PA_HW1

Alex Blohm

8/30/2019

Contents

1.

Page 52, Ex. 2. Explain whether each scenario is a classification or regression problem, and indicate whether we are most interested in inference or prediction. Finally, provide n and p.

(a)

We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

A: It's regression because we are not putting things into categories and we are interested in inference because we want to look at the factors. n = 500, p = 3.

(b)

We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

A: Classification because we are predicting which group (class) the products will go in; success or failure. We are most interested in prediction. n = 20, p = 13.

(c)

We are interesting in predicting the % change in the US dollar in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the dollar, the % change in the US market, the % change in the British market, and the % change in the German market.

A: Regression and prediction. n = 52, p = 3.

2. Page 52, Ex. 4.

Parts A and B only.

A

Describe three real-life applications in which classification might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

1. We could the predict the preferred food of penguins (fish, pellets, other, I'm not a science guy...). Predictors would be like age, species of penguin, zoo/wild, etc. We would be interested in both inference and preciction, so I think inference is more important because I would want more information on food preference changing with the different factors.
2. Predict if students will pass their graduation tests based on race, income, years in school, or more. We would want to look at inference to see how these factors affect graduation.
3. Predicting if a child will pee their pants. Predictors include age, gender, amount drank (double gulp). I think we want inference to explore the effects influencing the output.

B

Describe three real-life applications in which regression might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

1. Predicting the amount of shipments of food, predictors are previous day's shipments, holidays, weather, etc. For accuracy we care more about prediction, we don't need to know much about what affects shipments.
2. Predicting profits given amount sold, company costs, etc. We would want inference to analyze all the factors and which are important to the business.
3. Predict time it takes to run a marathon given training time, diet factors, height, age, etc. I'd want to focus on inference to look at the factors and how they affect time.

3. (UNDERGRAD ONLY)

Skip

4.

The training data (http://bit.ly/340jDyc) set contains 175 observations classified as red or green. The test data set (http://bit.ly/2L7uhdO) contains 1750 observations classified as either red or green.

```
train <- read.csv("https://www.dropbox.com/s/nnzaeo73zl2ktwg/PA_HW1_train.csv?dl=1")
test <- read.csv("https://www.dropbox.com/s/u6iuxfm4lz8isg6/PA_HW1_test.csv?dl=1")
```

K-nearest

Perform k-nearest neighbor classification using the training data with k = 1. Use this model to predict the class of each observtion in the training data set. How many observations were incorrectly classified? Is this good?

```
library(class)
knn_fit <- knn(train[c(1, 2)], train[c(1, 2)], cl = train$col, k = 1, prob = T)
table(knn_fit, train$col)
```

```
##
## knn_fit green red
##   green    75   0
##   red       0 100
```

No observations were incorrectly classified. This is bad, this is overfit because k = 1. It categorizes perfectly.

Again using k = 1,

build a classification model with the training data set and use it to classify the observations in the test data set. How many obervations were incorrectly classified? Is this good?

```
knn_fit <- knn(train[c(1, 2)], test[c(1, 2)], cl = train$col, k = 1, prob = T)
t <- table(knn_fit, test$col)

t[1,2]+t[2,1]
```

```
## [1] 719
```

```
cat(367+352, "were wrongly classified out of", length(knn_fit), "\n")
```

```
## 719 were wrongly classified out of 1750
```

```
incorrect <- (367+352)/ (length(knn_fit))
cat("Which is", incorrect, "percent wrong")
```

```
## Which is 0.4108571 percent wrong
```

This is 2/5 incorrectly classified which seems pretty terrible to me...

(GRAD ONLY)

Train a model for for each value of k between 1 and 100. For each model, predict the class of the observations in the training data set and the observations in the test data set. Make a plot of the value of k on the x-axis and the error rate on the y- axis. (This is similar to Figure 2.4 in the book Elements of Statistical Learning) Comment on this graph.

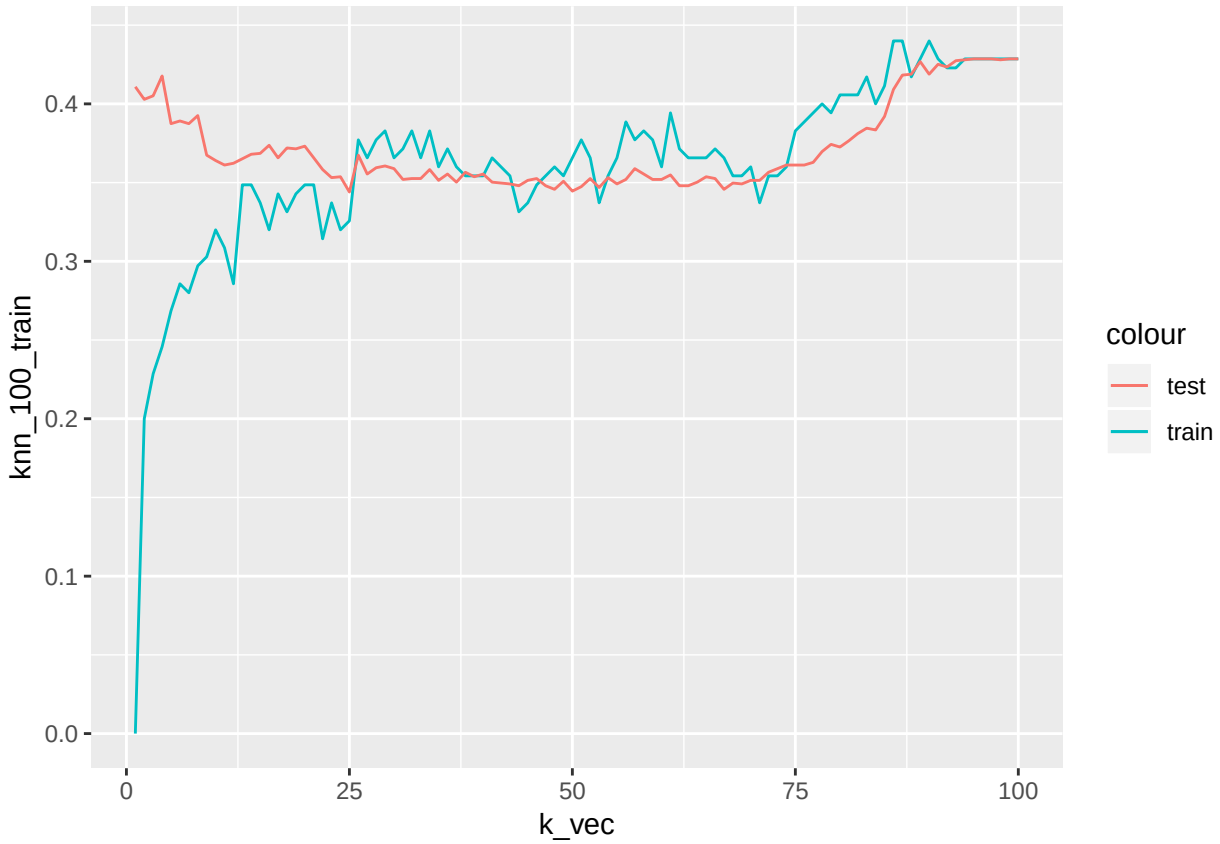
```r
set.seed(1)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidy
## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.2.1       v forcats 0.3.0
## Warning: package 'ggplot2' was built under R version 3.5.2
## Warning: package 'tibble' was built under R version 3.5.2
## Warning: package 'tidyr' was built under R version 3.5.2
## Warning: package 'purrr' was built under R version 3.5.2
## Warning: package 'dplyr' was built under R version 3.5.2
## Warning: package 'stringr' was built under R version 3.5.2
## -- Conflicts ------------------------------------------------------------------------- tidyverse_
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
k_vec <- c(1:100)

#knn(train[c(1, 2)], test[c(1, 2)], cl = train$col, k = 1, prob = T)
#knn(train[c(1, 2)], test[c(1, 2)], cl = train$col, k = 2, prob = T)

knn_100_test <- sapply(k_vec,
       function(x){
    output_knn <- knn(train[c(1, 2)], test[c(1, 2)], cl = train$col, k = x, prob = T)
    table <- table(output_knn, test$col)
    error <- (table[2]+table[3])/1750
    return(error)
        }
)
knn_100_train <- sapply(k_vec,
       function(x){
    output_knn <- knn(train[c(1, 2)], train[c(1, 2)], cl = train$col, k = x, prob = T)
    table <- table(output_knn, train$col)
    #print(table)
    error <- (table[2]+table[3])/175
    return(error)
        }
)
ggplot() + geom_line(aes(x=k_vec, y = knn_100_train, color = "train")) + geom_line(aes(x=k_vec, y= knn_
```

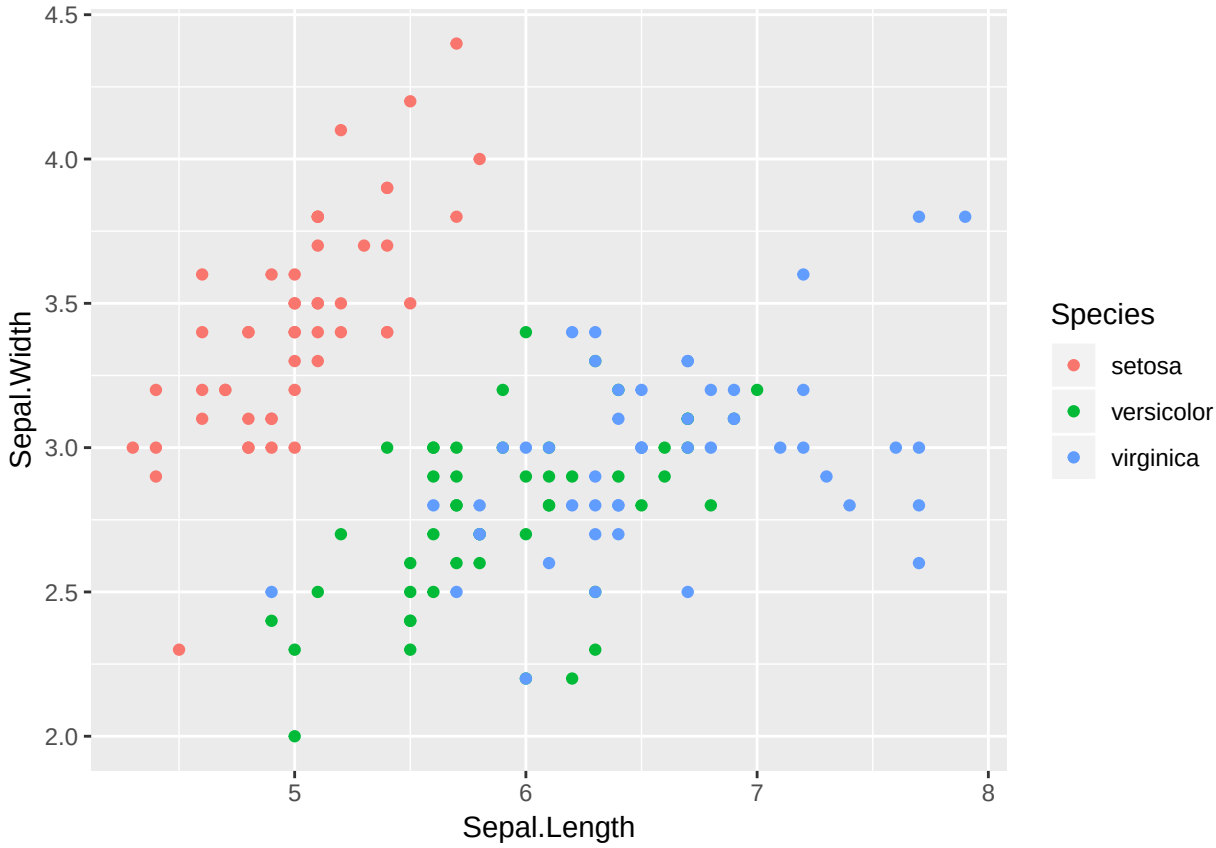


With k from 0 to 10 the model seems to be overfit. I believe the training and test errors should be pretty similar and we want smaller error rates and those seems to be similar from k = 25 until 100, however the error rate increases around k = 70.

5.

Plot all irises based on their Sepal.Length and Sepal.Width values using different colors for each species.

```
p <- ggplot(iris, aes(x = Sepal.Length, y= Sepal.Width, color = Species)) + geom_point()
p
```



6.

Perform linear discriminant analysis using the iris data with only Sepal.Length and Sepal.Width as predictors. Make predictions about the species of each iris and create a confusion matrix for this predictions.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
lda_fit <- lda(Species ~ Sepal.Length + Sepal.Width, data=iris)
lda_pred <- predict(lda_fit)
table(iris$Species, lda_pred$class)
```

```
##
##              setosa versicolor virginica
##   setosa         49          1         0
##   versicolor      0         36        14
##   virginica       0         15        35
```

7. (GRAD ONLY)

On the plot produced in part 1, plot the predicted value of species over a grid of points using the same color scheme used in part 1.

```
grid <- expand.grid(seq(min(iris$Sepal.Width), max(iris$Sepal.Width), by = .01), seq(min(iris$Sepal.Leng
names(grid) <- c("Sepal.Width", "Sepal.Length")
preds <- predict(lda_fit, grid)
boundries <- data.frame(preds$class, grid)

p + geom_point(data=boundries, aes(x = Sepal.Length, y= Sepal.Width, color = preds.class), alpha=.01)
```