MSIN0094 Second Assignment Answer Sheet

Anonymous Candidate Number: VSXB6

Self-reported word count: 1,482 words

1. Break-Even Response Rate (16pts)

Question 1. Compute the break-even response rate for Tom's targeting campaign based on the cost information given and assign it into a variable named `breakeven_response_rate`. Explain the role of break-even response rate in the focal targeted marketing campaign. (**7 pt**)

[My Answer]:

The break-even response rate indicates the minimum proportion of customers who must subscribe for the targeted marketing campaign to break even. If the actual response rate exceeds the break-even response rate, the campaign is profitable; otherwise, it results in a loss. In this case, the break-even response rate is approximately 0.21, meaning that as long as more than 21% of targeted customers subscribe, this event is profitable and worth executing.

```
# complete the code below.
# you can create additional intermediate variables if needed

# Marketing Expenditure = cost of printing and mailing a marketing offe
r
cost_per_offer <- 1.5

# Contribution Margin Per Unit = discounted subscription fee + profit o
f goods purchased - shipping cost
COGS <- 0.9
shipping_cost <- 4
profit_per_customer <- 6.99 + 40 * (1 - COGS) - shipping_cost

# Break-even Response Rate = Marketing Expenditure / Contribution Margi
n Per Unit
breakeven_response_rate <- cost_per_offer / profit_per_customer

# pls do not modify the codes below
# these are for TAs to check results
print(paste("cost_per_offer is ", cost_per_offer))
```

```
[1] "cost_per_offer is  1.5"
```

```
print(paste("profit_per_customer is", profit_per_customer))
```

```
[1] "profit_per_customer is 6.99"
```

```
print(paste("breakeven_response_rate is", breakeven_response_rate))
```

```
[1] "breakeven_response_rate is 0.214592274678112"
```

Question 2. Compute the ROI of marketing for **blanket marketing** by sending offers to all customers in the `data_full`. (**5 pt**)

• Discuss whether it's profitable for Tom to go ahead with blanket marketing for the remaining customers

[My Answer]:

Based on the ROI and actual response rate, blanket marketing is not profitable to execute. The ROI of blanket marketing is negative, which is -0.86, indicating that the net profit gained from this campaign is lower than its total cost. In addition, the actual response rate is around 0.16, which is lower than the break even response rate of 0.21. It means that the proportion of subscribed customers is not enough to cover the cost of blanket marketing. Therefore, Tom should not go ahead with blanket marketing method.

```
# total 5000 customers
n <- 5000

# total costs of marketing event = cost per offer * total customers
total_costs_of_mailing_blanket <- cost_per_offer * n

# actual response rate based on data in data_full
actual_response_rate <- mean(data_full$subscribe == 'yes')
print(actual_response_rate)
```

```
[1] 0.1676
```

```
# total profit of marketing event = profit per customer * subscribed cu
stomers
total_profit_blanket <- profit_per_customer * n * actual_response_rate

# ROI = total profit of marketing event * response rate / total costs o
f marketing event
ROI_blanket <- (total_profit_blanket * actual_response_rate - total_cos
ts_of_mailing_blanket) / total_costs_of_mailing_blanket

# do not modify the code below
print(paste("total_costs_of_mailing_blanket is ", total_costs_of_mailin
g_blanket))
```

```
[1] "total_costs_of_mailing_blanket is  7500"
```

```
print(paste("total_profit_blanket is ", total_profit_blanket))
```

```
[1] "total_profit_blanket is  5857.62"
```

```
print(paste("ROI_blanket is ", ROI_blanket))
```

```
[1] "ROI_blanket is  -0.8691017184"
```

2. Descriptive Analytics for Segmentation and Targeting: RFM Analysis (24 pts)

Question 3. Use `dplyr` data wrangling tools to compute the RFM variables `recency`, `frequency`, and `monetary_value` based on existing variables in the data. Also create a binary numeric variable `subscribe_yes` which equals 1 if a user subscribes and 0 otherwise. Report the summary statistics AND correlation table of `recency`, `frequency`, `monetary_value`, and `subscribe_yes`. Discuss any notable findings relevant for targeted marketing from the correlation table (**10 pts**)

[My Answer]:

From the correlation table, the correlation between frequency and monetary is moderate positive, which is approximately 0.52, indicating that customers who purchase more frequently tend to spend more on Amazon. Moreover, customers with more recent purchase are more likely to subscribe Amazon Prime, which is around -0.19 of correlation between the variables “recency” and “subscribe_yes”.

```
# create the required variables below
data_RFM <-
  data_full %>%
  mutate(
    # number of days since the last purchase
    recency = last,
    # total number of purchases
    frequency = home + sports + clothes + health + books + digital + to
ys,
    # total spending
    monetary_value = electronics + nonelectronics,
    # subscribe_yes == 1 if subscribe == 'yes', else subscribe_yes == 0
    subscribe_yes = ifelse(subscribe == 'yes', 1, 0)
    )

# report the summary statistics below
pacman::p_load(modelsummary)

data_RFM %>%
  select(recency, frequency, monetary_value, subscribe_yes) %>%
  datasummary_skim()
```

	Unique	Missing Pct.	Mean	SD	Min	Median	Max	Histogram
recency	18	0	11.8	8.0	1.0	11.0	35.0	
frequency	12	0	3.9	3.5	1.0	2.0	12.0	
monetary_value	445	0	210.8	102.6	15.0	213.0	477.0	
subscribe_yes	2	0	0.2	0.4	0.0	0.0	1.0	

```
# report the correlation table below
data_RFM %>%
  select(recency, frequency, monetary_value, subscribe_yes) %>%
  cor()

                    recency    frequency monetary_value subscribe_yes
recency         1.000000000 -0.001123097    0.004861382    -0.1925138
frequency      -0.001123097  1.000000000    0.524403745     0.1521874
monetary_value  0.004861382  0.524403745    1.000000000     0.1078519
subscribe_yes  -0.192513829  0.152187442    0.107851929     1.0000000
```

Question 4. Use R's `dplyr` package to implement the RFM segmentation algorithm described above, dividing customers into 125 R-F-M segments based on their RFM variables. The resulting dataset should have the same rows as `data_RFM` but include the following new variables: `recency_quintile`, `frequency_quintile`, `monetary_quintile`, and `avg_response_rate` (the average response rate for each RFM segment). Report which RFM segment has the highest average response rate? (**8 pts**)

[My Answer]:

The R5-F2-M5 segment has the highest average response rate.

```
# create the RFM segments below
# recency segment
# divide into 5 quantile segments # lower recency -> more valuable -> h
igher quintile -> reverse order
data_RFM <- data_RFM %>%
  mutate(recency_quintile = 6 - ntile(recency, 5))

# frequency segment
data_RFM <- data_RFM %>%
  group_by(recency_quintile) %>%
  # divide into 5 quantile segments
  mutate(frequency_quintile = ntile(frequency, 5)) %>%
  ungroup()

# monetary segment
data_RFM <- data_RFM %>%
```

```r
  group_by(recency_quintile, frequency_quintile) %>%
  # divide into 5 quantile segments
  mutate(monetary_quintile = ntile(monetary_value, 5)) %>%
  ungroup()

# average response rate
data_RFM <- data_RFM %>%
  group_by(recency_quintile, frequency_quintile, monetary_quintile) %>%
  # calculate the mean values of response rate in each segment
  summarise(
    avg_response_rate = mean(subscribe_yes),
    n_each_segment = n()  # how many customers in each segment
  ) %>%
  ungroup()
```

```
`summarise()` has grouped output by 'recency_quintile', 'frequency_quin
tile'.
You can override using the `.groups` argument.
```

```r
# show all the segments and order by (DESC) average response rate
data_RFM %>%
  arrange(desc(avg_response_rate)) %>%
  dplyr::glimpse()
```

```
Rows: 125
Columns: 5
$ recency_quintile   <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 4, 5, 5, 5, 3, 4, 4,
 4, 5, …
$ frequency_quintile <int> 2, 2, 5, 4, 5, 4, 1, 5, 5, 5, 2, 4, 2, 4, 5,
 5, 1, …
$ monetary_quintile  <int> 5, 3, 3, 1, 4, 5, 5, 5, 2, 1, 4, 3, 5, 2, 3,
 4, 3, …
$ avg_response_rate  <dbl> 0.575, 0.550, 0.550, 0.500, 0.500, 0.475, 0.
450, 0.…
$ n_each_segment     <int> 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
40, 40,…
```

```r
# report the RFM segment with highest average response rate below
# observe which RFM segment has the highest response rate
highest_avg_response_rate_segment <- data_RFM %>%
  # set the order of avg_response_rate as DESC order -> higher values s
how at the top
  arrange(desc(avg_response_rate)) %>%
  # only show the first row -> highest average response rate
  slice_head(n = 1)

# do not modify the code below, this is for TA to check the results
highest_avg_response_rate_segment %>%
  dplyr::glimpse()
```

```
Rows: 1
Columns: 5
$ recency_quintile   <dbl> 5
$ frequency_quintile <int> 2
$ monetary_quintile  <int> 5
$ avg_response_rate  <dbl> 0.575
$ n_each_segment     <int> 40
```

Question 5. Among the RFM segments created in the previous step, which segment(s) should Tom target for the marketing campaign (i.e., send marketing offers to all customers in those segments)? Compute the ROI of marketing if Tom conducts targeted marketing to the segment you selected. (**6 pts**)

• Explain the basis for your targeting decision among the RFM segments

[My Answer]:

Based on the calculation of average response rate in each segment in Question 4, Tom should target R5-F2-M5 segment since the average response rate is 0.575, which is the highest among all segments. The ROI of this segment is around 1.6795, meaning every £1 spent on marketing would return about £1.68 in net profit. While some of the RFM segments also exceed the break-even response rate, none of them provide a higher expected return. Therefore, R5–F2–M5 is the optimal target segment.

```
# Write your codes below to compute the ROI for the targeted marketing
based on RFM segments
# choose the segment that has the highest response rate
n_targeted <- highest_avg_response_rate_segment$n_each_segment
avg_response_rate_targeted <- highest_avg_response_rate_segment$avg_res
ponse_rate

# calculate expected numbers of responders using this segment
responders_targeted <- n_targeted * avg_response_rate_targeted

# total costs of mailing (RFM)
total_costs_of_mailing_RFM <- cost_per_offer * n_targeted

# total profit (RFM)
total_profit_RFM <- profit_per_customer * responders_targeted

# compute ROI of RFM
ROI_RFM <- (total_profit_RFM - total_costs_of_mailing_RFM) / total_cost
s_of_mailing_RFM

# do not modify the code below
print(paste("total_costs_of_mailing_RFM is ", total_costs_of_mailing_RF
M))
```

```
[1] "total_costs_of_mailing_RFM is  60"
```

```
print(paste("total_profit_RFM is ", total_profit_RFM))
```

```
[1] "total_profit_RFM is  160.77"
```

```
print(paste("ROI_RFM is ", ROI_RFM))
```

```
[1] "ROI_RFM is  1.6795"
```

3. Unsupervised Learning for Segmentation and Targeting (22 pts)

Question 6. Use the three RFM variables to segment customers using the K-means clustering algorithm. Please use `set.seed(888)` for reproducibility. Specify x and `centers` in the `kmeans()` function, while using the default values for ALL remaining arguments for simplicity. (**8 pts**)

• Explain why we need each data pre-processing steps

[My Answer]:

Firstly, the RFM variables (recency, frequency, monetary_value) are constructed since they are the primary inputs for segmentation, and the variable "subscribe_yes" is created for evaluating cluster results later. Secondly, only RFM variables are kept in "data_kmeans" since K-means only operates on numeric variables. Thirdly, the RFM variables need to be standardized since "monetary_value" is measured in currency and has a larger scale, which might result in biased clustering. Standardisation ensures all three variables contribute equally to clustering.

