SDM_Assignment2_2

Sri Balaji Muruganandam

02/10/2021

Setting Working Directory

```
rm(list = ls())
setwd("G:\\SDM_Sem01\\Assignment2")
```

Importing necessary libraries

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

Loading the test and the training data

```
load("zip.test.Rdata")
load("zip.train.Rdata")
```

Checking the Dimension of the data

```
training_data = zip.train
testing_data = zip.test
head(na.omit(training_data),1)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]   [,9] [,10]  [,11] [,12] [,13]
## [1,]    6   -1   -1   -1   -1   -1   -1   -1 -0.631 0.862 -0.167    -1    -1
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23]  [,24] [,25]
## [1,]    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1 -0.992 0.297
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## [1,]     1 0.307    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
##      [,38] [,39] [,40] [,41] [,42]  [,43] [,44] [,45] [,46] [,47] [,48] [,49]
## [1,]    -1    -1 -0.41     1 0.986 -0.565    -1    -1    -1    -1    -1    -1
##      [,50] [,51] [,52] [,53] [,54]  [,55] [,56] [,57] [,58] [,59] [,60] [,61]
## [1,]    -1    -1    -1    -1    -1 -0.683 0.825     1 0.562    -1    -1    -1
##      [,62] [,63] [,64] [,65] [,66] [,67] [,68] [,69]  [,70] [,71] [,72] [,73]
## [1,]    -1    -1    -1    -1    -1    -1    -1    -1 -0.938  0.54     1 0.778
##       [,74] [,75] [,76] [,77] [,78] [,79] [,80] [,81] [,82] [,83] [,84] [,85]
## [1,] -0.715    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1    -1
##      [,86] [,87] [,88]  [,89] [,90] [,91] [,92] [,93] [,94] [,95] [,96] [,97]
## [1,]   0.1     1 0.922 -0.439    -1    -1    -1    -1    -1    -1    -1    -1
##      [,98] [,99] [,100] [,101] [,102] [,103] [,104] [,105] [,106] [,107] [,108]
## [1,]    -1    -1     -1 -0.257   0.95      1 -0.162     -1     -1     -1 -0.987
##      [,109] [,110] [,111] [,112] [,113] [,114] [,115] [,116] [,117] [,118]
## [1,] -0.714 -0.832     -1     -1     -1     -1     -1 -0.797  0.909      1
##      [,119] [,120] [,121] [,122] [,123] [,124] [,125] [,126] [,127] [,128]
## [1,]    0.3 -0.961     -1     -1  -0.55  0.485  0.996  0.867  0.092     -1
##      [,129] [,130] [,131] [,132] [,133] [,134] [,135] [,136] [,137] [,138]
## [1,]     -1     -1     -1  0.278      1  0.877 -0.824     -1 -0.905  0.145
##      [,139] [,140] [,141] [,142] [,143] [,144] [,145] [,146] [,147] [,148]
## [1,]  0.977      1      1      1   0.99 -0.745     -1     -1  -0.95  0.847
##      [,149] [,150] [,151] [,152] [,153] [,154] [,155] [,156] [,157] [,158]
## [1,]      1  0.327     -1     -1  0.355      1  0.655 -0.109 -0.185      1
##      [,159] [,160] [,161] [,162] [,163] [,164] [,165] [,166] [,167] [,168]
## [1,]  0.988 -0.723     -1     -1  -0.63      1      1  0.068 -0.925  0.113
##      [,169] [,170] [,171] [,172] [,173] [,174] [,175] [,176] [,177] [,178]
## [1,]   0.96  0.308 -0.884     -1 -0.075      1  0.641 -0.995     -1     -1
##      [,179] [,180] [,181] [,182] [,183] [,184] [,185] [,186] [,187] [,188]
## [1,] -0.677      1      1  0.753  0.341      1  0.707 -0.942     -1     -1
##      [,189] [,190] [,191] [,192] [,193] [,194] [,195] [,196] [,197] [,198]
## [1,]  0.545      1  0.027     -1     -1     -1 -0.903  0.792      1      1
##      [,199] [,200] [,201] [,202] [,203] [,204] [,205] [,206] [,207] [,208]
## [1,]      1      1  0.536  0.184  0.812  0.837  0.978  0.864  -0.63     -1
##      [,209] [,210] [,211] [,212] [,213] [,214] [,215] [,216] [,217] [,218]
## [1,]     -1     -1     -1 -0.452  0.828      1      1      1      1      1
##      [,219] [,220] [,221] [,222] [,223] [,224] [,225] [,226] [,227] [,228]
## [1,]      1      1      1  0.135     -1     -1     -1     -1     -1     -1
##      [,229] [,230] [,231] [,232] [,233] [,234] [,235] [,236] [,237] [,238]
## [1,] -0.483  0.813      1      1      1      1      1      1  0.219 -0.943
##      [,239] [,240] [,241] [,242] [,243] [,244] [,245] [,246] [,247] [,248]
## [1,]     -1     -1     -1     -1     -1     -1     -1 -0.974 -0.429  0.304
##      [,249] [,250] [,251] [,252] [,253] [,254] [,255] [,256] [,257]
## [1,]  0.823      1  0.482 -0.474 -0.991     -1     -1     -1     -1
```

```
head(na.omit(testing_data),1)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]   [,7]   [,8]  [,9] [,10] [,11] [,12]  [,13]
## [1,]    9   -1   -1   -1   -1   -1 -0.948 -0.561 0.148 0.384 0.904  0.29 -0.782
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]  [,22] [,23] [,24] [,25]
## [1,]    -1    -1    -1    -1    -1    -1    -1    -1 -0.748 0.588     1     1
##      [,26] [,27] [,28] [,29]  [,30] [,31] [,32] [,33] [,34] [,35] [,36]  [,37]
## [1,] 0.991 0.915     1 0.931 -0.476    -1    -1    -1    -1    -1    -1 -0.787
##      [,38] [,39] [,40]  [,41]  [,42]  [,43]  [,44] [,45] [,46]  [,47] [,48]
## [1,] 0.794     1 0.727 -0.178 -0.693 -0.786 -0.624 0.834 0.756 -0.822    -1
##      [,49] [,50] [,51]  [,52] [,53] [,54] [,55]  [,56] [,57] [,58] [,59] [,60]
## [1,]    -1    -1    -1 -0.922  0.81     1  0.01 -0.928    -1    -1    -1    -1
##      [,61] [,62] [,63] [,64] [,65] [,66] [,67] [,68] [,69] [,70] [,71] [,72]
## [1,] -0.39     1 0.271    -1    -1    -1    -1 0.012     1 0.248    -1    -1
##      [,73] [,74] [,75]  [,76] [,77] [,78] [,79]  [,80] [,81] [,82]  [,83] [,84]
## [1,]    -1    -1    -1 -0.402 0.326     1 0.801 -0.998    -1    -1 -0.981 0.645
##      [,85]  [,86] [,87] [,88] [,89] [,90]  [,91] [,92] [,93] [,94] [,95]  [,96]
## [1,]     1 -0.687    -1    -1    -1    -1 -0.792 0.976     1     1 0.413 -0.976
##      [,97] [,98]  [,99] [,100] [,101] [,102] [,103] [,104] [,105] [,106] [,107]
## [1,]    -1    -1 -0.993  0.834  0.897 -0.951     -1     -1     -1 -0.831   0.14
##      [,108] [,109] [,110] [,111] [,112] [,113] [,114] [,115] [,116] [,117]
## [1,]      1      1  0.302 -0.889     -1     -1     -1     -1  0.356  0.794
##      [,118] [,119] [,120] [,121] [,122] [,123] [,124] [,125] [,126] [,127]
## [1,] -0.836     -1 -0.445  0.074  0.833      1      1  0.696 -0.881     -1
##      [,128] [,129] [,130] [,131] [,132] [,133] [,134] [,135] [,136] [,137]
## [1,]     -1     -1     -1     -1 -0.368  0.955      1      1      1      1
##      [,138] [,139] [,140] [,141] [,142] [,143] [,144] [,145] [,146] [,147]
## [1,]  0.905      1      1 -0.262     -1     -1     -1     -1     -1     -1
##      [,148] [,149] [,150] [,151] [,152] [,153] [,154] [,155] [,156] [,157]
## [1,]     -1 -0.507  0.451  0.692  0.692 -0.007 -0.237      1  0.882 -0.795
##      [,158] [,159] [,160] [,161] [,162] [,163] [,164] [,165] [,166] [,167]
## [1,]     -1     -1     -1     -1     -1     -1     -1     -1     -1     -1
##      [,168] [,169] [,170] [,171] [,172] [,173] [,174] [,175] [,176] [,177]
## [1,]     -1     -1  0.155      1  0.436     -1     -1     -1     -1     -1
##      [,178] [,179] [,180] [,181] [,182] [,183] [,184] [,185] [,186] [,187]
## [1,]     -1     -1     -1     -1     -1     -1     -1 -0.991  0.703      1
##      [,188] [,189] [,190] [,191] [,192] [,193] [,194] [,195] [,196] [,197]
## [1,] -0.025     -1     -1     -1     -1     -1     -1     -1     -1     -1
##      [,198] [,199] [,200] [,201] [,202] [,203] [,204] [,205] [,206] [,207]
## [1,]     -1     -1     -1 -0.833  0.959      1 -0.629     -1     -1     -1
##      [,208] [,209] [,210] [,211] [,212] [,213] [,214] [,215] [,216] [,217]
## [1,]     -1     -1     -1     -1     -1     -1     -1     -1     -1   -0.6
##      [,218] [,219] [,220] [,221] [,222] [,223] [,224] [,225] [,226] [,227]
## [1,]  0.998  0.841 -0.932     -1     -1     -1     -1     -1     -1     -1
##      [,228] [,229] [,230] [,231] [,232] [,233] [,234] [,235] [,236] [,237]
## [1,]     -1     -1     -1     -1     -1 -0.424      1  0.732     -1     -1
##      [,238] [,239] [,240] [,241] [,242] [,243] [,244] [,245] [,246] [,247]
## [1,]     -1     -1     -1     -1     -1     -1     -1     -1     -1     -1
##      [,248] [,249] [,250] [,251] [,252] [,253] [,254] [,255] [,256] [,257]
## [1,]     -1 -0.908   0.43  0.622 -0.973     -1     -1     -1     -1     -1
```

```
dim(training_data)
```

```
## [1] 7291  257
```

```
dim(testing_data)
```

```
## [1] 2007  257
```

