# Comprehension Check: Practice with Machine Learning, Part 2

We will practice building a machine learning algorithm using a new dataset, `iris`, that provides multiple predictors for us to use to train. To start, we will remove the `setosa` species and we will focus on the `versicolor` and `virginica` iris species using the following code:

```
library(caret)
data(iris)
iris <- iris[-which(iris$Species=='setosa'),]
y <- iris$Species
```

The following questions all involve work with this dataset.

## Q7

1/1 point (graded)

First let us create an even split of the data into `train` and `test` partitions using `createDataPartition`. The code with a missing line is given below:

```
set.seed(2)    # if using R 3.6 or later, use set.seed(2, sample.kind="Rounding")
# line of code
test <- iris[test_index,]
train <- iris[-test_index,]
```

Which code should be used in place of `# line of code` above?

- `test_index <- createDataPartition(y,times=1,p=0.5)`
- `test_index <- sample(2,length(y),replace=FALSE)`
- **`test_index <- createDataPartition(y,times=1,p=0.5,list=FALSE)`** ✔
- `test_index <- rep(1,length(y))`

**Answer**
Correct:
Good choice! The `createDataPartition` function has a number of parameters that allow the user to specify a test/training partition by the percentage of data that goes to training. See the associated help file.

**Explanation**
`test_index <- createDataPartition(y,times=1,p=0.5,list=FALSE)` is the best answer because the `createDataPartition` function has a number of parameters that allow the user to specify a test/training partition by the percentage of data that goes to training. See the associated help file.

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

## Q8

1/1 point (graded)
Next we will figure out the singular feature in the dataset that yields the greatest overall accuracy when predicting species. You can use the code from the introduction and from Q7 to start your analysis.

Using only the `train` iris dataset, for each feature, perform a simple search to find the cutoff that produces the highest accuracy, predicting virginica if greater than the cutoff and versicolor otherwise. Use the `seq` function over the range of each feature by intervals of 0.1 for this search.

Which feature produces the highest accuracy?

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

**Explanation**
This sample code can be used to determine that `Petal.Length` is the most accurate singular feature.

```
foo <- function(x){
        rangedValues <- seq(range(x)[1],range(x)[2],by=0.1)
        sapply(rangedValues,function(i){
                y_hat <- ifelse(x>i,'virginica','versicolor')
                mean(y_hat==train$Species)
        })
}
predictions <- apply(train[,-5],2,foo)
sapply(predictions,max)
```

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

## Q9

1/1 point (graded)

Using the smart cutoff value calculated on the training data from Q8, what is the overall accuracy in the `test` data?

0.90 ✔ **Answer:** 0.90

0.90

**Explanation**

The code below can be used to calculate the overall accuracy:

```
predictions <- foo(train[,3])
rangedValues <- seq(range(train[,3])[1],range(train[,3])[2],by=0.1)
cutoffs <-rangedValues[which(predictions==max(predictions))]

y_hat <- ifelse(test[,3]>cutoffs[1],'virginica','versicolor')
mean(y_hat==test$Species)
```

Submit You have used 2 of 5 attempts

Answers are displayed within the problem

## Q10

1/1 point (graded)

Notice that we had an overall accuracy greater than 96% in the training data, but the overall accuracy was lower in the test data. This can happen often if we overtrain. In fact, it could be the case that a single feature is not the best choice. For example, a combination of features might be optimal. Using a single feature and optimizing the cutoff as we did on our training data can lead to overfitting.

Given that we know the test data, we can treat it like we did our training data to see if the same feature with a different cutoff will optimize our predictions.

Which feature best optimizes our overall accuracy?

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

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

## Q11

1/1 point (graded)

Now we will perform some exploratory data analysis on the data.

```
plot(iris,pch=21,bg=iris$Species)
```

Notice that `Petal.Length` and `Petal.Width` in combination could potentially be more information than either feature alone.

Optimize the the cutoffs for `Petal.Length` and `Petal.Width` separately in the train dataset by using the `seq` function with increments of 0.1. Then, report the overall accuracy when applied to the test dataset by creating a rule that predicts virginica if `Petal.Length` is greater than the length cutoff OR `Petal.Width` is greater than the width cutoff, and versicolor otherwise.

What is the overall accuracy for the test data now?

0.90 ✔ **Answer:** 0.90

0.90

**Explanation**

The following code can be used to calculate this overall accuracy:

```
library(caret)
data(iris)
iris <- iris[-which(iris$Species=='setosa'),]
y <- iris$Species

plot(iris,pch=21,bg=iris$Species)

set.seed(2)
test_index <- createDataPartition(y,times=1,p=0.5,list=FALSE)
test <- iris[test_index,]
train <- iris[-test_index,]

petalLengthRange <- seq(range(train$Petal.Length)[1],range(train$Petal.Length)
[2],by=0.1)
petalWidthRange <- seq(range(train$Petal.Width)[1],range(train$Petal.Width)[2],by=0.1)

length_predictions <- sapply(petalLengthRange,function(i){
                y_hat <- ifelse(train$Petal.Length>i,'virginica','versicolor')
                mean(y_hat==train$Species)
        })
length_cutoff <- petalLengthRange[which.max(length_predictions)] # 4.7

width_predictions <- sapply(petalWidthRange,function(i){
                y_hat <- ifelse(train$Petal.Width>i,'virginica','versicolor')
                mean(y_hat==train$Species)
        })
width_cutoff <- petalWidthRange[which.max(width_predictions)] # 1.5

y_hat <- ifelse(test$Petal.Length>length_cutoff |
test$Petal.Width>width_cutoff,'virginica','versicolor')
mean(y_hat==test$Species)
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem