Exercise 14: Fit a Logistic Regression Model to Previous Dataset

Michael Hotaling

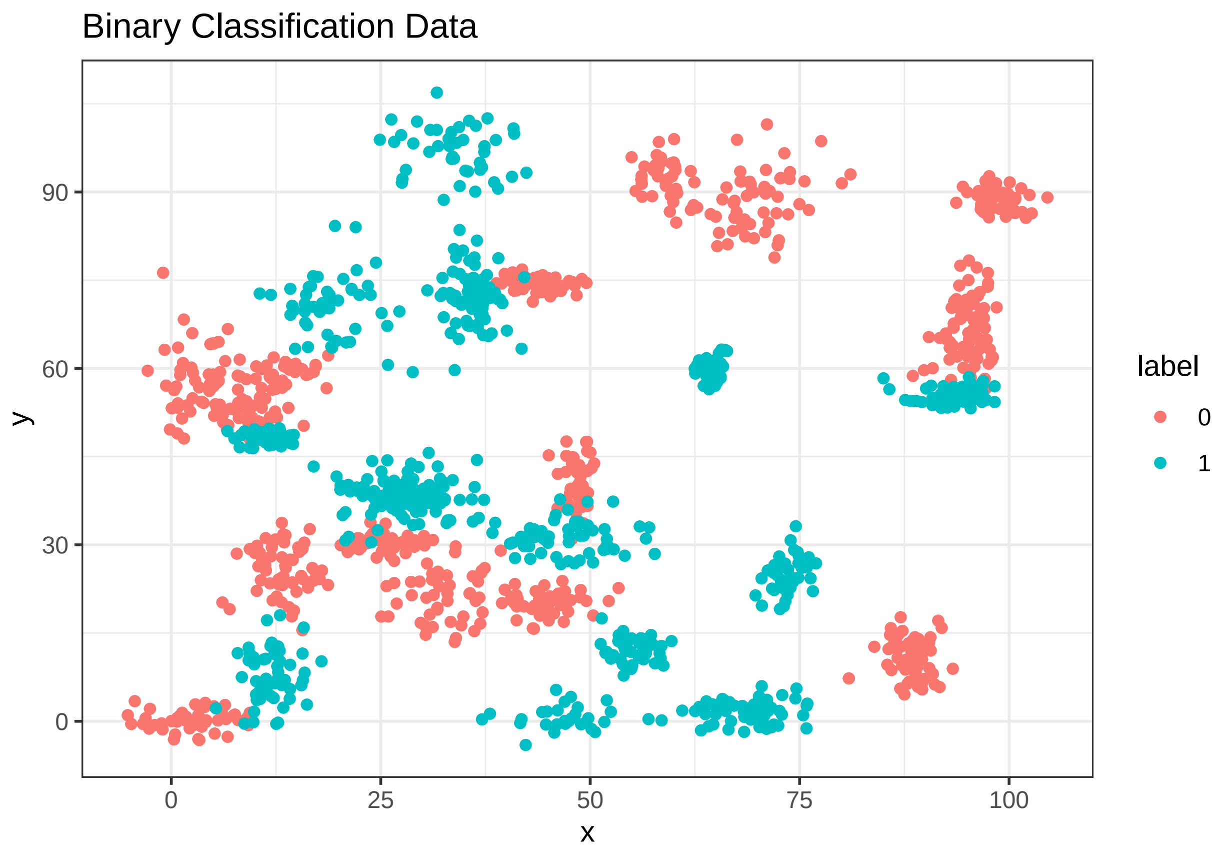
10/20/2020

Exercise 14: Fit a Logistic Regression model to Previous Dataset

```r
library(caTools)
library(ggplot2)

df <- read.csv("binary-classifier-data.csv")
df$label <- as.factor(df$label)

ggplot(data = df, aes(x = x, y = y, color = label)) + geom_point() +
ggtitle("Binary Classification Data") +
theme_bw()
```



a. Fit a logistic regression model to the binary-classifier-data.csv dataset from the previous assignment.

```r
library(caTools)


df <- read.csv("binary-classifier-data.csv")

set.seed(520)

sample <- sample.split(df$label, SplitRatio = 0.70)

train = subset(df, sample == TRUE)
test = subset(df, sample == FALSE)

glm.model = glm(label ~ . , family = binomial(logit), data = train)

summary(glm.model)
```

```
##
## Call:
## glm(formula = label ~ ., family = binomial(logit), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4193  -1.1758  -0.9198   1.1550   1.4110
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.526165   0.140903   3.734 0.000188 ***
## x           -0.003615   0.002201  -1.643 0.100444
## y           -0.009011   0.002232  -4.037 5.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1453.6  on 1048  degrees of freedom
## Residual deviance: 1430.4  on 1046  degrees of freedom
## AIC: 1436.4
##
## Number of Fisher Scoring iterations: 4
```