Taking the number 4 and 7 alone from the dataset

```
head(training_data[,1])
```

```
## [1] 6 5 4 7 3 6
```

```
four_seven1 = which(training_data[,1] == 4 | training_data[,1] == 7)
train_data = data.frame(training_data[four_seven1,])

four_seven2 = which(testing_data[,1] == 4 | testing_data[,1] == 7)
test_data = data.frame(testing_data[four_seven2,])

dim(train_data)
```

```
## [1] 1297  257
```

```
dim(test_data)
```

```
## [1] 347 257
```

Fitting a Linear Model

```
y_train = train_data[,1]
y_test = test_data[,1]
model_fit = lm(y_train~., data = train_data[,-1])
#summary(model_fit)
```

Predicting for the test data and **Rounding off** the final prediction

```
model_predict_test <- predict.lm(model_fit, test_data[,-1])
```

```
## Warning in predict.lm(model_fit, test_data[, -1]): prediction from a rank-
## deficient fit may be misleading
```

```
model_predict_test = round(model_predict_test, digits = 0)
test_MSE <- mean((y_test - model_predict_test)^2)
print(test_MSE)
```

```
## [1] 0.5216138
```

MSE of Linear Regression is 0.521

As we are classifying data. We can use a different metrics as

Calculating Error

```
false_result=which(y_test==model_predict_test)
accuracy=length(false_result)/length(y_test)
print(accuracy)
```

