# Cordero_week10.2

Joaquin Cordero

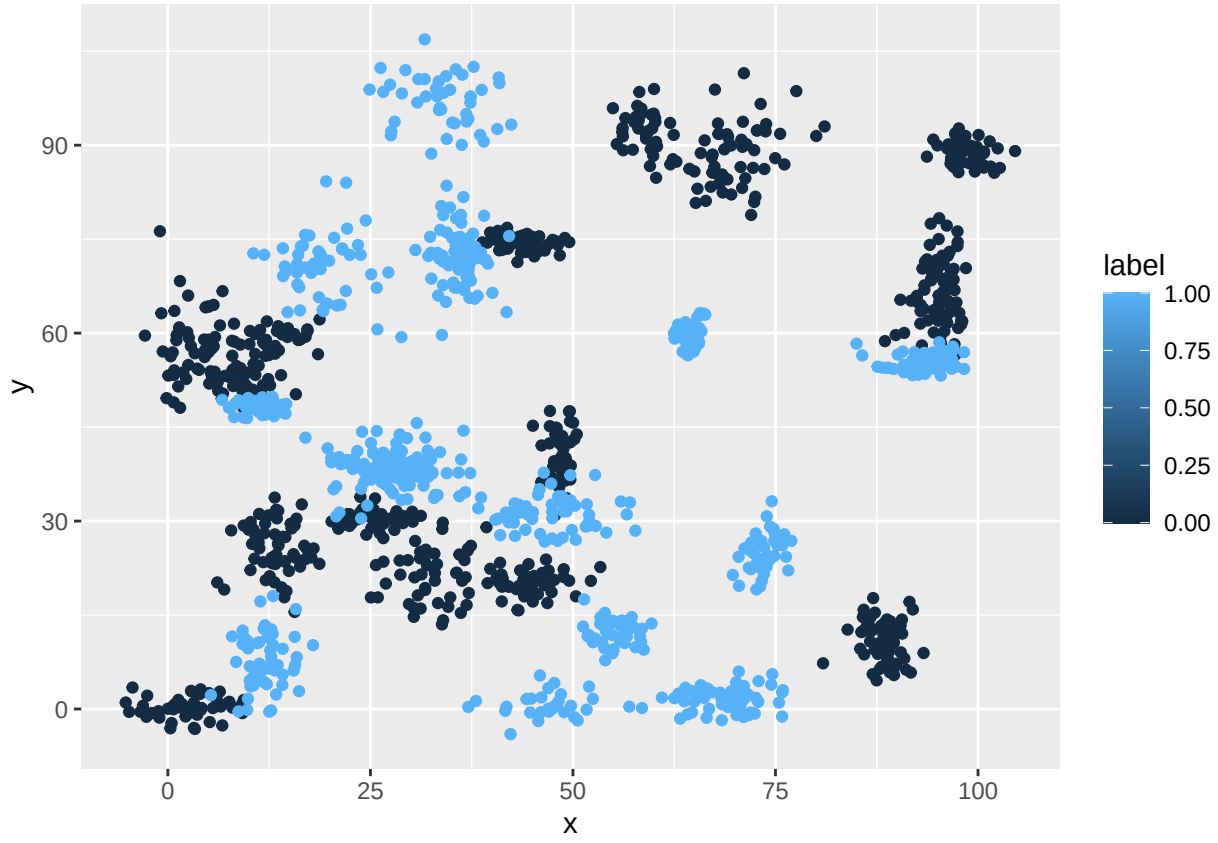
2024-08-5

```
## Loading required package: lattice
```