a. What is the accuracy of the logistic regression classifier?

```r
library(knitr)
test$predicted = predict(glm.model, newdata=test, type="response")
kable(table(test$label, test$predicted> 0.5))
```

	FALSE	TRUE
0	104	126
1	86	133

Really bad. It isn't surprising since we can't split our data into two sections with a line. If we had more attributes, we might be able to make better predictions.

b. How does the accuracy of the logistic regression classifier compare to the nearest neighbors algorithm?

```
library(class)

knn.model <- knn(train[2:3], test[2:3], train$label, k = 1)

mean(test$label != knn.model)
```

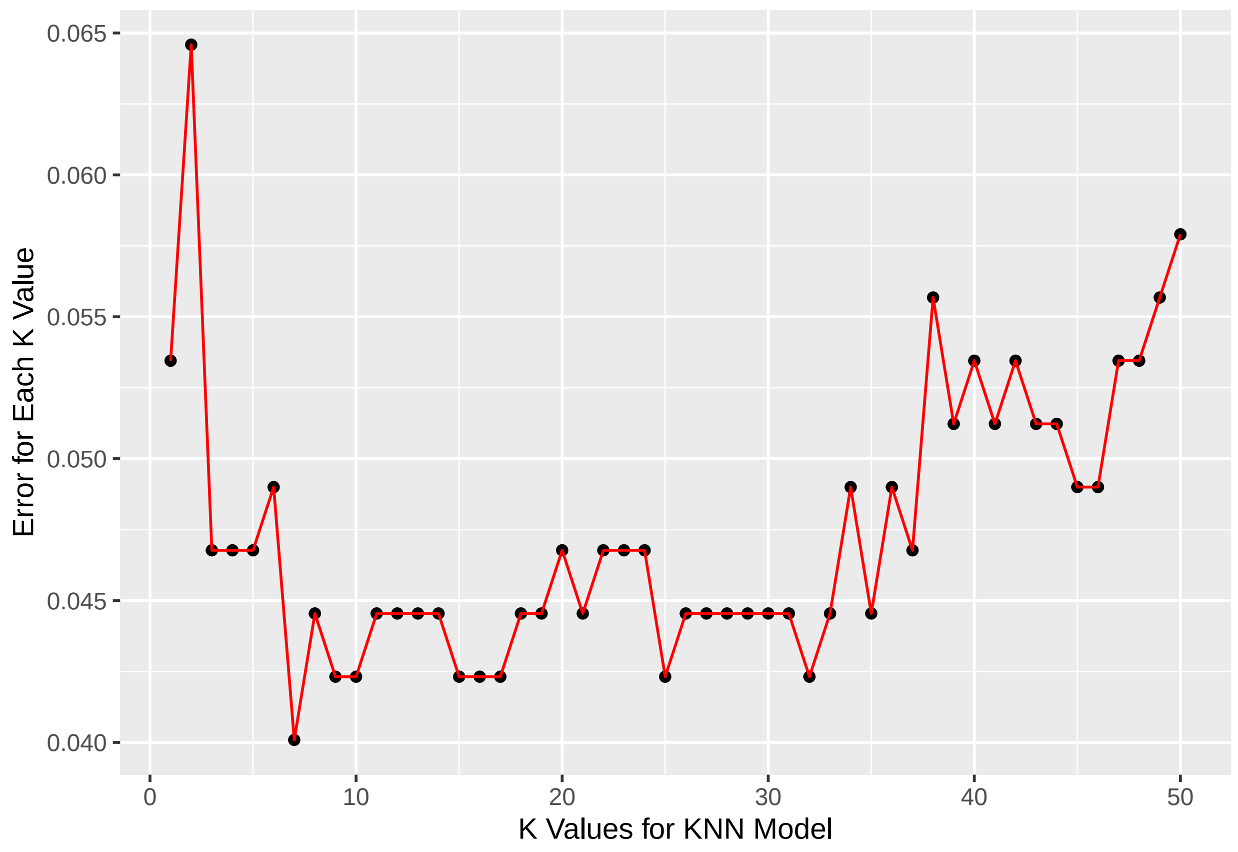
```
## [1] 0.05345212
```

```
predicted.values <- NULL
error.rate <- NULL

for(i in 1:50){
    set.seed(520)
    predicted.values <- knn(train[2:3],test[2:3],train$label,k=i)
    error.rate[i] <- mean(test$label != predicted.values)
}

k.values <- 1:50
error.df <- data.frame(error.rate,k.values)

ggplot(error.df,aes(x=k.values,y=error.rate)) +
geom_point() +
geom_line(color='red') +
xlab("K Values for KNN Model") +
ylab("Error for Each K Value")
```