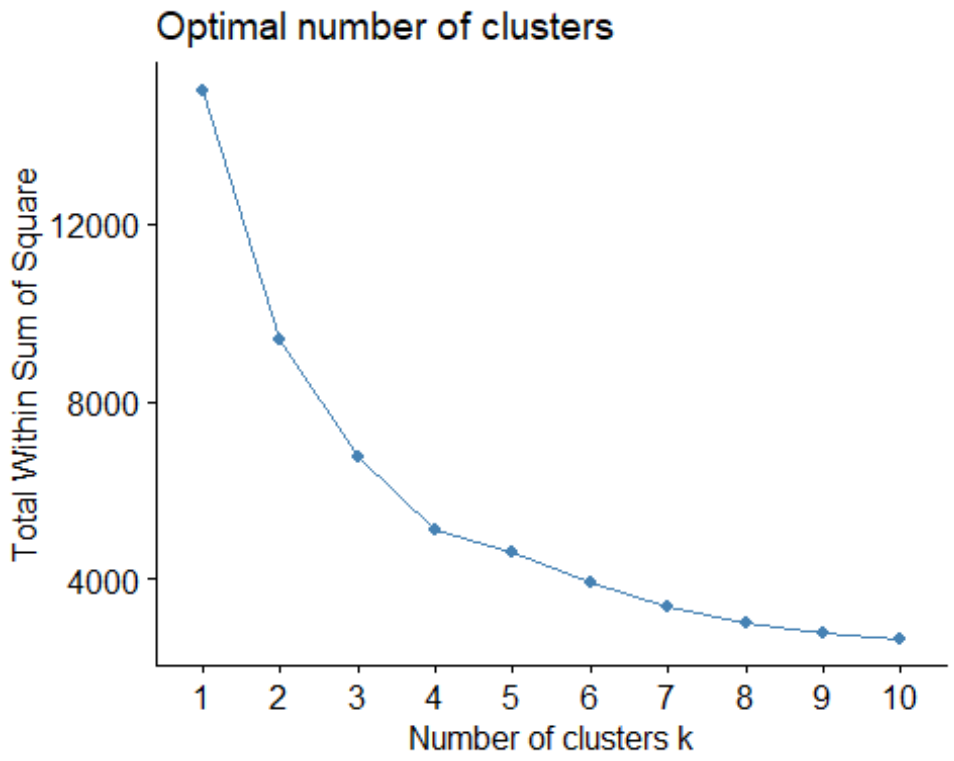
```
# create the RFM variables below
data_RFM_kmeans <- data_full %>%
  mutate(
    # number of days since the last purchase
    recency = last,
    # total number of purchases
    frequency = home + sports + clothes + health + books + digital + to
ys,
    # total spending
    monetary_value = electronics + nonelectronics,
    # divide users to 0 and 1 based on subscribe or not
    subscribe_yes = ifelse(subscribe == 'yes', 1, 0)
  ) %>%
  # select the three variables which would be calculated later
  select(recency, frequency, monetary_value, subscribe_yes)

# Complete the code below to pre-process the data for k-means clusterin
g
```

```r
data_kmeans <- data_RFM_kmeans %>%
  # select the three variables to scale
  select(recency, frequency, monetary_value) %>%
  # scaling
  mutate(across(everything(),scale))

# Determine the optimal number of clusters using the Elbow method below
# do not modify the seed below
set.seed(888)

pacman::p_load(factoextra)
data_kmeans %>%
  fviz_nbclust(kmeans, method = "wss")
```




```r
# implement k-means clustering below

# do not modify seeds
set.seed(888)

# Tom used Elbow method to determine the optimal number of clusters to
be 4 clusters
result_kmeans <- kmeans(data_kmeans, centers = 4)

## My explanation:  Based on the above chart, the elbow is at around nu
mber of cluster=4, so Tom decides to determine the optimal number of cl
usters to be 4 clusters
```

```
# do no modify the code below, this is for TA to check the results

# use broom::tidy() to check the clusters.
pacman::p_load(broom)
result_kmeans %>%
  tidy()
```

```
# A tibble: 4 × 6
  recency frequency monetary_value  size withinss cluster
    <dbl>     <dbl>          <dbl> <int>    <dbl> <fct>
1   1.94     -0.234         -0.138   668    1064. 1
2  -0.178     1.57           0.952  1163    1819. 2
3  -0.355    -0.520          0.429  1575    1057. 3
4  -0.332    -0.536         -1.06   1594    1180. 4
```

Question 7 . Among the segments from k-means, which segment should Tom target for the marketing campaign (i.e., send marketing offers to all customers in that segment)? Tip: `result_kmeans$cluster` returns an R vector that tells us which segment each customer is in, in the same order as the rows in `data_kmeans`. (**6 pts**)

- **Explain the basis for your targeting decision**

[My Answer]:

Firstly, based on the cluster summary below, the cluster 2 has the highest average response rate, which is approximately 0.26. Secondly, the cluster 2 is the only cluster whose average response rate exceeds the break-even threshold, indicating that it is the only cluster that is profitable enough to cover total costs of the marketing event. Therefore, Tom should target cluster 2 for marketing campaign.

```
# Write your codes here to reason which segment to target
# Show numbers from your coding to support your argument.

# mutate cluster and subscribe_yes to data_kmeans
data_kmeans <- data_kmeans %>%
  mutate(cluster = result_kmeans$cluster,
         subscribe_yes=data_RFM_kmeans$subscribe_yes)

# summarise the average response rate in each cluster
cluster_summary <- data_kmeans %>%
  group_by(cluster) %>%
  summarise(
    n = n(),
    average_response = mean(subscribe_yes),
  )

# show the cluster summary
cluster_summary
```

```
# A tibble: 4 × 3
  cluster     n average_response
    <int> <int>            <dbl>
1       1   668           0.0584
2       2  1163           0.262
3       3  1575           0.177
4       4  1594           0.135
```

```r
# Compute the ROI of marketing if Tom conducts k-means targeted marketi
ng to the segment you selected

# the number of customers in cluster=2
n_targeted_kmeans <- data_kmeans %>%
  # filter cluster=2
  filter(cluster == 2) %>%
  # row counts
  nrow()

# average response rate in cluster 2
avg_response_kmeans <- data_kmeans %>%
  # filter cluster=2
  filter(cluster == 2) %>%
  # summarise the average response rate
  summarise(avg_response = mean(subscribe_yes)) %>%
  # pull the avg_response vector
  pull(avg_response)

# expected responders
expected_responders_kmeans <- n_targeted_kmeans * avg_response_kmeans

# total costs of marketing event
total_costs_of_mailing_kmeans <- cost_per_offer * n_targeted_kmeans

# total profit
total_profit_kmeans <- profit_per_customer * expected_responders_kmeans

# calculate ROI of K-means method
ROI_kmeans <- (total_profit_kmeans - total_costs_of_mailing_kmeans) / t
otal_costs_of_mailing_kmeans

# do not modify the code below

print(paste("ROI_kmeans is ", ROI_kmeans))
```

```
[1] "ROI_kmeans is  0.222098022355975"
```

Question 8. Discuss the pros and cons of using K-means clustering to conduct segmentation and targeting for this dataset. (**4pts**, 200 words)

[My Answer]:

Pros:

1. K-means clustering is simple, fast, and easy to apply, which makes it a good starting point for grouping customers.
2. After clustering, the cluster centres make it easy to interpret each segment's characteristics and decide which group is more suitable for targeting.

Cons:

1. The number of clusters (k) needs to be chosen before running the k-means model, and choosing the wrong k may lead to meaningless segments.
2. Since the algorithm uses Euclidean distance, all variables must be scaled properly, or else variables with larger ranges will dominate the clustering results.

(refer to reference 1)

4. Decision Tree Analysis (24 pts)

Question 9. Complete the following data preparation tasks before training a decision tree model. (**10 pts**)

• Do we need to scale the predictor variables when training a decision tree model? Explain your answer.

[My Answer]:

We don't need to scale the predictor variables when training a decision tree model since decision trees split data based on individual features, such as recency and frequency. The quality of each split is evaluated using impurity measures such as Gini Impurity and Information Gain (based on entropy). The model does not rely on distance-based calculations; therefore, it is not necessary to do scaling to standardise the variables into smaller numbers, as whether the variable scale is 1-10 or 1-1000 would not change which feature is chosen or how the tree splits.

• Explain why we need to split data into a training set and a test set for supervised learning tasks, rather than using the full dataset for both training and evaluation.

[My Answer]:

We need to split the data into a training set and a test set to evaluate how well the supervised learning model performs. The training set is used to train machine learning models, while the test set is used to evaluate the prediction accuracy by comparing the predicted outcomes versus the actual outcomes. If we use the full dataset for both training and evaluation, the model is likely to overfit. That is, the predictive model learns from one single training dataset too well, then it may be too