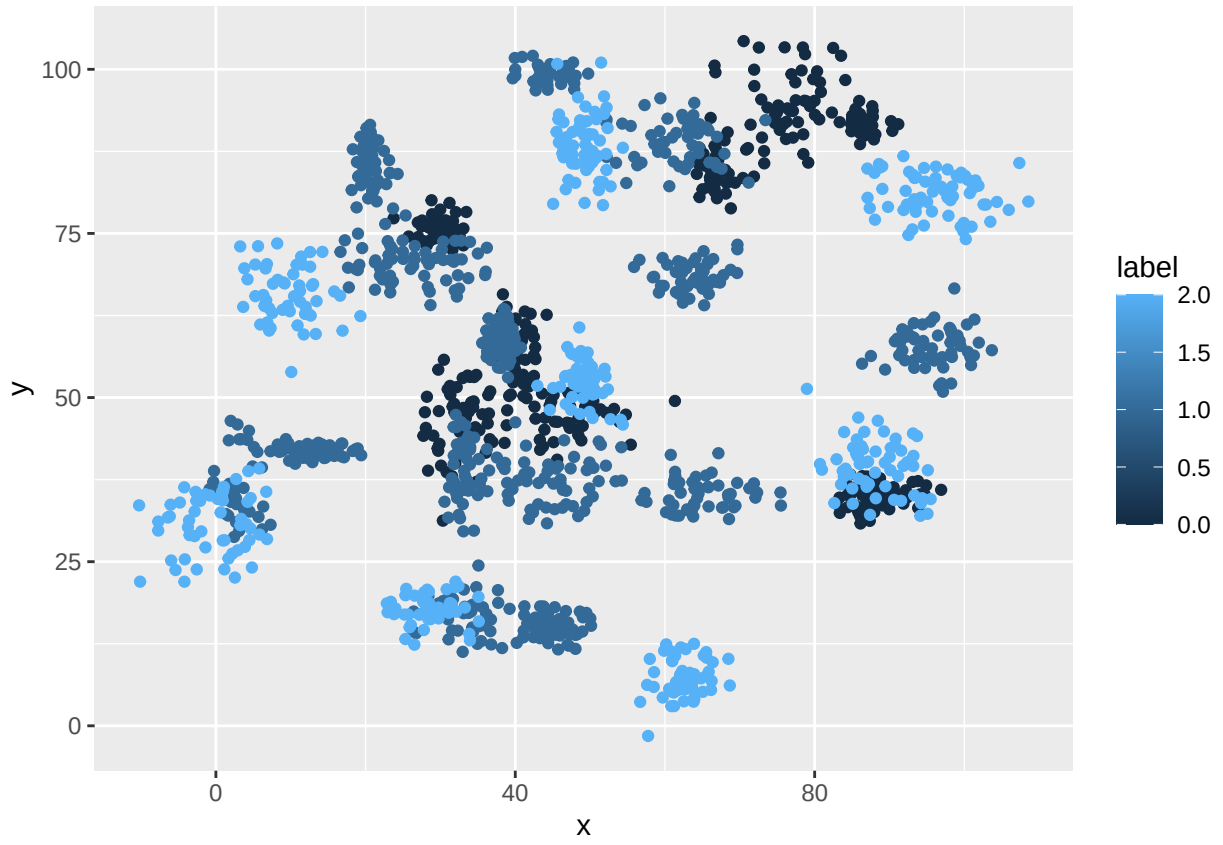
```
#Binary Classifier data
bc_data <- read.csv("binary-classifier-data.csv")
#Trinary Classifier data
tc_data <- read.csv("trinary-classifier-data.csv")

clustering_data <- read.csv("clustering-data.csv")
```