Our KNN model is much better at predicting which categories our data falls into. With an accuracy of 95%, we can say that this model is much more robust than our logistic model. We can increase our accuracy even more by using k=7 in our model.

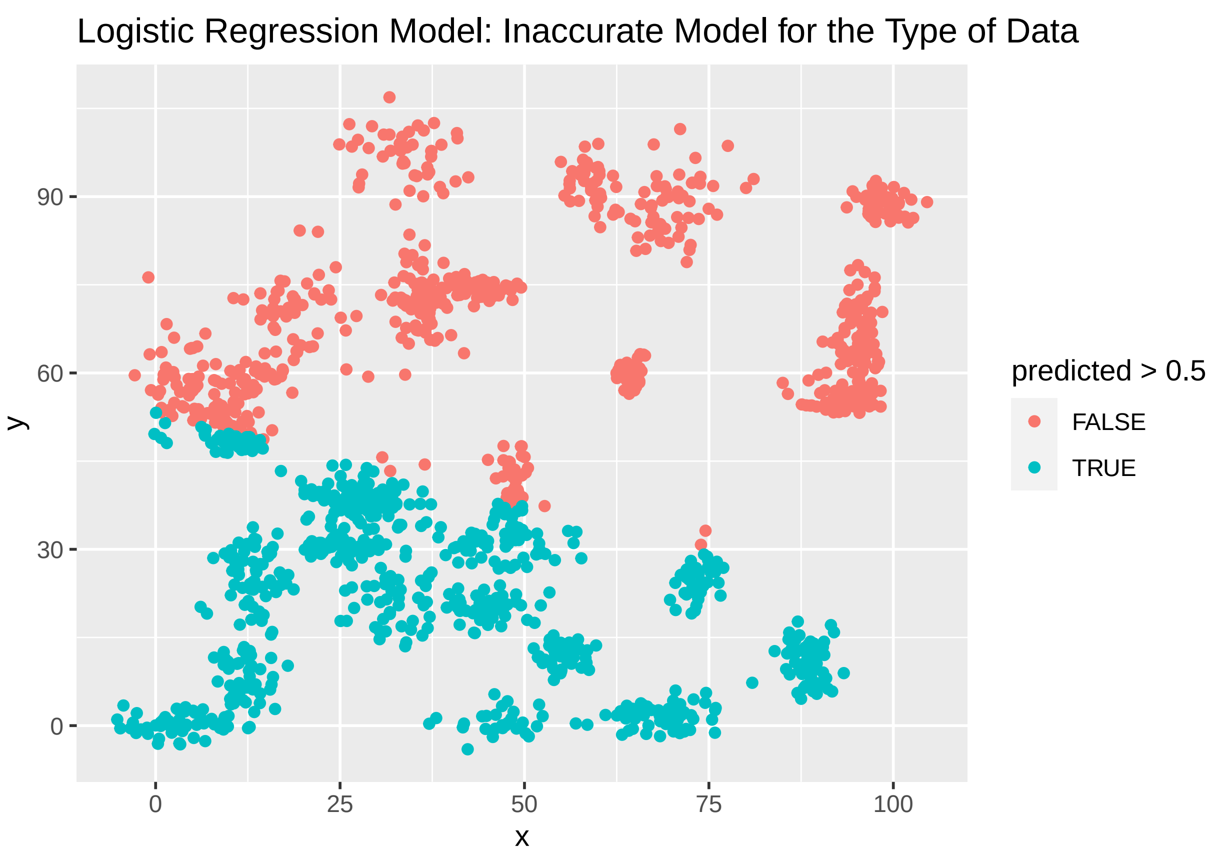
c. Why is the accuracy of the logistic regression classifier different from that of the nearest neighbors?