rigid and specialised and thus have a higher chance of failing to make predictions for another dataset accurately. Overfitting leads to low bias but high variance, which is not the ideal result as we expect to have higher prediction accuracy for new data.

```
# set seed, please do not change the seed
set.seed(1314520)

# subscribe is a character, need to convert it into 1,0 variable
data_tree <- data_full %>%
    mutate(
    # number of days since the last purchase
    recency = last,
    # total number of purchases
    frequency = home + sports + clothes + health + books + digital + to
ys,
    # total spending
    monetary_value = electronics + nonelectronics,
    # convert subscribe into binary variable
    subscribe_yes = ifelse(subscribe == "yes", 1, 0)
  )

# use nrow() to count the number of rows in data_tree
n_rows_data_full <- nrow(data_tree)

# sample the index for training data
# use sample() to randomly draw row index from data_full
training_set_index <- sample(
  # draw from 1 until 5000
  x = 1:n_rows_data_full,
  # size is 70% of 5000
  size = 0.7 * n_rows_data_full,
  # do not sample with replacement
  replace = FALSE
)

# create data_training and data_test
# data for training
data_training <- data_tree %>%
    slice(training_set_index)
# data for testing
data_test <- data_tree %>%
    slice(-training_set_index)

# Do not modify this code block.
# This is to print out first 5 customers
training_set_index[1:5]
```

```
[1] 4439 4646 3620 3803   43
```

Question 10. Complete the following steps to train a decision tree model (**10 pts**)

• Train a decision tree model named tree1 on data_training using the 3 RFM variables as predictors. Explain the arguments of the function including formula, data, and method.

[My Answer]:

```
# 1. Formula: The formula “subscribe_yes ~ recency + frequency + moneta
ry_value” is used to predict the binary outcome variable “subscribe_yes”
 based on the explanatory RFM variables “recency”, “frequency”, and “mo
netary_value”.

# 2. Data: The “data” argument here indicates which training dataset to
 train the model. I use “data_training” dataset in this case to ensure
the model learns only from the training data rather than the full datas
et.

# 3. Method: The “method” argument specifies the type of decision tree
to build. The common methods include “class” and “anova”, which are for
 classification and regression tasks respectively. In this case, though
 “subscribe_yes” is a binary variable, I still use the “anova” method s
ince the outcome can be interpreted as an estimated response probabilit
y. It will be easier to compare with the break-even response rate and c
omputing ROI later.
```

• Visualize tree1. Based on the visualization, explain how tree1 makes the first initial split from the root node on the training data of past customers

[My Answer]:

The decision tree will try to split customers into 2 groups based on each unique value of each variable, and see which split can lead to customers being most different and produce the largest reduction in within-node variance in terms of the outcome “subscribe_yes”. In this case, decision tree finds that the recency is the best variable and 10 is the best cut-off. Therefore, decision tree splits customers into 2 groups based on recency with a threshold of 10 days, which most effectively differentiates customers with different average subscription probabilities.

Therefore, the customers in training data are split into two groups:

1. Customers with recency >= 10
2. Customers with recency < 10

• Explain how tree1 predicts the response probability for a future new customer given the new customer’s characteristics

[My Answer]:

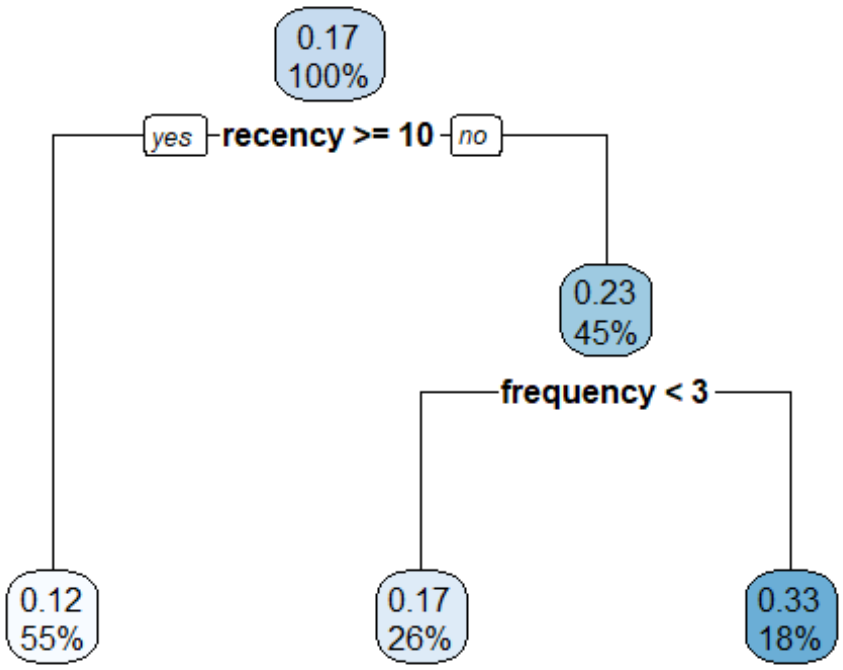
For a new customer, we can follow the tree from the root node to the leaf node to predict the outcome. In this case, tree1 model splits customers based on RFM values. Starting from the root node, the model splits the customer based on if the recency is bigger or equal to 10, then splits again based on whether customer's frequency is lower than 3, until reaching the terminal leaf node.

