Midterm II Study Guide and Review

Deepak Bastola

2024-05-15

Midterm II Study Guide

Format: In-class with open-ended questions.

One-sided Cheat-sheet allowed (A4 paper) and a basic calculator allowed.

- `shiny` and `purrr` cheatsheet will be provided
- You are not permitted to use a laptop or classroom computer.

Topics

- The exam covers iterations, functionals, web scraping, shiny, kNN, linear regression and logistic regression (through Mon. 05/20)
- You will be evaluated on your understanding of the tools, algorithms, Shiny function logic, and the construction of the workflow in R that we have covered in class. While I won't require you to write complex code from scratch, you should be ready to write brief segments of code. Other methods I might use to assess your knowledge of R include:
  - Identifying the error in written code.
  - Putting lines of code in order to complete a specified task.
  - Describing the output resulting from a code/code-chunk.

Sample Questions

Q1

Given below are the monthly deaths from bronchitis, emphysema and asthma in the UK from 1974 to 1979.

```
knitr::kable(mydata)
```

year	jan	feb	mar	apr	may	jun	jul	aug	sep	oct	nov	dec
1974	3035	1721	2933	1607	2787	1489	3102	1498	2815	1529	3084	1461
1975	2552	1524	2889	1545	3891	1300	2294	1361	3137	1366	2605	1354
1976	2704	1596	2938	1396	3179	1356	2385	1346	2679	1357	2573	1333
1977	2554	2074	2497	1787	2011	1653	2444	1564	1969	1570	2143	1492
1978	2014	2199	1870	2076	1636	2013	1748	1640	1870	1535	1693	1781
1979	1655	2512	1726	2837	1580	2823	1554	2293	1633	2491	1504	1915

a. Describe what is returned by the code below, including the type of R object produced, the length or dimension of the object, and the information contained in the object.

```
map_dbl(mydata %>% select(-1), mean) %>% mean()
```

b. Describe what is returned by the code below, including the type of R object produced, the length or dimension of the object, and the information contained in the object.

```
ratio_fun <- function(x) quantile(x, probs= c(0.25, 0.5, 0.75))
map_dfc(mydata %>% select(-1), ratio_fun)
```

c. Describe what is returned by the code below, including the type of R object produced, the length or dimension of the object, and the information contained in the object.

```
lapply(mydata %>% select(-1), ratio_fun) %>%
  unlist()
```

Q2 Shiny UI and Server Logic

a. Describe the relationship between `ui` and `server` in a Shiny application. How do they interact?

b. Given the code snippet below, identify and explain any errors that would prevent the app from displaying a histogram of the generated data.

```
library(shiny)
ui <- fluidPage(
  titlePanel("Data Histogram"),
  sidebarLayout(
    sidebarPanel(
      sliderInput("obs", "Number of observations:", min = 0, max = 1000, value = 500)
    ),
    mainPanel(
      plotOutput("distPlot")
    )
  )
)

server <- function(input, output) {
  output$distPlot <- renderPlot({
    data <- rnorm(input$obs, input = 100)
    hist(data)
  })
}

shinyApp(ui = ui, server = server)
```

c. Consider a Shiny app that allows users to select a dataset from a dropdown menu and displays a summary of that dataset. Identify potential problems in this hypothetical Shiny code snippet and suggest improvements.

```
library(shiny)
ui <- fluidPage(
```

```
  selectInput("dataset", "Choose a dataset:", choices = c("mtcars", "iris")),
  tableOutput("summaryTable")
)

server <- function(input, output) {
  data <- reactive({
    switch(input$dataset,
           "mtcars" = mtcars,
           "iris" = iris)
  })

  output$summaryTable <- renderTable({
    summary(data())
  })
}

shinyApp(ui = ui, server = server)
```

d. Given a Shiny application that uses actionButton to trigger calculations, explain the purpose of this input type and describe a scenario where it might be useful.

```
ui <- fluidPage(
  actionButton("calc", "Calculate"),
  numericInput("num", "Enter a number:", 10),
  textOutput("result")
)

server <- function(input, output) {
  observeEvent(input$calc, {
    result <- sqrt(input$num)
    output$result <- renderText({
      paste("The square root is:", result)
    })
  })
}

shinyApp(ui = ui, server = server)
```

Q3 Web scraping

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Sample Web Page</title>
</head>
<body>
    <h1>Welcome to My Website</h1>
    <p>This is a paragraph of text on the main page.</p>
    <a href="/about">About Us</a>
    <a href="/contact">Contact Us</a>
</body>
</html>
```

Assume the HTML above is served from `http://mywebpage.com`.

a. Write the R code to extract and print the text inside the `<h1>` tag using `rvest` and/or `polite`.

b. Write the R code to extract the href attributes of all links in the document.

c. Given the base URL `http://mywebpage.com`, write the R code to convert the relative URLs found in the href attributes into absolute URLs.

d. Write the R code to extract and print the text of each link (tag).

Q4 : K-nearest neighbor

Consider the `Smarket` dataset, which provides daily percentage returns for the S&P 500 stock index from 2001 to 2005. Your task is to fit a K-nearest neighbor model to predict the direction of stock returns.

```
set.seed(1234)

data_Smarket <- as_tibble(Smarket)
split <- initial_split(data_Smarket, strata = Direction, prop = 4/5)
Smarket_train <- training(split)
Smarket_test <- testing(split)

# glimpses of data
glimpse(Smarket_train)
## Rows: 999
## Columns: 9
## $ Year      <dbl> 2001, 2001, 2001, 2001, 2001, 2001, 2001, 2001, 2001, 2001, ~
## $ Lag1      <dbl> 1.032, 1.392, -0.498, 0.546, 0.359, -0.623, 1.183, -0.865, 0~
## $ Lag2      <dbl> 0.959, 0.213, 0.287, -0.562, -1.747, -0.841, -1.334, 1.183, ~
## $ Lag3      <dbl> 0.381, 0.614, 1.303, 0.701, 0.546, -0.151, -0.623, -1.334, -~
## $ Lag4      <dbl> -0.192, -0.623, 0.027, 0.680, -0.562, 0.359, -0.841, -0.623,~
## $ Lag5      <dbl> -2.624, 1.032, -0.403, -0.189, 0.701, -1.747, -0.151, -0.841~
## $ Volume    <dbl> 1.4112, 1.4450, 1.2580, 1.1188, 1.0130, 1.1072, 1.0391, 1.07~
## $ Today     <dbl> -0.623, -0.403, -0.189, -1.747, -0.151, -1.334, -0.865, -0.2~
## $ Direction <fct> Down, Down, Down, Down, Down, Down, Down, Down, Down, Down, ~
glimpse(Smarket_test)
## Rows: 251
## Columns: 9
## $ Year      <dbl> 2001, 2001, 2001, 2001, 2001, 2001, 2001, 2001, 2001, 2001, ~
## $ Lag1      <dbl> 0.381, 0.614, 0.213, 0.287, 0.701, -0.562, -0.151, -0.841, -~
## $ Lag2      <dbl> -0.192, -0.623, 0.614, 1.303, 0.680, 0.701, 0.359, -0.151, -~
## $ Lag3      <dbl> -2.624, 1.032, -0.623, 0.027, -0.189, 0.680, -1.747, 0.359, ~
## $ Lag4      <dbl> -1.055, 0.959, 1.032, -0.403, -0.498, -0.189, 0.546, -1.747,~
## $ Lag5      <dbl> 5.010, 0.381, 0.959, 1.392, 0.287, -0.498, -0.562, 0.546, 0.~
## $ Volume    <dbl> 1.19130, 1.20570, 1.34910, 1.30900, 1.14980, 1.29530, 1.0596~
## $ Today     <dbl> 0.959, 0.213, 1.392, -0.498, -0.562, 0.546, -0.841, -0.623, ~
## $ Direction <fct> Up, Up, Up, Down, Down, Up, Down, Down, Up, Up, Up, Up, Up, ~
```

a. . The following code splits the Smarket dataset into a training set and a test set. Explain the purpose of splitting the data and what stratification accomplishes in this context.

```
Smarket_recipe <- recipe(Direction ~ Lag1 + Lag2 + Lag3 + Year + Volume,
                         data = Smarket_train) %>%
  step_center(all_predictors()) %>%
  step_scale(all_predictors())

Smarket_knn_spec <- nearest_neighbor(mode = "classification",
                             engine = "kknn",
                             weight_func = "rectangular",
                             neighbors = 5)

Smarket_workflow <- workflow() %>%
```

```
  add_recipe(Smarket_recipe) %>%
  add_model(Smarket_knn_spec)

Smarket_fit <- fit(Smarket_workflow, data = Smarket_train)

test_features <- Smarket_test %>% select(Direction, Lag1, Lag2, Lag3, Year, Volume)
nn1_pred3 <- predict(Smarket_fit, test_features, type = "raw")
Smarket_results <- Smarket_test %>%
  select(Direction) %>%
  bind_cols(predicted = nn1_pred3) %>% mutate(Direction = as.factor(Direction))
```

b. The following trained model is used to produce a data-frame of the actual and predicted `Direction` in the test dataset. Call this data-frame `Smarket_results`. What information does `Smarket_results` contain? What is the dimension of this dataset? Explain.

```
conf_mat(Smarket_results, truth = Direction, estimate = predicted)
##           Truth
## Prediction Down Up
##       Down   50 58
##       Up     71 72
```

c. Calculate the sensitivity, specificity, accuracy, and positive predictive value of the classifier based on the confusion matrix below. Discuss the interpretation of these metrics in the context of this problem.

Q5 Miscellaneous

(a) Differentiate between supervised and unsupervised learning by providing examples of situations where each would be most appropriate.

(b) Discuss the significance of feature scaling in the K-NN algorithm and illustrate its impact with a hypothetical example.

(c) (True/False) Recall is a more relevant metric than precision in scenarios where False Negatives are more detrimental than False Positives. Explain your answer.

(d) Explain the rationale behind data preprocessing in the KNN algorithm, discussing how the algorithm's performance might be influenced by unprocessed data.

(e) (Multiple Choice) Among the following values for K in K-NN, which is most likely to cause underfitting?

1. 30
2. 5
3. 1

(f) Logistic regression is a machine learning algorithm that is typically used to predict the probability of what kind of variable? Explain the reasoning behind your choice.

(A) categorical independent variable
(B) categorical dependent variable.
(C) numerical dependent variable.
(D) numerical independent variable.

(g) For a K-NN classification model, how would you expect the model's performance to change as K increases? Discuss considering both bias-variance trade-off and the model's complexity.

(h) In logistic regression, which link function is commonly used?

A) Logit
B) Probit
C) Identity
D) Log

(i) Which method of data splitting ensures that the training and test sets have approximately the same percentage of samples of each class?

A) Random split
B) Stratified split
C) Sequential split
D) Clustered split