hw1

Libraries

```r
library <- function(...) {suppressPackageStartupMessages(base::library(...))}
if (!require(caret)) install.packages("caret"); library(caret)
```

```
Loading required package: caret

Loading required package: ggplot2

Loading required package: lattice
```

```r
if (!require(dplyr)) install.packages("dplyr"); library(dplyr)
```

```
Loading required package: dplyr


Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

Starting point

We would like to use kNN method for recognizing the type of iris (class label *versicolor* or *virginica*). The predictors include:

- Sepal.Length
- Sepal.Width
- Petal.Length -Petal.Width

The objective of the analysis is to find a model that can be used to predict the type of the flower based on features. Also we will use extensive search approach to identify a good k for the task.

Data management

```
dat <- read.csv('iris.csv') |> filter(Species %in% c("virginica", "versicolor"))
dat$Species <- as.factor(dat$Species)
dat |> head()
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1          7.0         3.2          4.7         1.4 versicolor
2          6.4         3.2          4.5         1.5 versicolor
3          6.9         3.1          4.9         1.5 versicolor
4          5.5         2.3          4.0         1.3 versicolor
5          6.5         2.8          4.6         1.5 versicolor
6          5.7         2.8          4.5         1.3 versicolor
```

Summary statistics

```
summary(dat)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.900   Min.   :2.000   Min.   :3.000   Min.   :1.000
 1st Qu.:5.800   1st Qu.:2.700   1st Qu.:4.375   1st Qu.:1.300
 Median :6.300   Median :2.900   Median :4.900   Median :1.600
 Mean   :6.262   Mean   :2.872   Mean   :4.906   Mean   :1.676
 3rd Qu.:6.700   3rd Qu.:3.025   3rd Qu.:5.525   3rd Qu.:2.000
```

```
 Max.   :7.900   Max.   :3.800   Max.   :6.900   Max.   :2.500  
       Species  
 versicolor:50  
 virginica :50  
```

Both class labels includes 50 observations.

Model creation and diagnostics

Train-Test-Split

We will use 70% of the data as training part and 30% as testing part.

```r
set.seed(1)
N = nrow(dat)
train_ind = sample(N, size = N * 70/100)
test_ind = setdiff(1:N, train_ind)
train_data = dat[train_ind,]
test_data = dat[test_ind,]
```

Fit model

```r
knn_model <- train(
  Species ~ .,
  data = train_data,
  method = "knn",
  tuneGrid = data.frame(k = 5)
)
```

Make predictions

```r
predictions <- predict(knn_model, newdata = test_data)
```

Accuracy for N=5

```r
# Calculate accuracy
confusion_matrix <- confusionMatrix(predictions, test_data$Species)
accuracy_r <- confusion_matrix$overall['Accuracy']
print(paste("kNN (5) Accuracy:", round(accuracy_r, 3)))
```

```
[1] "kNN (5) Accuracy: 0.967"
```

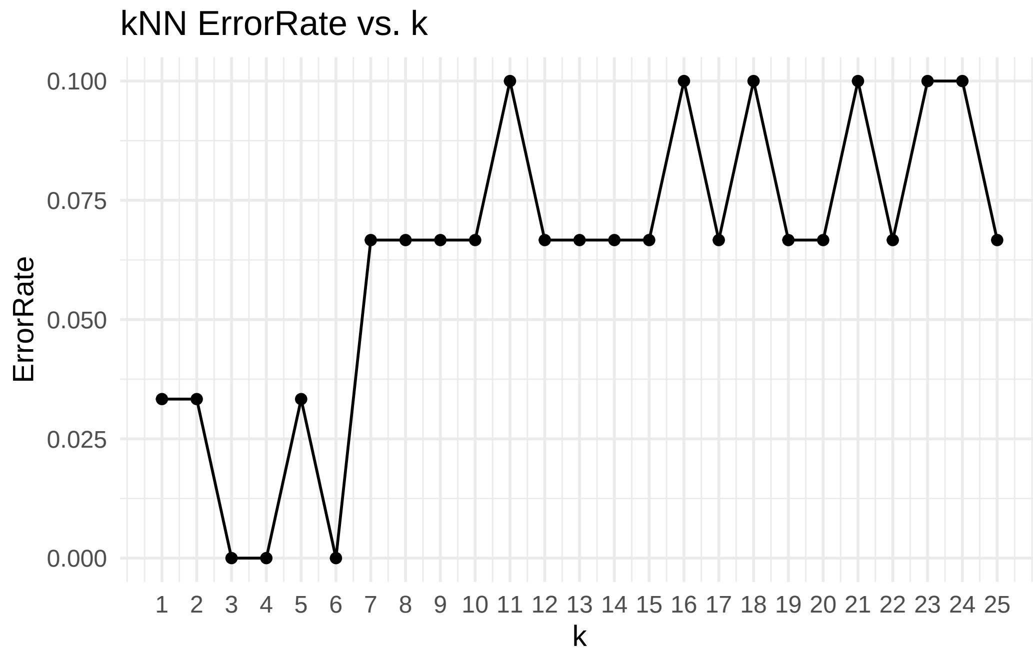
```r
print(paste("kNN (5) Error rate:", round(1-accuracy_r,3)))
```

```
[1] "kNN (5) Error rate: 0.033"
```

```r
set.seed(1)
k_values <- 1:25
error_rate <- numeric(length(k_values))

for (i in k_values) {
  knn_model <- train(
    Species ~ .,
    data = train_data,
    method = "knn",
    tuneGrid = data.frame(k = k_values[i])
  )
  predictions <- predict(knn_model, newdata = test_data)
  confusion_matrix <- confusionMatrix(predictions, test_data$Species)
  error_rate[i] <- 1-confusion_matrix$overall['Accuracy']
}

# Plotting the accuracies
plot_data <- data.frame(k = k_values, ErrorRate = error_rate)
ggplot(plot_data, aes(x = k, y = ErrorRate)) +
  geom_line(aes(group = 1)) +
  geom_point() +
  scale_x_continuous(breaks = k_values) +
  labs(title = "kNN ErrorRate vs. k", x = "k", y = "ErrorRate") +
  theme_minimal()
```



We should consider odd numbers. It make sense to use 1 as good k because the error rate is very small (less then 4%), but the model achieve the absolute 0 error rate when k=3, so it is the best k for our purpose.