### 1.e.i

```
ggplot(bc_data, aes(x = x, y = y, color = label)) + geom_point()
```



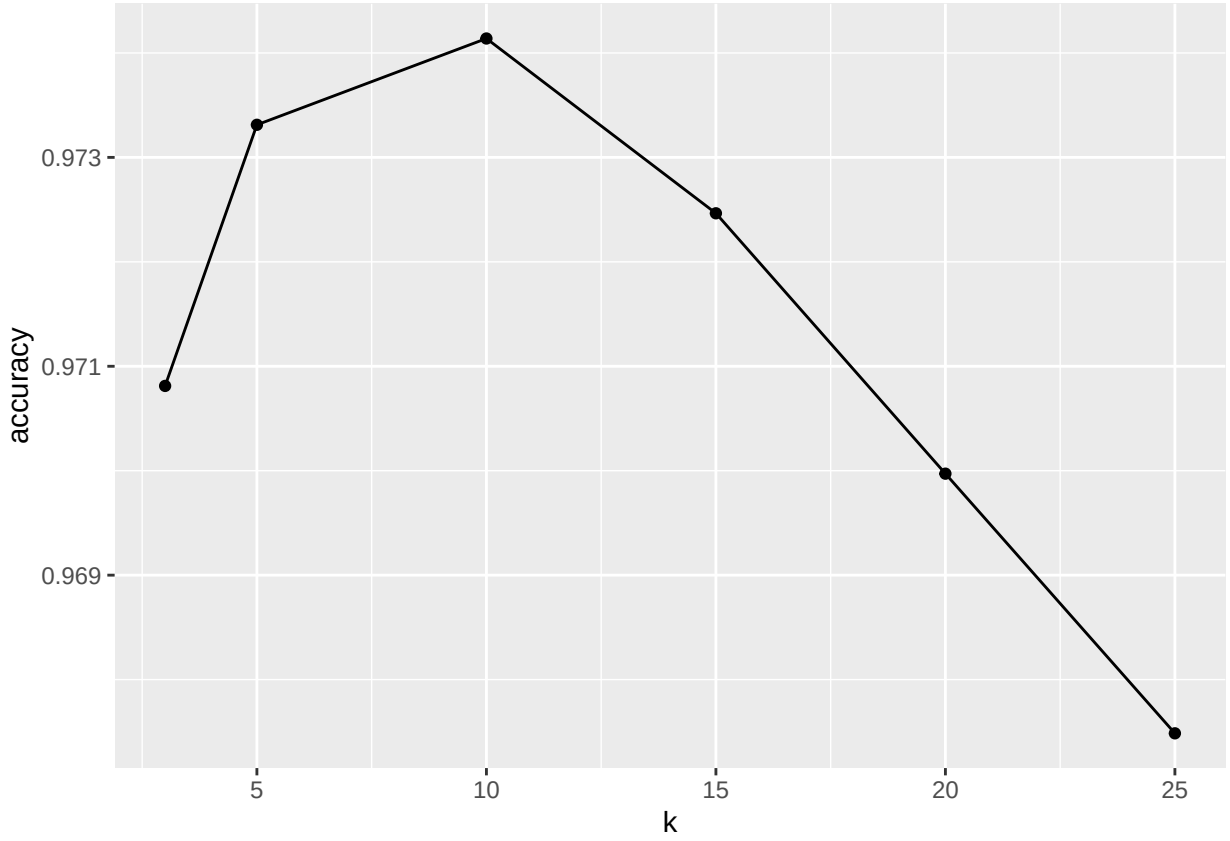
```
ggplot(tc_data, aes(x = x, y = y, color = label)) + geom_point()
```