We can see from the plot above that our data is randomly dispersed tight clusters. Our logistic model only had two inputs, x and y. We can't find any lines that easily cross through our model. Plotting our estimates, we can see that the logistic regression attempted to cut a line through our data.

```
glm.model = glm(label ~ . , family = binomial(logit), data = df)

df$predicted = predict(glm.model, newdata=df, type="response")

ggplot(data = df, aes(x = x, y = y, color = predicted > 0.5)) +
  geom_point() +
  ggtitle("Logistic Regression Model: Inaccurate Model for the Type of Data")
```

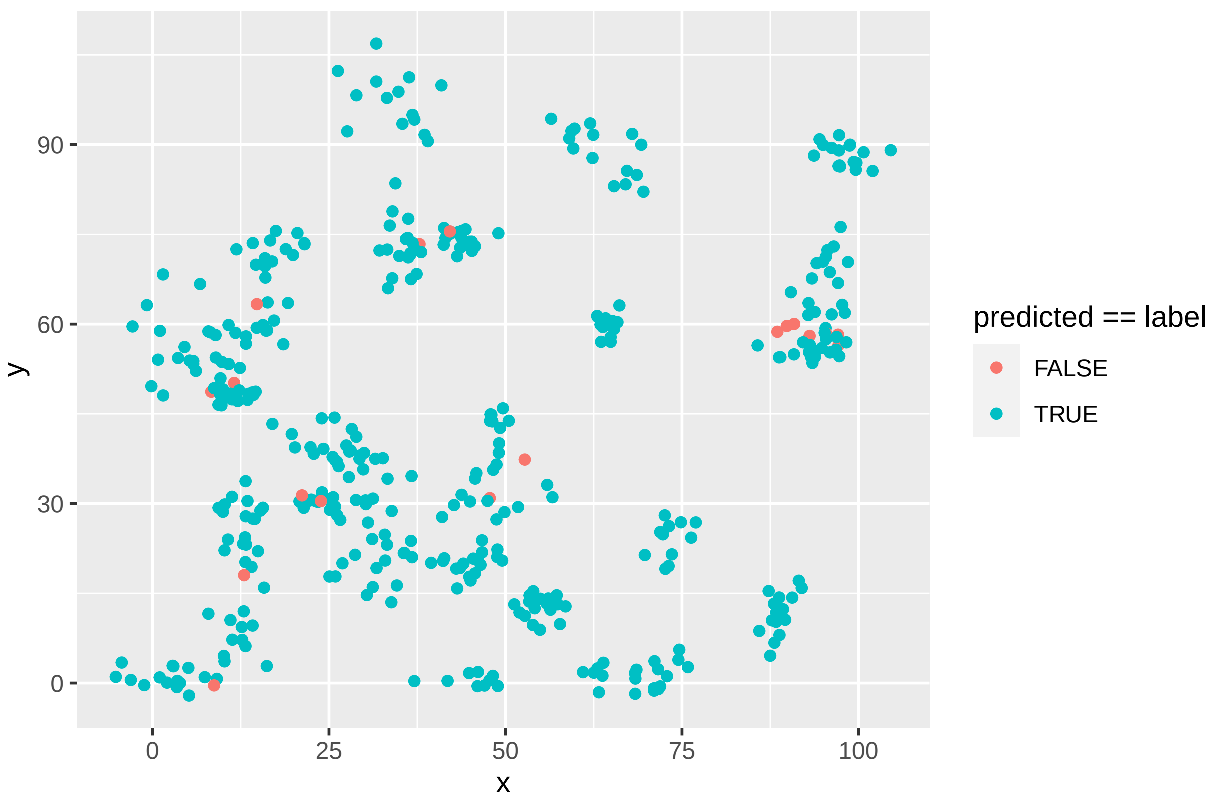


Since our clusters can't be split linearly, a clustering algorithm must be used. We can see from the results of the KNN model that our accuracy is much higher and only report incorrect values when the opposing labeled clusters overlap.

```
test$predicted <- knn(train[2:3], test[2:3], train$label, k = 7)

ggplot(data = test, aes(x = x, y = y, color = predicted == label)) +
  geom_point() +
  ggtitle("KNN Model: Good fit for the data provided")
```

KNN Model: Good fit for the data provided


y
x
predicted == label
FALSE
TRUE