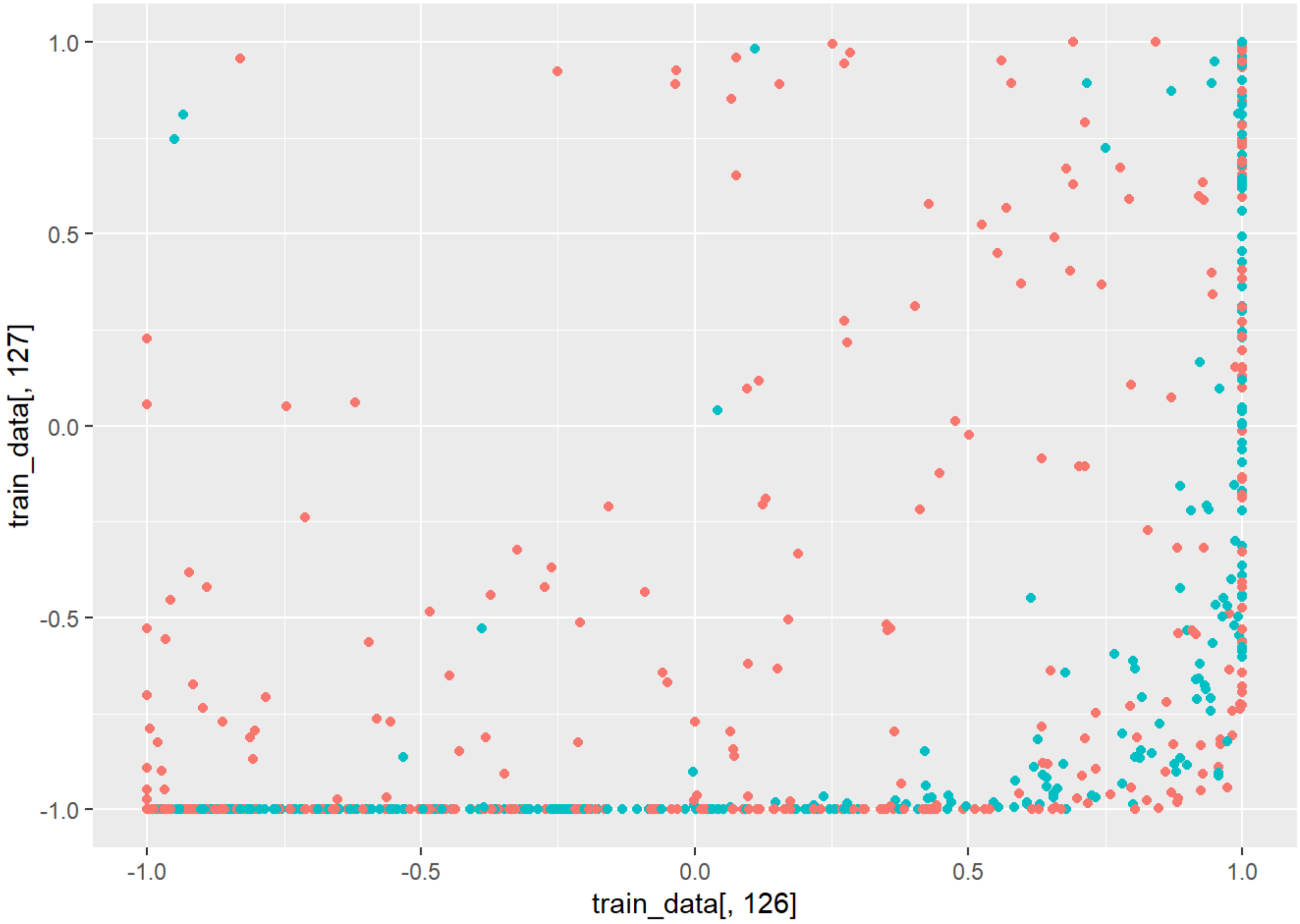
```
## [1] 0.6599424
```

For Example, plotting the graph pixel 126 vs 127 to visualize the data

```
temp_plot = train_data[,126:127]
head(temp_plot)
```

```
##     X126   X127
## 1 -0.977 -1.000
## 2 -1.000 -1.000
## 3  0.958  0.097
## 4 -1.000 -1.000
## 5 -1.000 -1.000
## 6 -1.000 -1.000
```

```
g1 <- ggplot(temp_plot, aes(train_data[,126],train_data[,127])) + geom_point(aes(colour = as.
factor(y_train))) + theme(legend.position = "none")
plot(g1)
```



### Modelling KNN with the values k = 1,3,5,7,9,11,13,15

```r
require(class)
```

```
## Loading required package: class
```

```r
kvalues = c(1,3,5,7,9,11,13,15)
model_knn = c()
error_knn = c()
knn_accuracy = c()
for(i in 1:8) {
  predict_knn <-knn(train_data[,-1],test_data[,-1],y_train,k=kvalues[i])
  #model_knn[i] = predict_knn
  error_knn[i]<- mean(predict_knn != y_test)

  false_result_knn = which(predict_knn == y_test)
  knn_accuracy[i] = length(false_result_knn)/length(y_test)
}
```

Error at each k value

```r
print(error_knn)
```

```
## [1] 0.02305476 0.02017291 0.02017291 0.02593660 0.02881844 0.02881844 0.03170029
## [8] 0.02593660
```

```r
print(min(error_knn))
```

```
## [1] 0.02017291
```