For example, if a new customer has recency <10 and frequency >= 3, then this customer will be split to the leaf with a predicted probability of 0.33, meaning that he has 33% of probability to subscribe.

```
pacman::p_load(rpart, rpart.plot)

# train model tree1 below
tree1 <- rpart(
  # RFM variables
  formula = subscribe_yes ~ recency + frequency + monetary_value,
  # use data_training to train data
  data = data_training,
  # choose anova method
  method = "anova"
)

# visualize tree1 below
rpart.plot(tree1)
```



Question 11. Compute the ROI from targeted marketing using `tree1`.

• Compare the ROI of k-means and tree1and report which model is better. Discuss why that model can achieve a higher ROI?

[My Answer]:

Based on the calculations, the ROI of the k-means model is approximately 0.22, while the ROI of the tree1 decision tree model is around 0.67. Apparently, tree1 model achieves a higher ROI than k-means model.

The main reason that tree1 model can achieve higher ROI is because the decision tree model predicts the response probability for each individual customer, which can split customers more accurately. On the other hand, k-means model separate customers into clusters based on similarity in RFM values rather than individual behaviour. Even in the same cluster, the response rate of each customer may still be different, leading to inaccurate predictions and lower ROI.

Therefore, tree1 is the better model since it provides more accurate probability and targets the customers more effectively, resulting in a higher ROI.

```
# use predict() to make prediction on the test set
prediction_from_decision_tree <- predict(tree1, data_test)

# mutate a new column in data_test for the predicted probability
data_test <- data_test %>%
  mutate(predicted_prob_decisiontree = prediction_from_decision_tree)

# mutate a new binary indicator for whether to target a customer based
on predicted prob from decision tree model
data_test <- data_test %>%
  mutate(is_target_decisiontree = ifelse(predicted_prob_decisiontree >
breakeven_response_rate, 1, 0))

# number of targeted customers
target_decision_tree <- data_test %>%
  filter(is_target_decisiontree == 1)
n_targeted_decision_tree <- nrow(target_decision_tree)

# Compute the ROI for tree1 below.
# total costs of mailing
total_costs_of_mailing_decisiontree1 <- cost_per_offer * nrow(target_de
cision_tree)

# total profit
total_profit_decisiontree1 <- profit_per_customer * sum(target_decision
_tree$subscribe_yes)

# ROI calculation
ROI_decisiontree1 <- (total_profit_decisiontree1 - total_costs_of_maili
ng_decisiontree1) / total_costs_of_mailing_decisiontree1
```

```
# do not modify the code below
print(paste("ROI_decisiontree1 is ", ROI_decisiontree1))
```

```
[1] "ROI_decisiontree1 is  0.6776"
```

