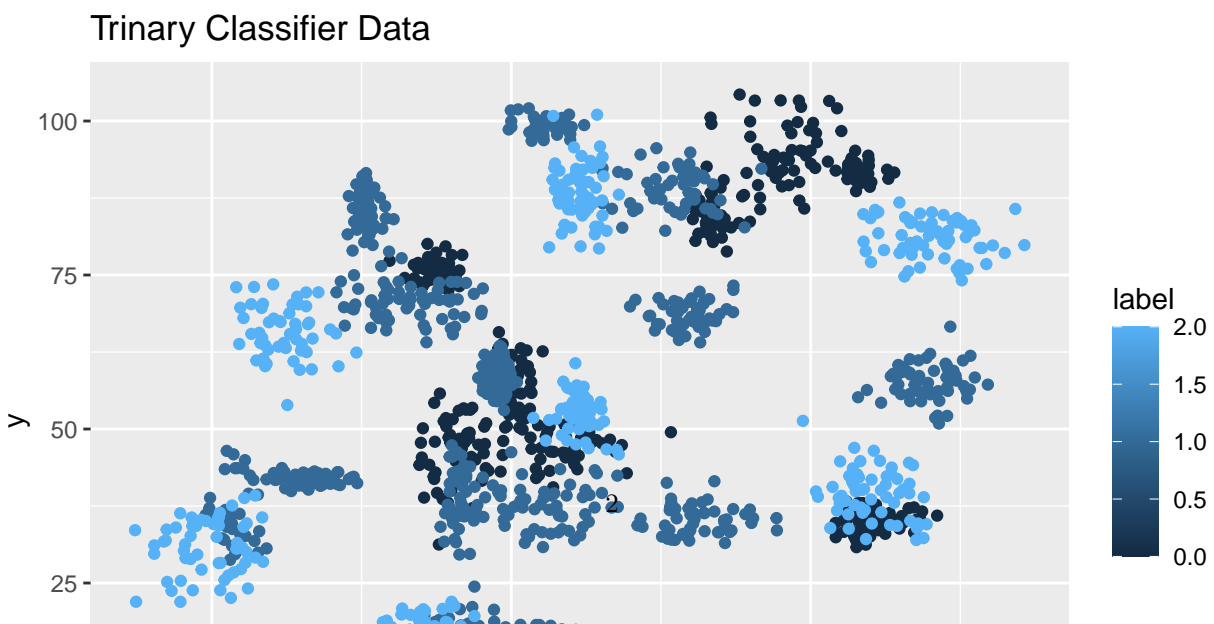
