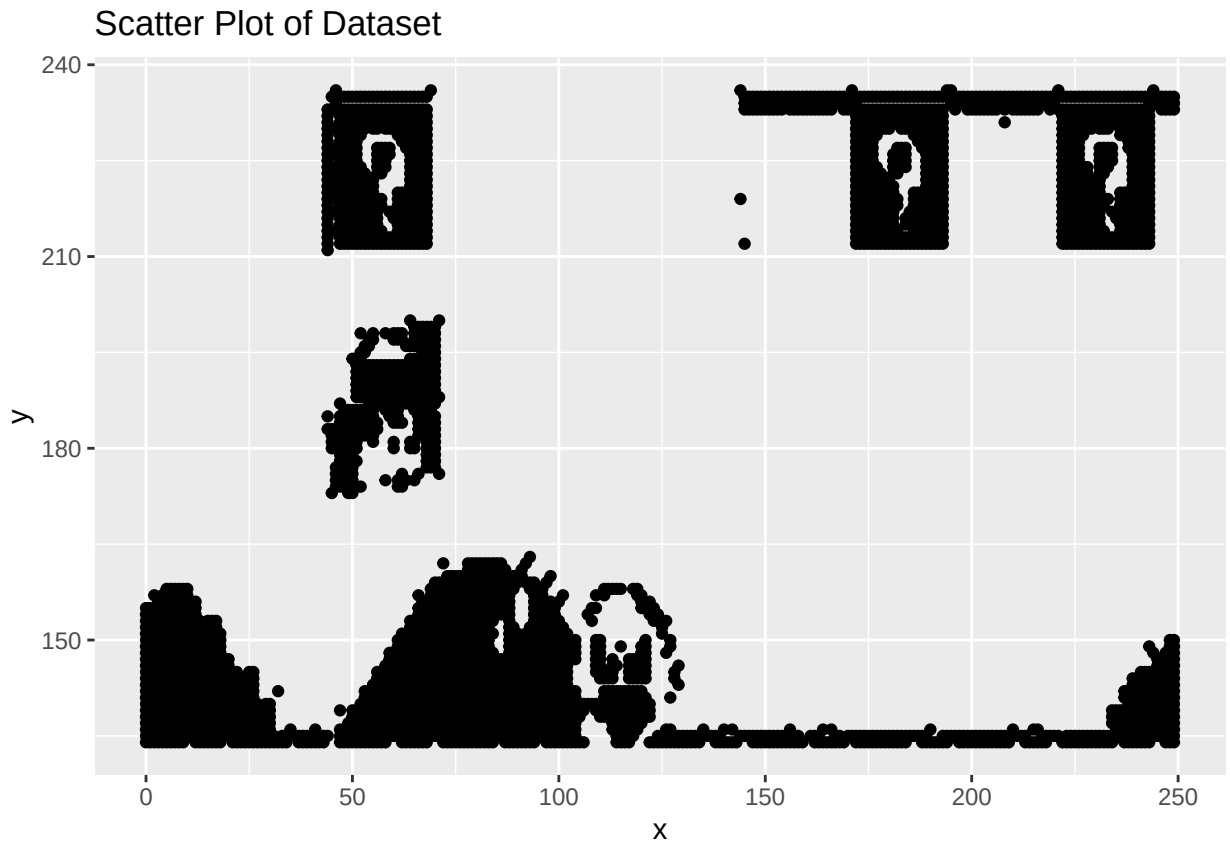
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(cluster)
library(pathviewr)
```

