

Midterm III Study Guide and Review

Deepak Bastola

2022-11-11

Midterm III Study Guide

Format: In Class with open-ended questions.

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

- You may use a calculator
- You are not permitted to use a laptop or classroom computer.

Topics

- The exam covers various machine learning topics including k-nearest neighbor, k-means, decision trees and random forests (through Fri. 11/11)
- You will be tested on your conceptual understanding of the machine learning algorithms, the accuracy metrics, and the associated construction of the workflow in R that we have discussed in the class. I will not make you write extremely complicated code from scratch, but be prepared to write small chunks of code. Additional ways I could assess your understanding of R include (but are not limited to):
 - 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: Random Forest

The following example uses `Carseats` data from ISLR package to classify the amount of sales (High or Low) depending on certain number of features.

```
Carseats <- as_tibble(Carseats) %>%
  mutate(High = factor(if_else(Sales <= 8, "No", "Yes"))) %>%
  select(-Sales)
glimpse(Carseats)
## Rows: 400
## Columns: 11
## $ CompPrice <dbl> 138, 111, 113, 117, 141, 124, 115, 136, 132, 132, 121, 117~
## $ Income <dbl> 73, 48, 35, 100, 64, 113, 105, 81, 110, 113, 78, 94, 35, 2~
## $ Advertising <dbl> 11, 16, 10, 4, 3, 13, 0, 15, 0, 0, 9, 4, 2, 11, 11, 5, 0, ~
## $ Population <dbl> 276, 260, 269, 466, 340, 501, 45, 425, 108, 131, 150, 503, ~
## $ Price <dbl> 120, 83, 80, 97, 128, 72, 108, 120, 124, 124, 100, 94, 136~
## $ ShelfLoc <fct> Bad, Good, Medium, Medium, Bad, Bad, Medium, Good, Medium, ~
## $ Age <dbl> 42, 65, 59, 55, 38, 78, 71, 67, 76, 76, 26, 50, 62, 53, 52~
## $ Education <dbl> 17, 10, 12, 14, 13, 16, 15, 10, 10, 17, 10, 13, 18, 18, 18~
## $ Urban <fct> Yes, Yes, Yes, Yes, Yes, No, Yes, Yes, No, No, No, Yes, Ye~
## $ US <fct> Yes, Yes, Yes, Yes, No, Yes, No, Yes, No, Yes, Yes, Yes, Yes, N~
## $ High <fct> Yes, Yes, Yes, No, No, Yes, No, Yes, No, No, Yes, Yes, No, ~
```

(a) Roughly, how many observations are in the test and train datasets? *Answer:* 100 and 300, respectively.

```
set.seed(1234)
Carseats_split <- initial_split(Carseats, prop = 0.75)
Carseats_train <- training(Carseats_split)
Carseats_test <- testing(Carseats_split)
```

(b) Why do we need to split the data into training and testing set? Explain. *Answer:*

We need to split the data into training and testing to train the model in a random subset and test the model in a different subset to get an estimate of a final, unbiased assessment of the model's performance.

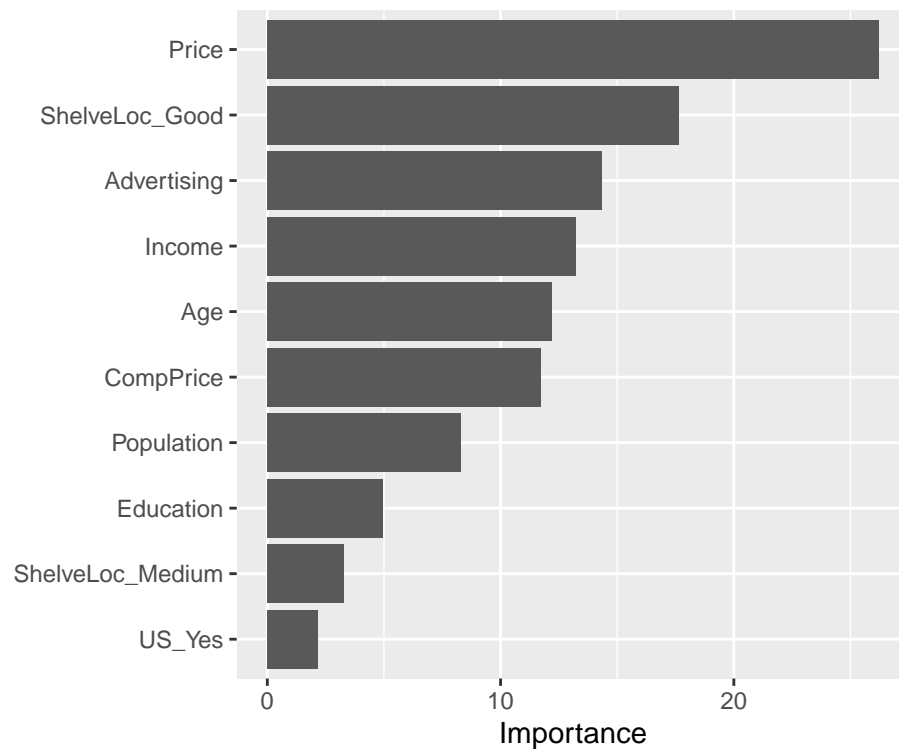
```
Carseats_recipe <- recipe(High ~ ., data = Carseats_train) %>%
  step_dummy(all_nominal(), -all_outcomes()) %>%
  prep()
```

```
decision_tree_rpart_spec <- rand_forest(mtry = tune()) %>%
  set_engine('ranger', importance = "impurity") %>%
  set_mode('classification')
```

(c) Explain the role of `tune()` inside the model specification. *Answer:*

The `tune()` is a placeholder, that will be used to tune the number of predictors that will be randomly sampled at each split when creating the tree models

(d) The importance plot for this algorithm is shown below. Which two predictors are most important? *Answer:* The two most important predictors according to random forest model are Price and ShelfLoc.



```
tree_last_fit <- final_tree_workflow %>% last_fit(Carseats_split)
tree_predictions <- tree_last_fit %>% collect_predictions()
conf_mat(tree_predictions, truth = High, estimate = .pred_class)
##           Truth
## Prediction No Yes
##           No  46  12
##           Yes  10  32
```

(e) Calculate the accuracy metric of this algorithm based on the following confusion matrix.
Accuracy:

```
(46+32)/(46+32+10+12)
## [1] 0.78
```

Q2 : K-nearest neighbor

Let's fit a K-nearest neighbor algorithm using `Smarket` dataset which has daily percentage returns for the S&P 500 stock index between 2001 and 2005.

```
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, ~
```

a. Briefly describe what the above set of codes do. Why do we need to split the data into training and test set? *Answer:*

The code creates a variable called “data_Smarket” that takes in the data from Smarket. The `initial_split` function shuffles, stratifies based on Direction variable, and splits the data using the proportion 4 to 5. `Smarket_train` fits a model with certain hyper-parameters on a particular subset of the dataset and `Smarket_test` tests the model on a different subset of the dataset to get an estimate of a final, unbiased assessment of the model's performance

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. *#Answer:**

Smarket_results contains the true labels of the dataset (Direction) and the Direction predicted by the k-nearest neighbor model. The dimension of this dataset is 252×2

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

Smarket_knn_spec <- nearest_neighbor(mode = "classification",
                                     engine = "kkn",
                                     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))
```

c. The following is a confusion matrix from the prediction results from b. Calculate by hand the sensitivity, specificity, accuracy, and positive predictive value of the classifier. *Answer:*

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

```
TP <- 72
TN <- 50
FP <- 71
FN <- 58
n <- TP + TN + FP + FN

(TP+TN)/n # accuracy
## [1] 0.4860558
(TP)/(TP + FN) #sensitivity
## [1] 0.5538462
(TN)/(FP+TN) #specificity
## [1] 0.4132231
(TP)/(TP+FP) #PPV
## [1] 0.5034965
```

Q3 Miscellaneous

(a) **Explain the difference between unsupervised learning and supervised learning.** *Answer:* The main difference between supervised and unsupervised learning: Labeled data. The main distinction between the two approaches is the use of labeled datasets. To put it simply, supervised learning uses labeled input and output data, while an unsupervised learning algorithm does not.

(b) **Is feature scaling required for the K-NN algorithm? Explain with proper justification.** *Answer:* Feature scaling is required in K-NN because the calculation of Euclidean distance is influenced by the feature that has more variability.

(c) **For two runs of K-Mean clustering is it expected to get same clustering results?** *Answer:* No, due to random initializing points, any two runs of K-means clustering is expected to produce different clustering results.

(d) **Is it possible that assignment of observations to clusters does not change between successive iterations in K-Means?** *Answer:* When the K-Means algorithm has reached the local or global minima, it will not alter the assignment of data points to clusters for two successive iterations.

(e) (True/False) **Precision is a useful metric in cases where False Positive is a higher concern than False Negatives. Provide explanations as well.** *Answer:* True, precision tells us how many of the correctly predicted cases actually turned out to be positive.

(f) **Explain how you can use total within cluster sum of squares to find the “best” choice of K in a K-means clustering algorithm.** *Answer:*

Calculate the Within-Cluster-Sum of Squared Errors (WSS) for different values of k, and choose the k for which WSS becomes first starts to diminish. In the plot of WSS-versus-k, this is visible as an elbow.

(g) **Briefly explain why do we preprocess data in k nearest neighbors algorithm.** *Answer:*

KNN is one of the simplest forms of machine learning algorithms mostly used for classification. It classifies the data point on how its neighbor is classified as KNN classifies the new data points based on the similarity measure of the earlier stored data points.

(h) (Multiple Choice) **Given the following models trained using K-NN, the model which could result in underfitting will most likely have the value of K as**

1. 30
2. 5
3. 1

Answer: 30, because a k-NN fit with large number of neighbors smooths out local structure and causes a lack of generalization.

(i) **Does centroid initialization affect K-means algorithm? Explain your answer.** *Answer:* Yes, it does. It is a random initialization, so depending on where it starts, the algorithm may be trapped in a local maxima.

(j) Logistic regression is a machine learning algorithm that is used to predict the probability of a _____. Write your letter choice in the blank.

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

Answer: B