Practical 01: K-Nearest Neighbours

Alex Mounsey

1/20/2021

K-Nearest Neighbours Classifier

Assumptions

- Predictors $x_1, ..., x_p$ are *numerical* variables
  - these are the characteristics of the individuals based on which class can be predicted
- Outcome Y is a *categorical* variable
  - which describes the class *(category)* of an observation

Classification Rule - an algorithm that predicts the category for Y, given the values of the numerical variables $x_1, ..., x_p$.

KNN Classifier

Let...

$$x_0 = (x_{0,1}, x_{0,2}, ..., x_{0,p})$$

...be predictors' values for a new observation which needs to be classified. Given K, the KNN classifier:

1. Identifies the K observations in the training data that are closest to x_0
2. Classifies the observation x_0 to the class that is the most numerous among the K nearest neighbours

Assessing the Accuracy of a Classifier

Training Error Rate

Training Error Rate is the proportion of training observations that are incorrectly classified.

Note: The training data is what is used to train the classifier, therefore the training error will be too optimistic and will not provide an unbiased idea as to how well the classifier performs.

Lab Exercise

Task A

1. Read the information about the dataset by using the command `?Auto`.

```
?Auto
```

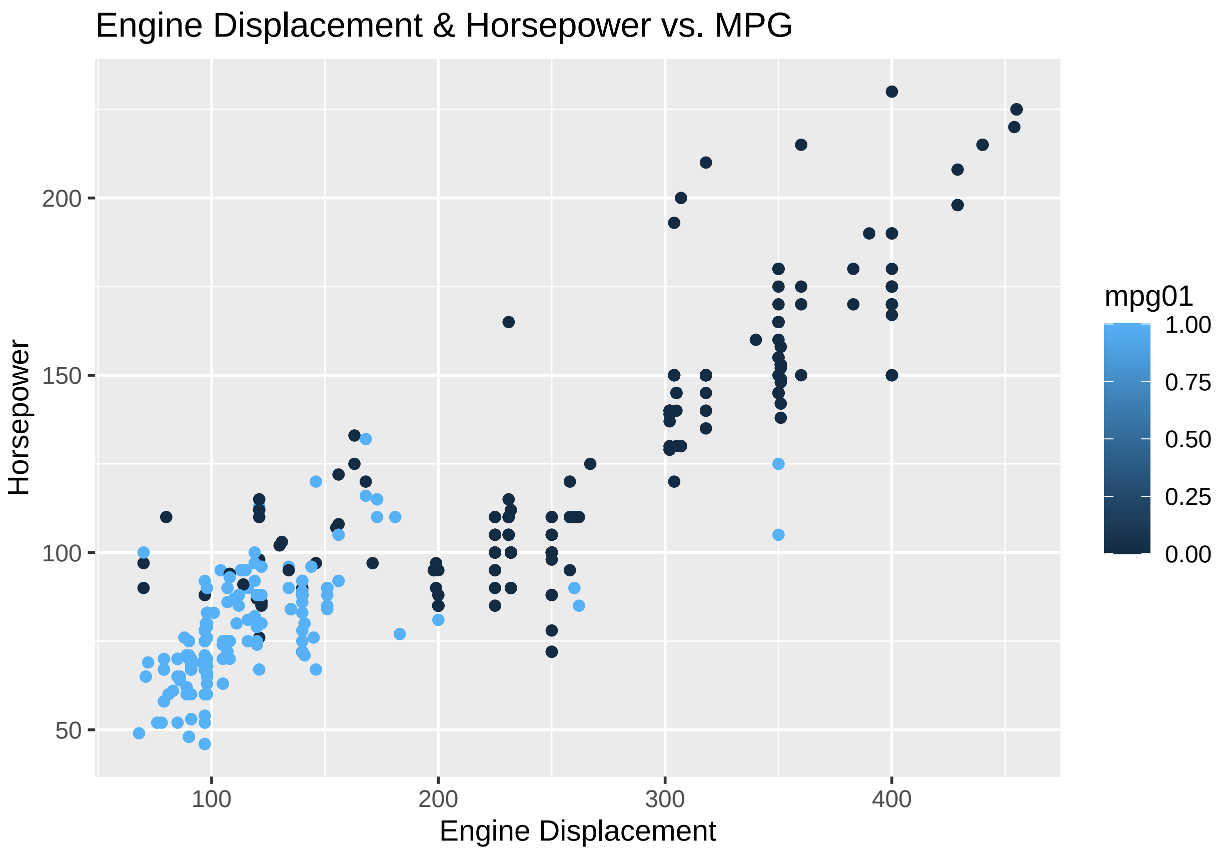
Task B

1. Create a binary variable, `mpg01`, which contains a value of 1 if `mpg` is above its median and 0 otherwise.

```
df <- Auto
df$mpg01 <- as.numeric(df$mpg > median(df$mpg))
```

Task C

1. Produce a scatter plot of `horsepower` against `displacement`
   - Use colour-coding to distinguish between high and low gas mileage.



Task D

1. Create the matrix `train.X` which contains two predictors: `horsepower` and `displacement`
2. Create the vector `cl` of classes
3. Randomly split the data into a training set and a test sset, with 292 and 100 observations respectively

```r
# Create matrix of predictors
train.X <- cbind(df$displacement, df$horsepower)

# Create vector of class labels
cl <- as.factor(df$mpg01)
```

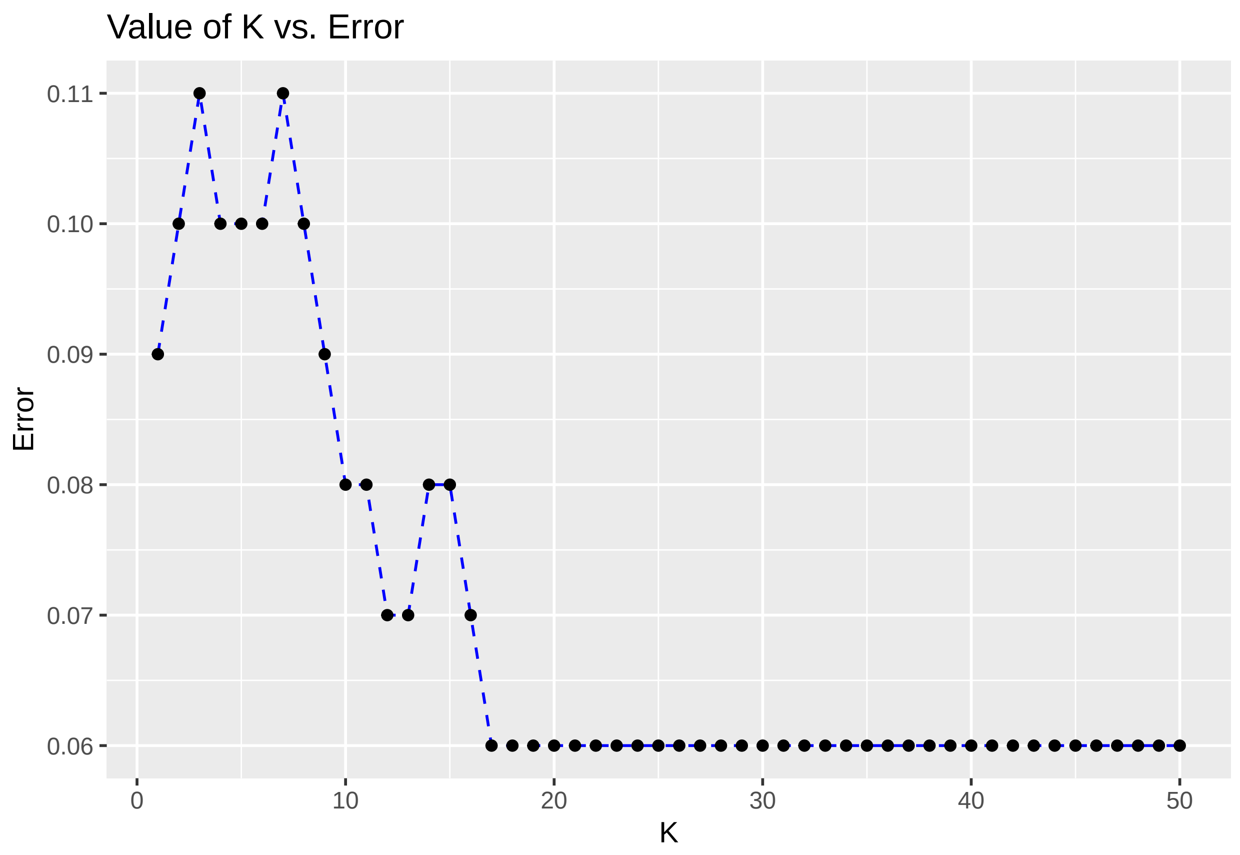
```r
set.seed(1)
df.subset <- sample(392, 100)  # randomly choose 100 numbers of 392

train.X.sub <- train.X[-df.subset, ]  # training predictors
cl.sub <- cl[-df.subset]              # training labels

test.X <- train.X[df.subset, ]  # test predictors
test.cl <- cl[df.subset]        # test labels
```