### 1.e.ii

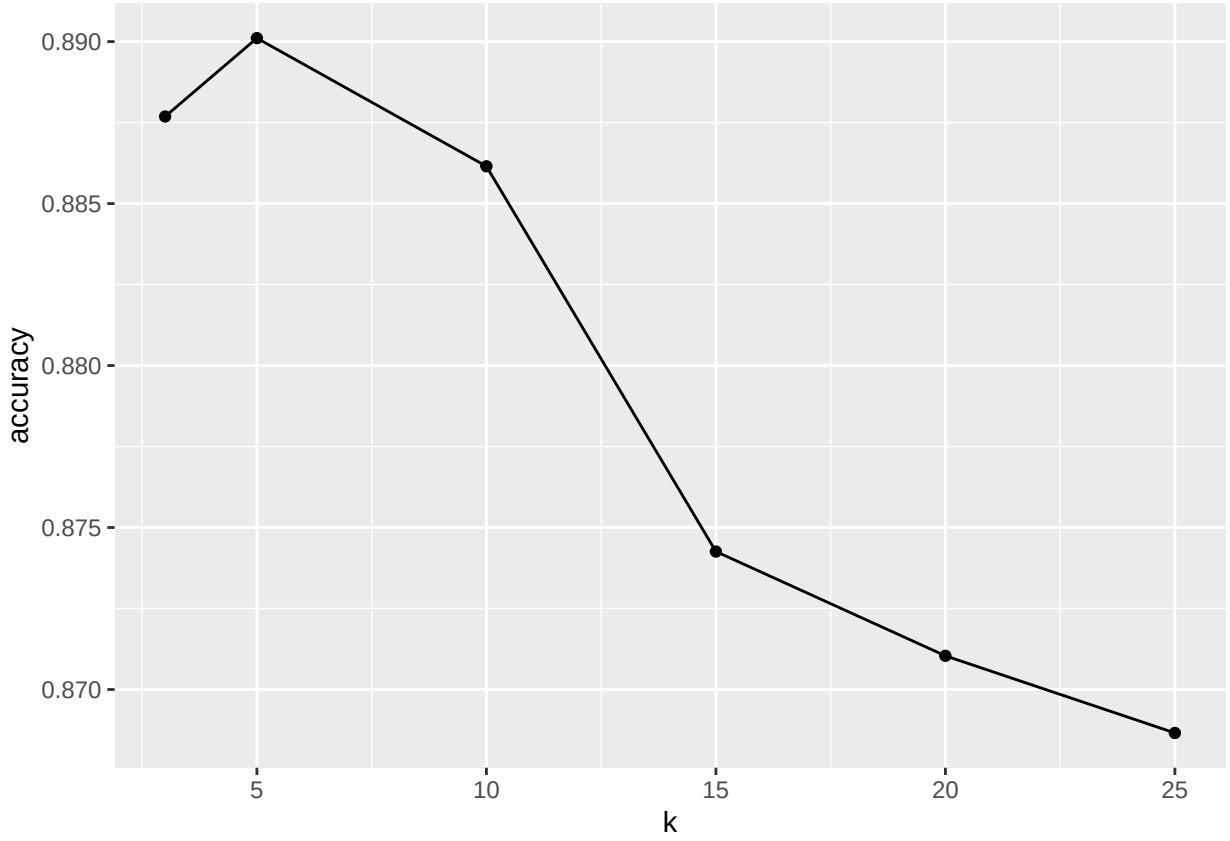
```
set.seed(123)

bc_data$label <- factor(bc_data$label, levels = c(0, 1))

bc_data_split <- createDataPartition(bc_data$label, times = 1, p = .8, list = FALSE)

bc_train <- bc_data[bc_data_split, ]
bc_test <- bc_data[-bc_data_split, ]

bc_preproc <- preProcess(bc_train, method = c("center", "scale"))
bc_train_trans <- predict(bc_preproc, bc_train)
bc_test_trans <- predict(bc_preproc, bc_test)

bc_knn_model <- train(
  label ~ .,
  data = bc_train_trans,
  method = "knn",
  trControl = trainControl(method = "cv"),
  tuneGrid = data.frame(k = c(3, 5, 10, 15, 20, 25))
```

```
)

bc_result <- bc_knn_model$results

bc_result_df <- data.frame(k = bc_result$k, accuracy = bc_result$Accuracy)

ggplot(bc_result_df, aes(x = k, y = accuracy)) + geom_line() + geom_point()
```




```
set.seed(234)

tc_data$label <- as.factor(tc_data$label)

tc_data_split <- createDataPartition(tc_data$label, times = 1, p = .8, list = FALSE)

tc_train <- tc_data[tc_data_split, ]
tc_test <- tc_data[-tc_data_split, ]

tc_preproc <- preProcess(tc_train, method = c("center", "scale"))
tc_train_trans <- predict(tc_preproc, tc_train)
tc_test_trans <- predict(tc_preproc, tc_test)

tc_knn_model <- train(
  label ~ .,
  data = tc_train_trans,
  method = "knn",
```

```
  trControl = trainControl(method = "cv"),
  tuneGrid = data.frame(k = c(3, 5, 10, 15, 20, 25))
)

tc_result <- tc_knn_model$results

tc_result_df <- data.frame(k = tc_result$k, accuracy = tc_result$Accuracy)

ggplot(tc_result_df, aes(x = k, y = accuracy)) + geom_line() + geom_point()
```