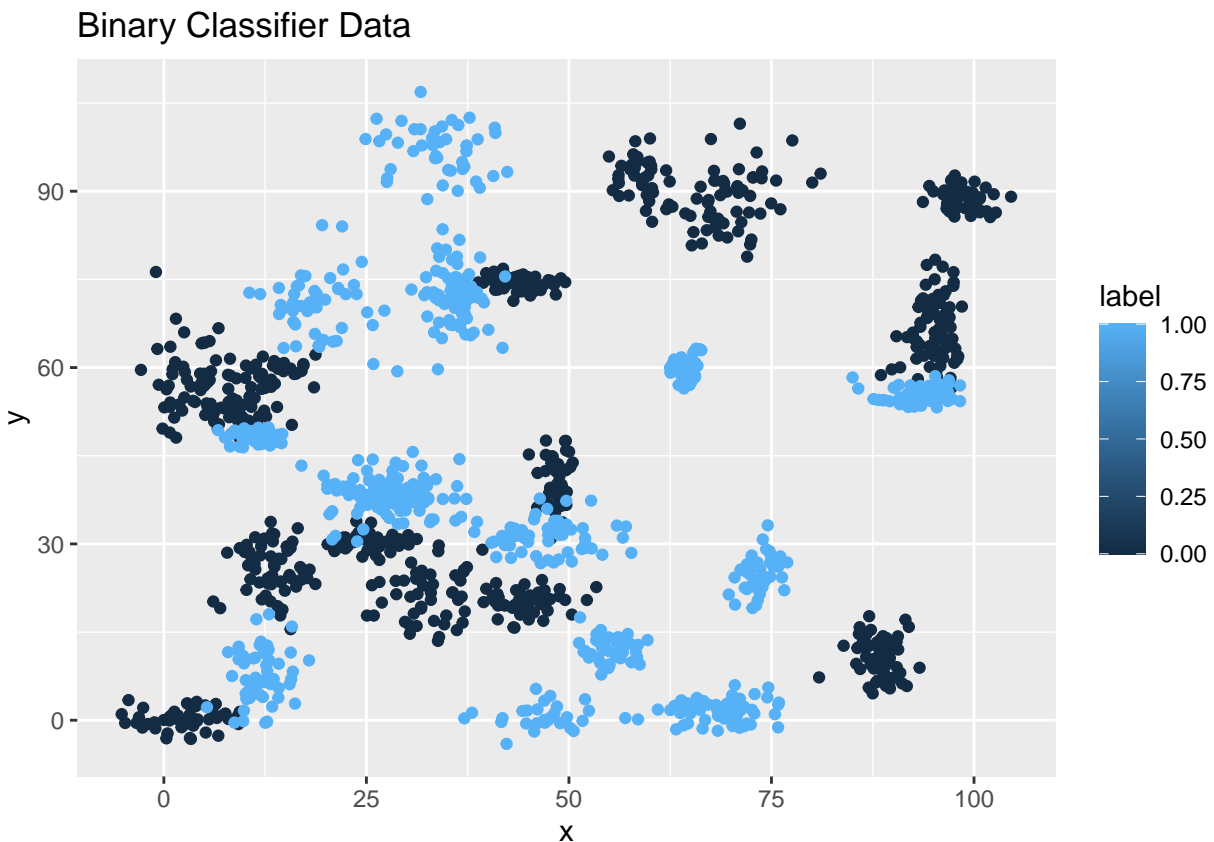