5. Random Forest (20 pts)

Question 12. Train a random forest model with the same set of predictors as `tree1` (**4 pts**)

```
pacman::p_load(ranger)

# train the random forest model below
# set seed 888
set.seed(888)

# random forest model
# The formula “subscribe_yes ~ recency + frequency + monetary_value” is
 used to predict the binary outcome variable “subscribe_yes” based on t
he explanatory RFM variables “recency”, “frequency”, and “monetary_valu
e”. The symbol “~” separates the outcome variable and the explanatory v
ariables in supervised learning.

randomforest <- ranger(
  # RFM variables: recency, frequency, monetary_value
  # outcomme variables: subscribe_yes
  formula = subscribe_yes ~ recency + frequency + monetary_value,
  # use data_training dataset to train data -> ensure the model learns
only from the training data rather than the full dataset
  data = data_training,
  # export the predicted probability, not 0/1 class label
  probability = TRUE,
  # set 2000 trees to train
  num.trees = 2000
)

# make prediction on the test set, which returns a model prediction obj
ect
prediction_from_randomforest <- predict(randomforest, data_test)

# mutate a new column in data_test for the predicted probability from r
andom forest
# the prediction_from_randomforest$predictions gives a matrix
# so we need to the take the second column using [,2]
data_test <- data_test %>%
    mutate(
      predicted_prob_randomforest = prediction_from_randomforest$predic
tions[, 2]
```

```
    )

# mutate a new binary indicator for whether to target a customer based
on predicted prob from random forest model
data_test <- data_test %>%
    mutate(
      is_target_randomforest = ifelse(predicted_prob_randomforest > bre
akeven_response_rate, 1, 0)
    )

# number of targeted customers
target_randomforest <- data_test %>%
  filter(is_target_randomforest == 1)

# total marketing costs
total_costs_of_mailing_randomforest <- cost_per_offer * nrow(target_ran
domforest)

# total profits from responding customers
total_profit_randomforest <- profit_per_customer * sum(target_randomfor
est$subscribe_yes)

# compute the ROI for random forest below below
ROI_randomforest <- (total_profit_randomforest - total_costs_of_mailing
_randomforest) / total_costs_of_mailing_randomforest

# do not modify the code below

print(paste("ROI_randomforest is ", ROI_randomforest))
```

```
[1] "ROI_randomforest is  0.48064039408867"
```

Question 13. Based on the results of different models you have obtained in this assignment so far, discuss the fundamental tradeoffs of supervised learning (**8 pts**, 300 words)

• The definitions of the tradeoffs

[My Answer]:

In supervised learning, there are several tradeoffs that affect the performance of a model.

One of the most fundamental tradeoffs is the bias-variance tradeoff. Increasing model complexity typically decreases bias but increases variance, leading to overfitting. On the other hand, decreasing model complexity typically increases bias but decreases variance, leading to underfitting.

Accuracy-interpretability tradeoff is also the tradeoff that is widely discussed. Simpler models are easier to interpret but typically give lower accuracy, while more complex models can give better prediction accuracy, but results are harder to interpret.

Therefore, choosing an appropriate supervised model requires balancing complexity, variance, accuracy, and interpretability. A good model should be neither too simple nor too complex but should achieve a balance that generalizes well with new data.

• Use concrete examples from previous models trained in this assignment to illustrate each tradeoff.

[My Answer]:

The tradeoffs mentioned above are clearly demonstrated in the models trained in this assignment, and the results also show that more complex models do not always guarantee a higher ROI.

In this assignment, the ROI of random forest model (0.48) is lower than the ROI of decision tree model (0.67), which is uncommon since random forests typically achieve better predictive performance than a decision tree.

This happened because the random forest produces more conservative probability estimates, causing fewer customers to exceed the break-even response rate threshold. Thus, fewer high-value customers were targeted, resulting in a lower ROI. In contrast, the decision tree splits customers at first based on recency, which is the segment with higher response, and more customers exceed the break-even response rate threshold.

This case reflects the bias-variance tradeoff. Although the random forest model usually provides more stable prediction with lower variance, in this specific case, the simpler model generated better results.

Question 14. Discuss at least two recommendations for Tom to improve the performance of the random forest model based without collecting more data. (**8pts**, 250 words)

[My Answer]:

1. Recommendation 1: Adjust the “mtry” parameter

   The “mtry” is the number of variables to randomly sample as candidates at each split. Adjusting this parameter can reduce the correlation between trees and make the model more diverse, especially when the trees are too similar and lead to high variance or overfitting. Currently, since Tom uses all three RFM variables to split customers, he can experiment with smaller values of “mtry” such as 1 or 2 to test if it can increase the diversity in each

split and select the value that has lowest prediction error. (refer to reference 1)

2. Recommendation 2: Adjust the “nodesize” parameter

   The “nodesize” is the minimum number of samples within the terminal nodes. A smaller minimum node size allows deeper, more complex trees, while a larger minimum node size produces shallower, simpler trees. However, small leaf nodes may lead to overfitting. Therefore, by increasing the value of “min.node.size” parameter, Tom could reduce variance and prevent overfitting. This leads to more stable predictions of subscription probability, which can result in creating higher ROI. (refer to reference 1)

Reference

- Reference 1: UC Business Analytics R Programming Guide, University of Cincinnati