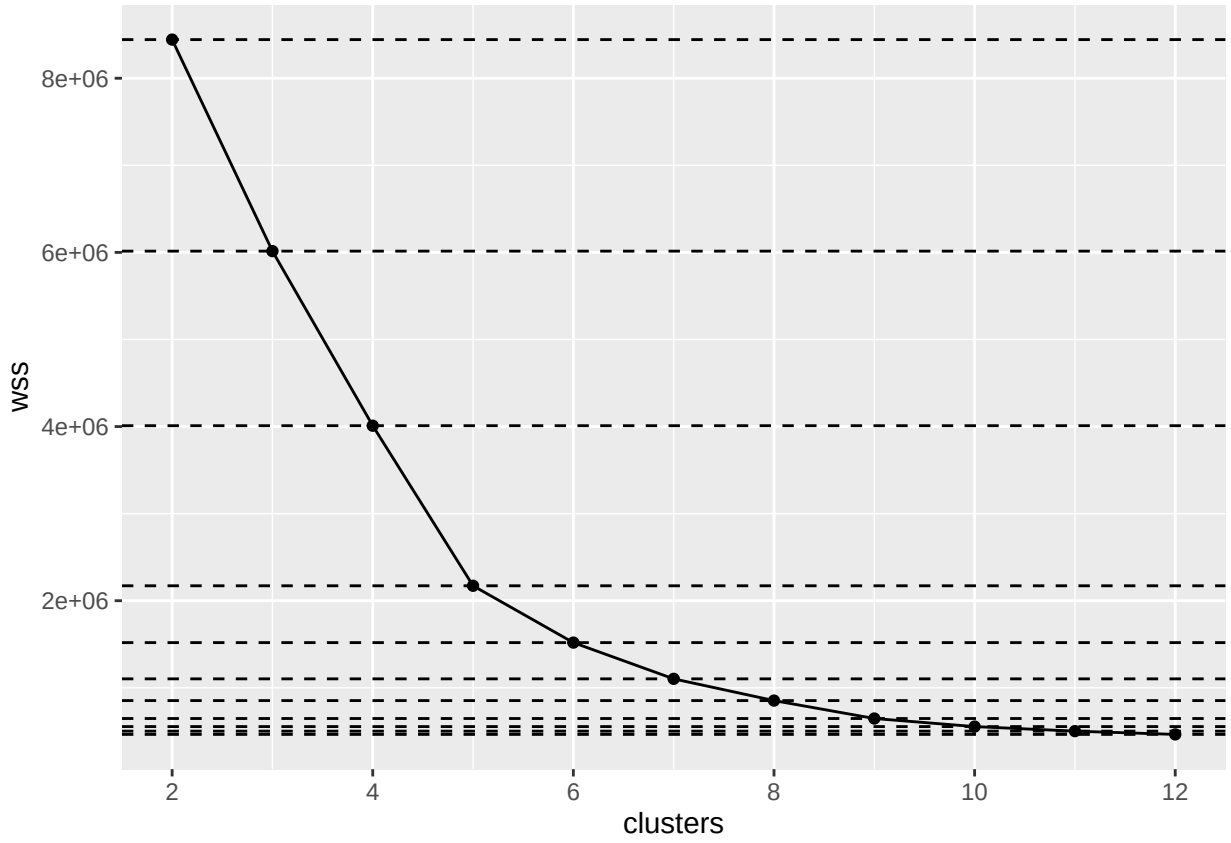



i. A linear classifier might not work well for the bc_data as the results form a curve. A linear classifier might not work as well for the tc_data as the results form a curved pattern.

ii. The accuracy is much better overall, the difference is using a different type of machine learning model.

```
clustering_k_count <- c(2,3,4,5,6,7,8,9,10,11,12)

clustering_wss <- numeric(length(clustering_k_count))

set.seed(345)

for (i in seq_along(clustering_k_count)) {
  km_clustering <- kmeans(clustering_data, center = clustering_k_count[i], nstart = 20)
  clustering_wss[i] <- km_clustering$tot.withinss
}
```

```r
clustering_wss_df <- tibble(clusters = unlist(clustering_k_count), wss = clustering_wss)

clustering_scree_plot <- ggplot(clustering_wss_df, aes(x = clusters, y = wss, group = 1)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(breaks = c(2, 4, 6, 8, 10, 12))

clustering_scree_plot +
  geom_hline(yintercept = clustering_wss, linetype = 'dashed')
```




The elbow point for this dataset is 5.