Task E

1. Construct a KNN classifier using the training data, considering several values of K, in order to predict `mpg01` based on the predictors `displacement` and
`horsepower`.

   - Find the training error and the test error for each of the considered classifiers. Which value of K performs best on this data?

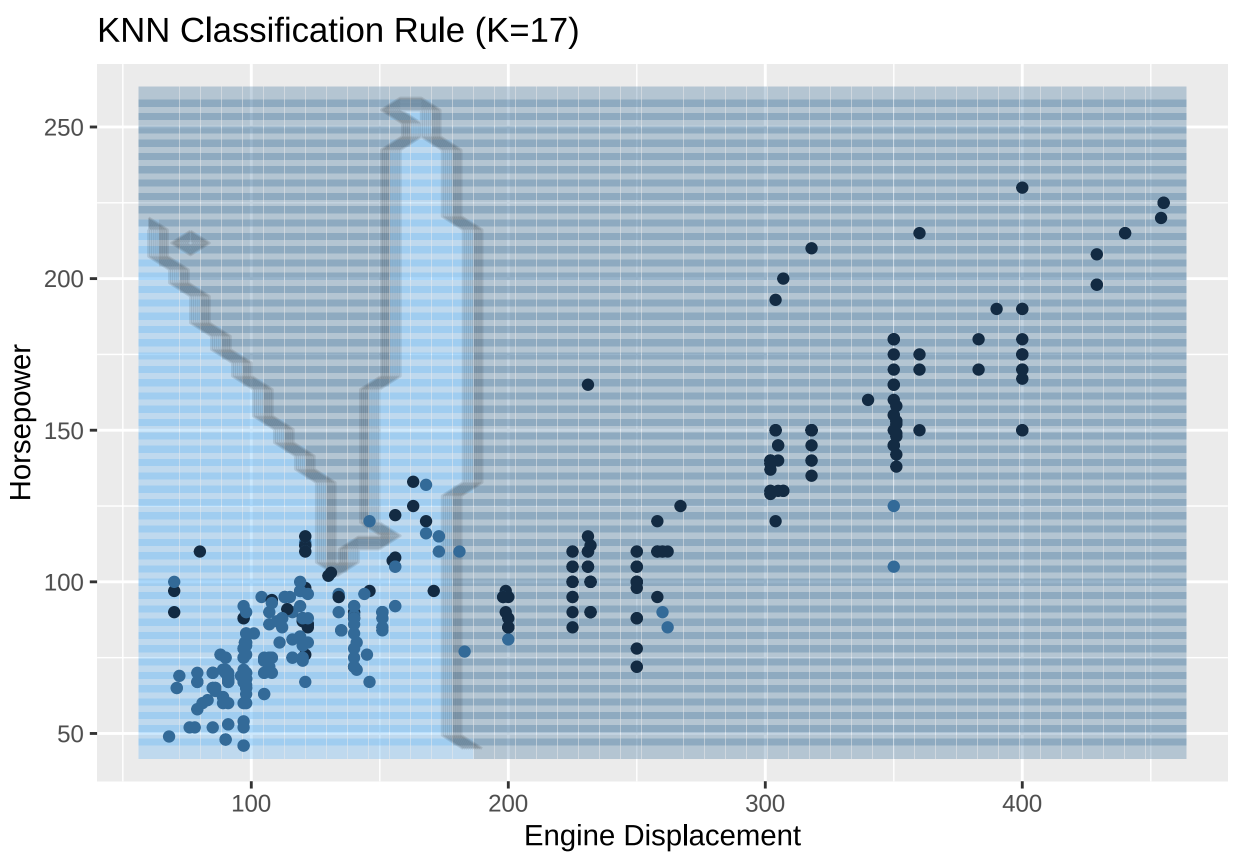
```r
test.error <- function(k) {
  predictions <- knn(train = train.X.sub, test = test.X, cl = cl.sub, k = k)
  results <- table(predictions, test.cl)

  return((results[1,2] + results[2,1]) / sum(results))
}

# Create empty dataframe to store K values and their error
errors_df <- data.frame(k_value = numeric(0), error = numeric(0))

# Calculate the error for a set of K values, add them to 'errors_df'
for (i in 1:50) errors_df[nrow(errors_df) + 1, ] <- c(i, test.error(k = i))
```