title: "Week 9: Assignment 15" author: "Megan Holford" date: "10/31/2020" output: pdf_document:
default html_document: default word_document: default

In this problem, you will use the nearest neighbors algorithm to fit a model on two simplified datasets. The first dataset (found in `binary-classifier-data.csv`) contains three variables; label, x, and y. The label variable is either 0 or 1 and is the output we want to predict using the x and y variables. The second dataset (found in `trinary-classifier-data.csv`) is similar to the first dataset except that the label variable can be 0, 1, or 2.

a. Plot the data from each dataset using a scatter plot.



b. Fit a k nearest neighbors model for each dataset for k=3, k=5, k=10, k=15, k=20, and k=25. Compute the accuracy of the resulting models for each value of k. Plot the results in a graph where the x-axis is the different values of k and the y-axis is the accuracy of the model.

```
library(caTools)

binary_data <- sample.split(binary_df$label, SplitRatio=0.8)
binary_train <- subset(binary_df, binary_data==TRUE)
binary_test <- subset(binary_df, binary_data==FALSE)

k_values<- 1:25

Accuracy <- NULL

for(i in k_values){
  test_pred <- knn(binary_train[2:3],binary_test[2:3],binary_train$label,k=k_values[i])
  confmatrix <- table(binary_test$label,test_pred)
  accuracy <- (confmatrix[[1,1]] + confmatrix[[2,2]]) / sum(confmatrix)
  Accuracy <- c(Accuracy, round((accuracy * 100), digits=2))
}

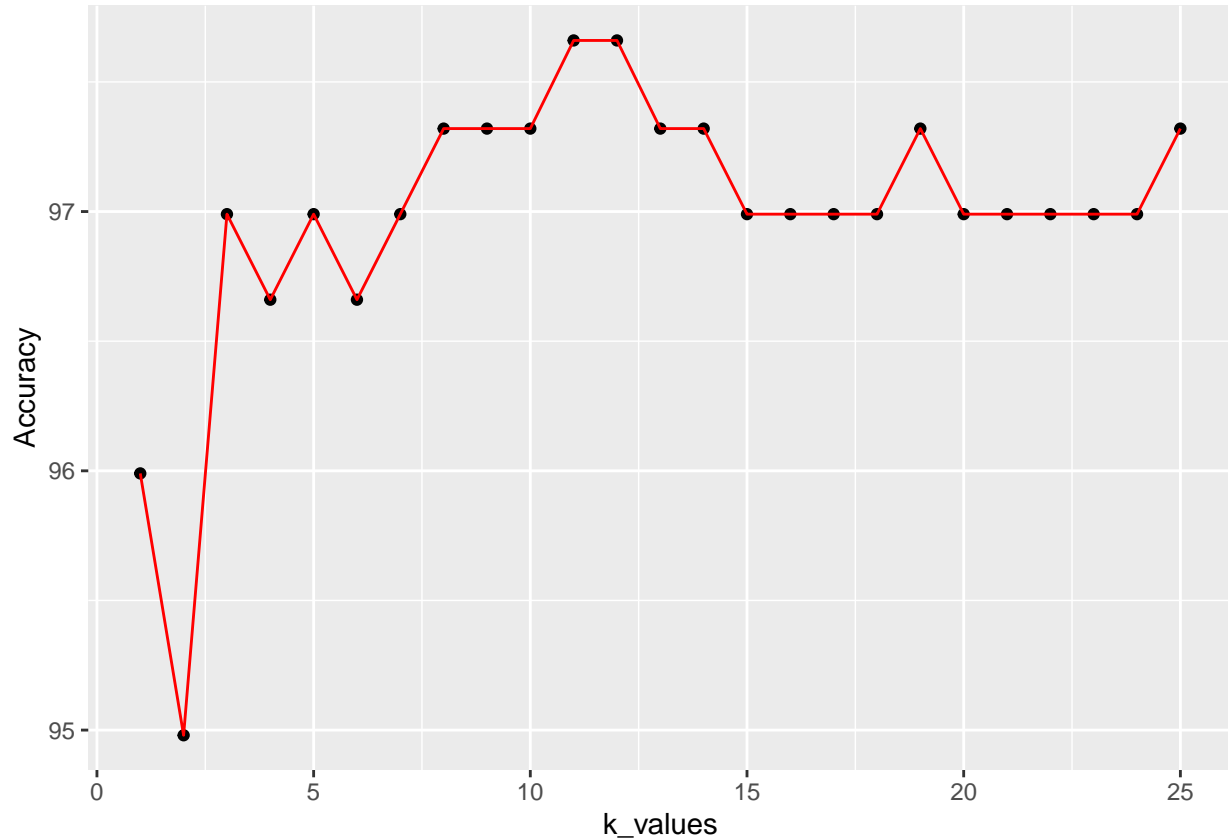
binary_results_df <- data.frame(k_values, Accuracy)

binary_results_df
```

```
##      k_values Accuracy
## 1           1    95.99
## 2           2    94.98
## 3           3    96.99
## 4           4    96.66
## 5           5    96.99
## 6           6    96.66
## 7           7    96.99
## 8           8    97.32
## 9           9    97.32
## 10          10    97.32
## 11          11    97.66
## 12          12    97.66
## 13          13    97.32
## 14          14    97.32
## 15          15    96.99
## 16          16    96.99
## 17          17    96.99
## 18          18    96.99
## 19          19    97.32
## 20          20    96.99
## 21          21    96.99
## 22          22    96.99
## 23          23    96.99
```

```
## 24      24      96.99
## 25      25      97.32
```

```
ggplot(binary_results_df, aes(x=k_values, y=Accuracy)) + geom_point() + geom_line(colour="red")
```