Accuracy at each k value

```r
print(knn_accuracy)
```

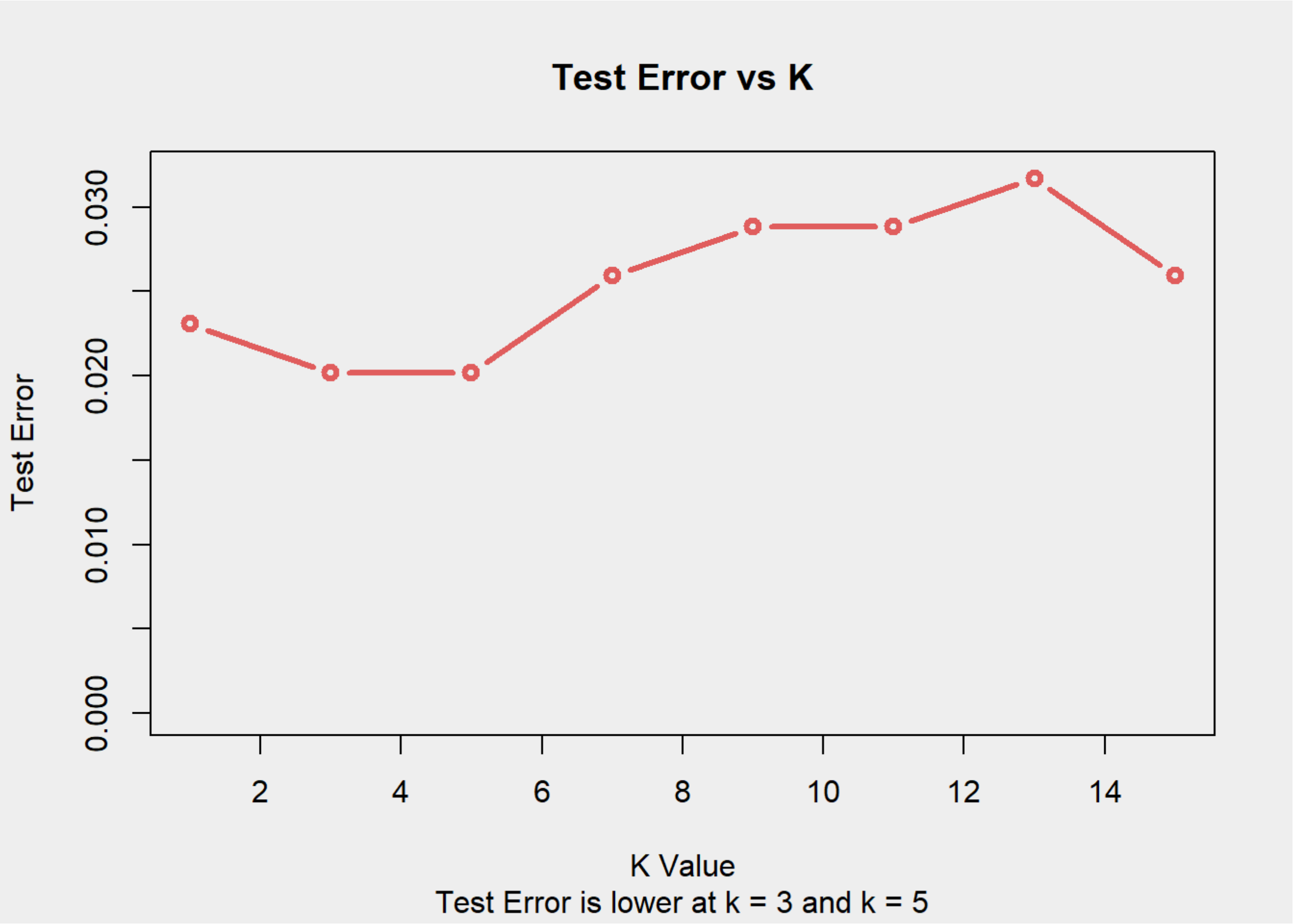
```
## [1] 0.9769452 0.9798271 0.9798271 0.9740634 0.9711816 0.9711816 0.9682997
## [8] 0.9740634
```

```r
print(max(knn_accuracy))
```

```
## [1] 0.9798271
```