```
## Warning: package 'pathviewr' was built under R version 4.2.3
```

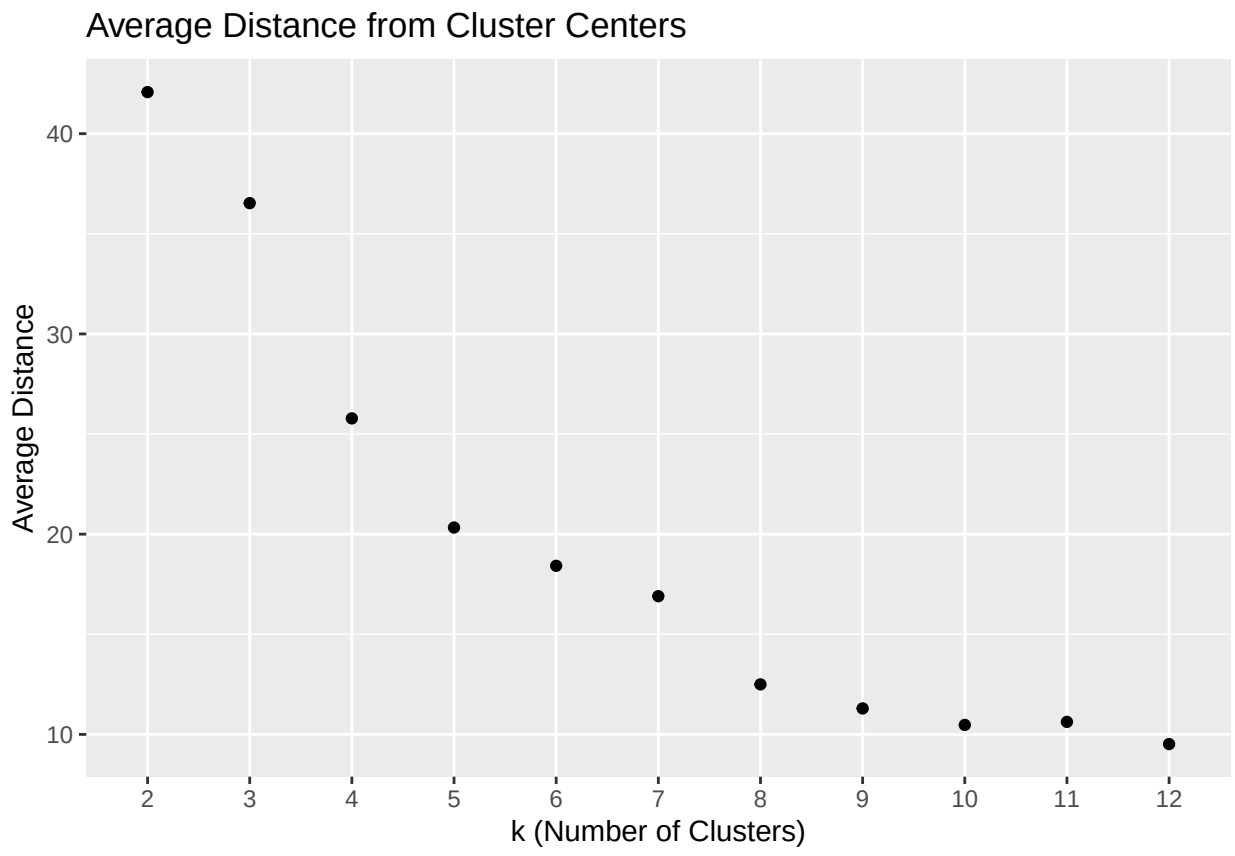
```
# Load the clustering-data.csv dataset
dataset <- read.csv("data/clustering-data.csv")

# Plot the clustering-data.csv dataset
ggplot(dataset, aes(x = x, y = y)) +
  geom_point() +
  labs(title = "Scatter Plot of Dataset")
```




```r
# Fit the dataset using k-means
k_values <- 2:12
distances <- numeric(length(k_values))

for (i in seq_along(k_values)) {
  k <- k_values[i]
  kmeans_model <- kmeans(dataset, centers = k)
  cluster_assignments <- kmeans_model$cluster
  cluster_centers <- kmeans_model$centers

  # Calculate average distance from cluster centers
  total_distance <- 0
  for (j in 1:nrow(dataset)) {
    data_point <- dataset[j, ]
    cluster_center <- cluster_centers[cluster_assignments[j], ]
    distance <- sqrt(sum((data_point - cluster_center)^2))
    total_distance <- total_distance + distance
  }
  avg_distance <- total_distance / nrow(dataset)

  distances[i] <- avg_distance

  ggplot(dataset, aes(x = x, y = y, color = factor(cluster_assignments))) +
    geom_point() +
    geom_point(data = as.data.frame(kmeans_model$centers), aes(x = x, y = y),
               shape = 4, size = 4) +
```

```
    labs(title = paste0("k-Means Clustering (k = ", k, ")")) +
    theme(legend.position = "none")
}

# Create line chart
distance_df <- data.frame(k = k_values, distance = distances)
ggplot(distance_df, aes(x = factor(k), y = distance )) +
  geom_line() +
  geom_point() +
  labs(title = "Average Distance from Cluster Centers",
       x = "k (Number of Clusters)",
       y = "Average Distance",
  scale_x_continuous(breaks = as.character(k)))
```

```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```




```
print(distance_df)
```

```
##    k  distance
## 1  2 42.077169
## 2  3 36.531930
## 3  4 25.782290
## 4  5 20.331782
```

```
## 5   6 18.420531
## 6   7 16.902325
## 7   8 12.498706
## 8   9 11.297054
## 9  10 10.474251
## 10 11 10.626290
## 11 12  9.517825
```

Looking at the graph it is hard to find the correct "elbow point", we can leverage pathviewr library to find the elbow curve

```
elbow_curve <- find_curve_elbow(distance_df, export_type = "row_num",
                                plot_curve = FALSE)

# Print the result
print(paste("Elbow point for the dataset: ", elbow_curve))
```

```
## [1] "Elbow point for the dataset:  4"
```