Trinary dataset:

```
trinary_data <- sample.split(trinary_df$label, SplitRatio=0.8)
trinary_train <- subset(trinary_df, trinary_data==TRUE)
trinary_test <- subset(trinary_df, trinary_data==FALSE)

k_values<- 1:25

Accuracy <- NULL

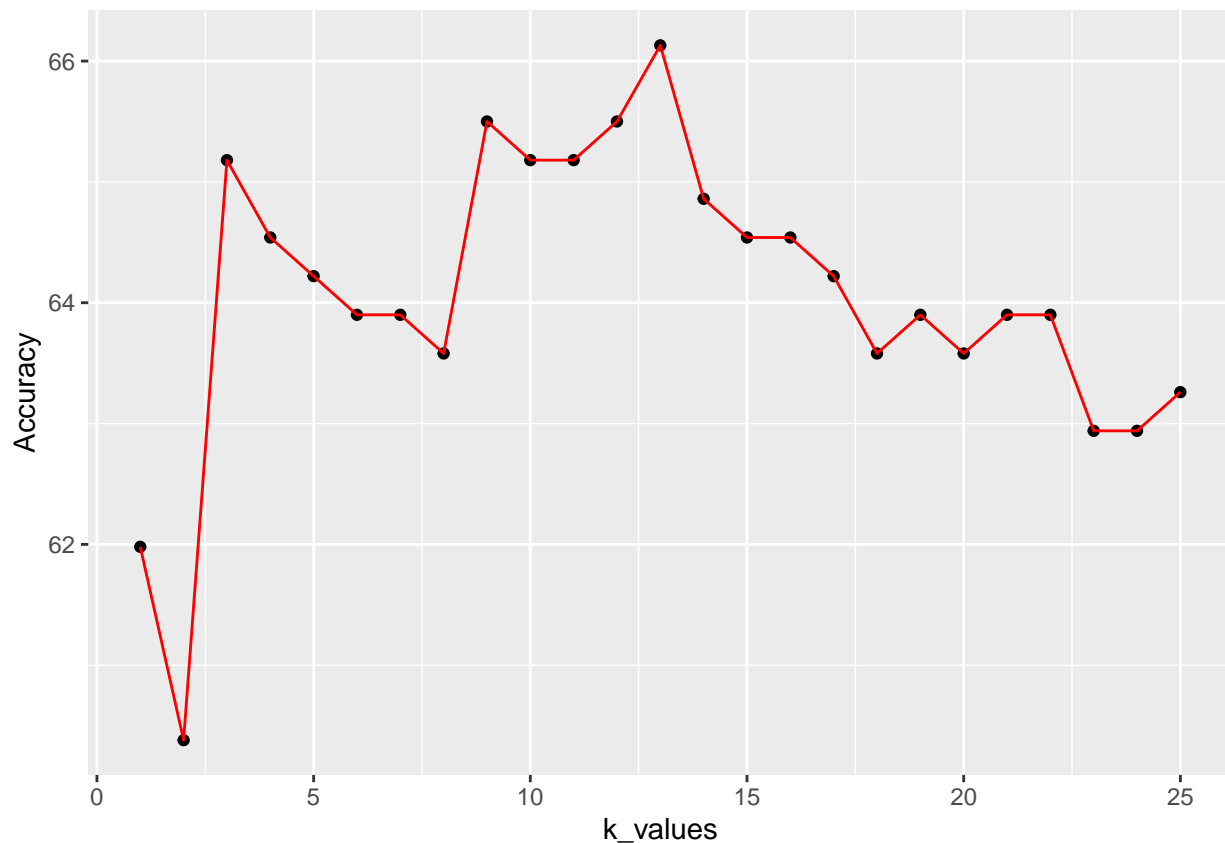
for (i in k_values){
  test_pred <- knn(trinary_train[2:3],trinary_test[2:3],trinary_train$label,k=k_values[i])
  confmatrix <- table(trinary_test$label,test_pred)
  accuracy <- (confmatrix[[1,1]] + confmatrix[[2,2]]) / sum(confmatrix)
  Accuracy <- c(Accuracy, round((accuracy * 100), digits=2))
}

trinary_results_df <- data.frame(k_values, Accuracy)
```

```
trinary_results_df
```

```
##      k_values Accuracy
## 1           1    61.98
## 2           2    60.38
## 3           3    65.18
## 4           4    64.54
## 5           5    64.22
## 6           6    63.90
## 7           7    63.90
## 8           8    63.58
## 9           9    65.50
## 10          10    65.18
## 11          11    65.18
## 12          12    65.50
## 13          13    66.13
## 14          14    64.86
## 15          15    64.54
## 16          16    64.54
## 17          17    64.22
## 18          18    63.58
## 19          19    63.90
## 20          20    63.58
## 21          21    63.90
## 22          22    63.90
## 23          23    62.94
## 24          24    62.94
## 25          25    63.26
```

```
ggplot(trinary_results_df, aes(x=k_values, y=Accuracy)) + geom_point() + geom_line(colour="red")
```



c. In later lessons, you will learn about linear classifiers. These algorithms work by defining a decision boundary that separates the different categories. Looking back at the plots of the data, do you think a linear classifier would work well on these datasets?

The data in both the binary and trinary data sets are very complex and all over. Because of this, linear classifiers would not work as they will not define proper boundaries and categories of the data since it is so mixed.