Task F

1. Visualise the obtained classification rule in a scatterplot of `horsepower` against `displacement`

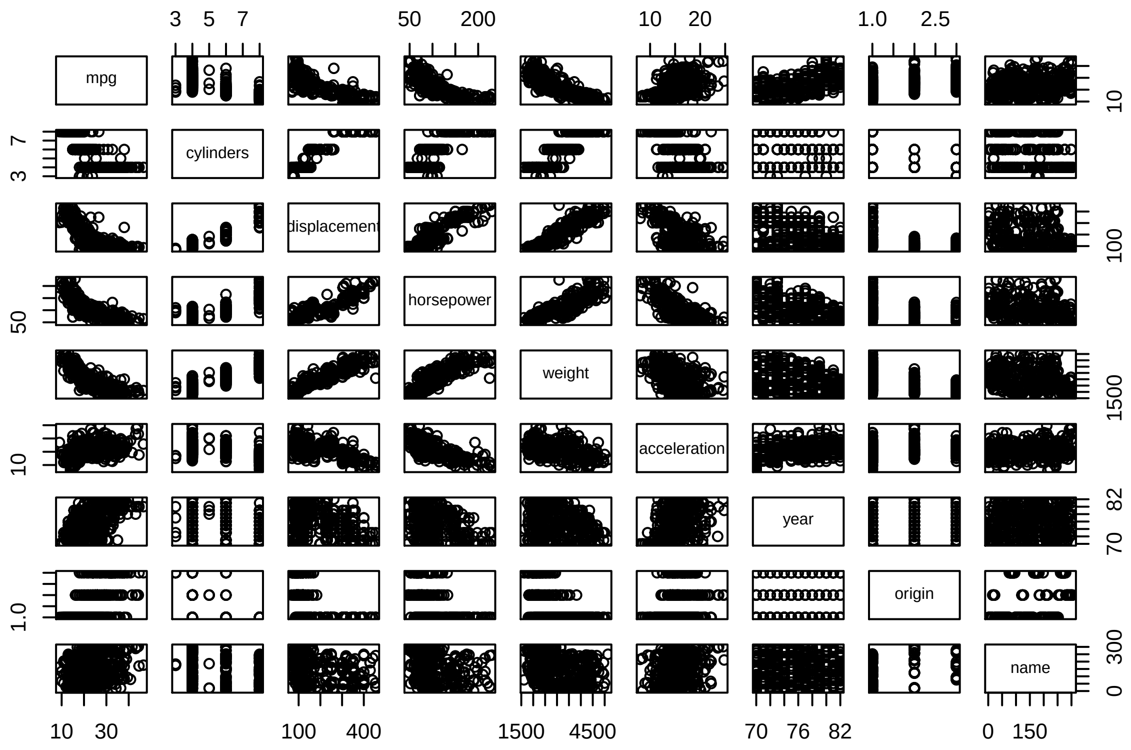
```r
k = 17  # the most suitable K value, from the plot above

len = 50
xp = seq(60, 460, length = len)  # points covering range of displacement
yp = seq(45, 260, length = len)  # points covering range of horsepower
plot_df <- expand.grid(displacement = xp, horsepower = yp)

plot_df$mpg01 <- as.numeric(knn(train = train.X, test = plot_df, cl = cl, k = k))
```