Plotting K and the error value corresponding to each k value

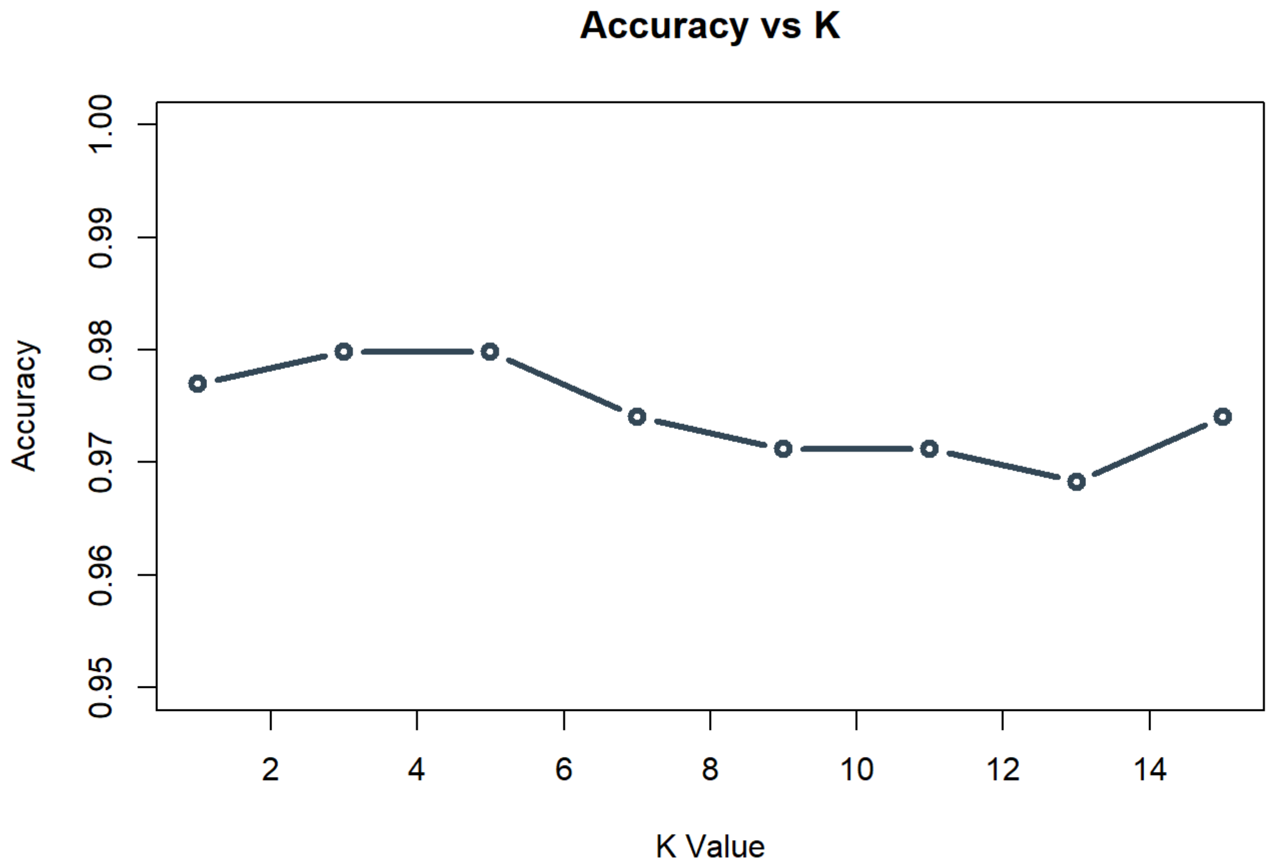
```r
par(bg = '#EEEEEE')
plot(kvalues,error_knn, col="#E05D5D", type = "b", xlab = "K Value", ylab = "Test Error", ylim = c(0,0.032), main = "Test Error vs K", sub="Test Error is lower at k = 3 and k = 5", lwd = 3.0)
```



### Test Error is low when the value of k = 3 and k = 5. ### From the given k values k = 1,3,5,7,9,11,13,15, k =3 and k =5 are the best value to fit the model

Plotting K and Accuracy

```
plot(kvalues,knn_accuracy, col="#334756", type = "b", xlab = "K Value", ylab = "Accuracy", yl
im = c(0.95,1.0), main = "Accuracy vs K", sub="Accuracy is more when k = 3 and k = 5", lwd =
3.0)
```



Accuracy is more when k = 3 and k = 5

When we used Linear Regression to classify the digits we get **65.99%** whereas when we used KNN model with k=3 and k=5 the accuracy is **97.98%**

The Test Error when k = 1,3,5,7,9,11,13,15 are 0.02305476, 0.02017291, 0.02017291, 0.02593660, 0.02881844, 0.02881844, 0.03170029, 0.02593660 respectively. Here the error is low when **k = 3** and **k = 5**