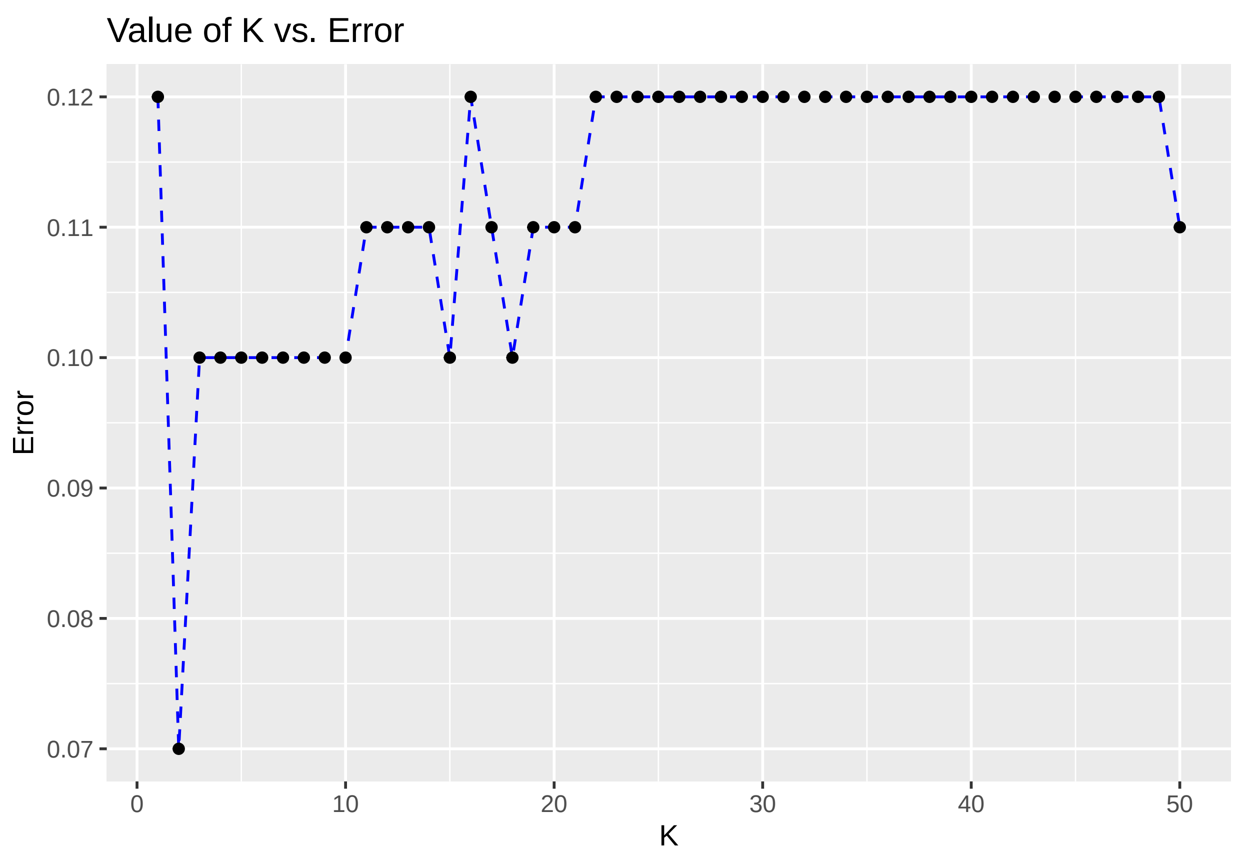
Task G

1. Using the command `pairs(Auto)`, produce scatter plots of all pairs of variables in the dataset in order to investigate the association between mileage and the other features.
   - Which other variables, besides `displacement` and `horsepower` appear most likely to be useful in the prediction of `mpg01`?
2. Perform KNN using one additional predictor.
   - Have you obtained a better classifier?

```
pairs(Auto)
```



```
# KNN with 'weight', 'displacement', and 'horsepower'
train.X <- cbind(df$displacement, df$horsepower, df$weight)

train.X.sub <- train.X[-df.subset, ]  # training predictors
test.X <- train.X[df.subset, ]        # test predictors

# Create empty dataframe to store K values and their error
errors_df <- data.frame(k_value = numeric(0), error = numeric(0))

# Calculate the error for a set of K values, add them to 'errors_df'
for (i in 1:50) errors_df[nrow(errors_df) + 1, ] <- c(i, test.error(k = i))
```



The lowest test error is achieved when K=2. This is lower than the previous classifier.