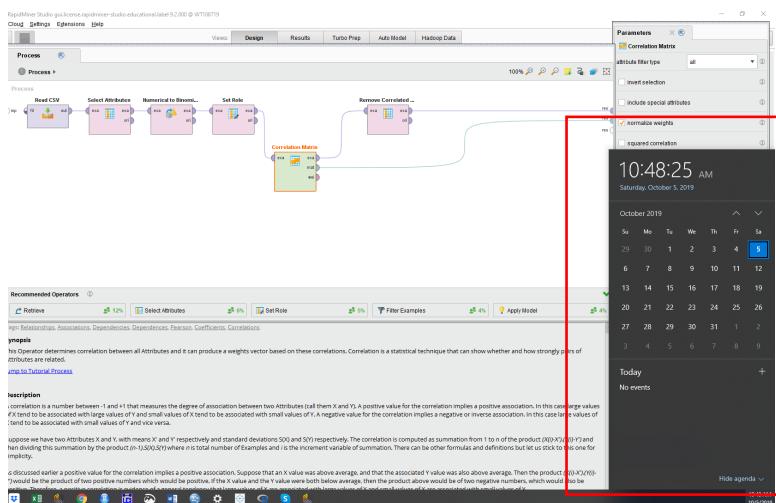


# CIDM 6355 Data Mining Methods Exam 1 Part 2 Submission

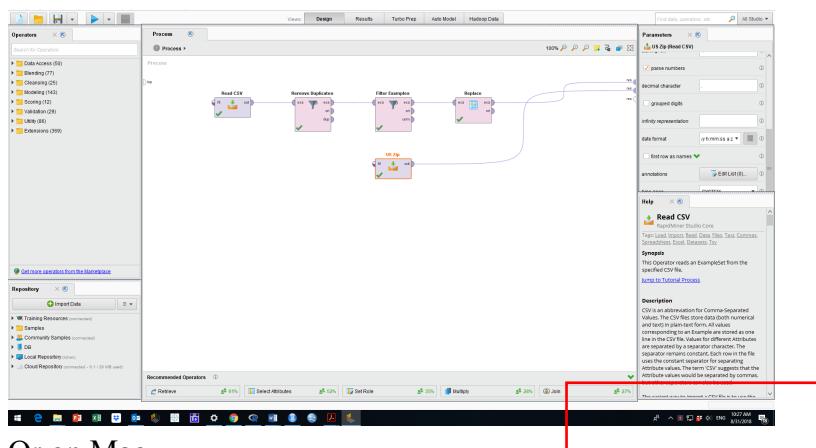
(50 points in total; due 11:59 pm CDT, October 13, 2024)

This exam is open book, open slides, and open notes, but you are not allowed to collaborate nor discuss with anyone else. Sharing your screenshots, RM processes, R script, or answers with other students are considered as cheating for this exam. Should you have any question or unclarity, please contact with the instructor. Please put all your deliverables in a word document and submit it to WT class before the deadline; Make sure that all your screenshots include dates and time [see the examples as below]; otherwise, a penalty of 50% of your grade will be applied. Please type your name below to indicate whether or not you have understood and complied with such requirements in this exam.

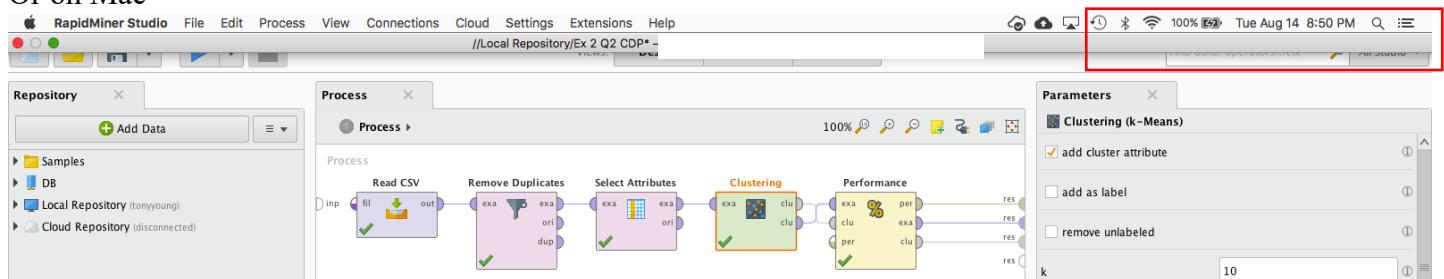
Your name (First Last): Sera Hill



Or



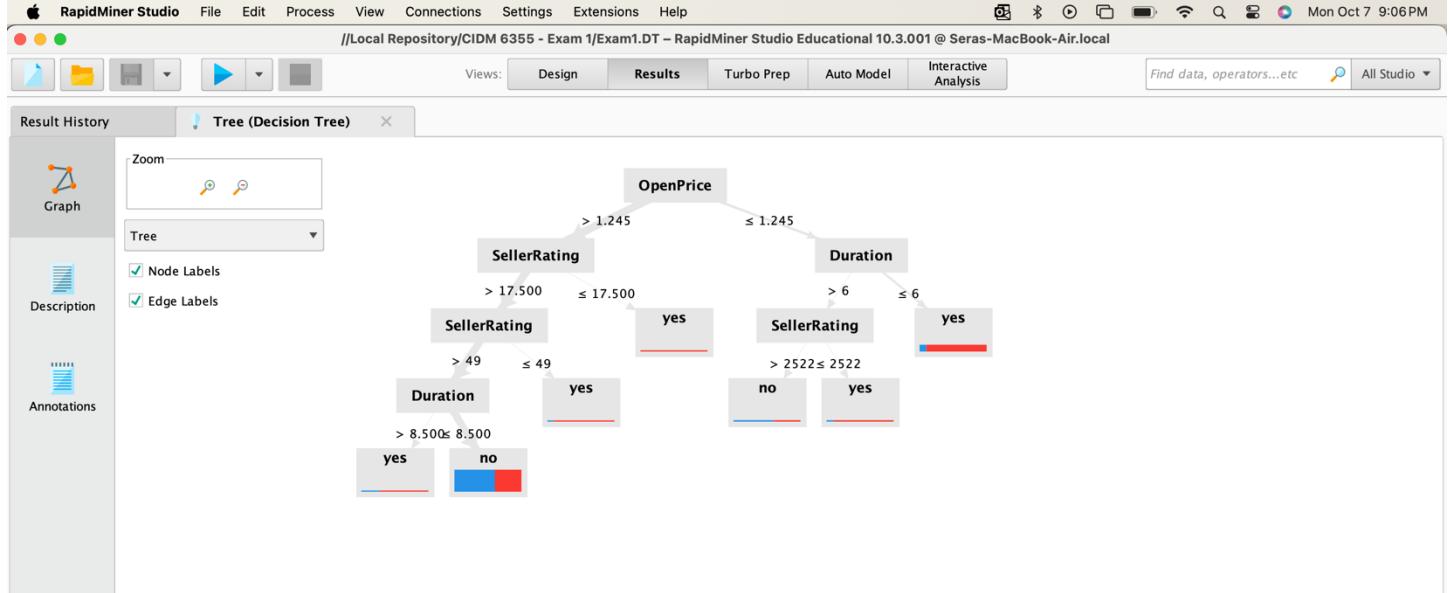
Or on Mac



**PLEASE MAKE SURE THAT YOU TYPE YOUR NAME AT THE FIRST PAGE; OTHERWISE, YOUR SUBMISSIN WILL NOT BE GRADED AND A ZERO POINT WILL BE ASSIGNED.**

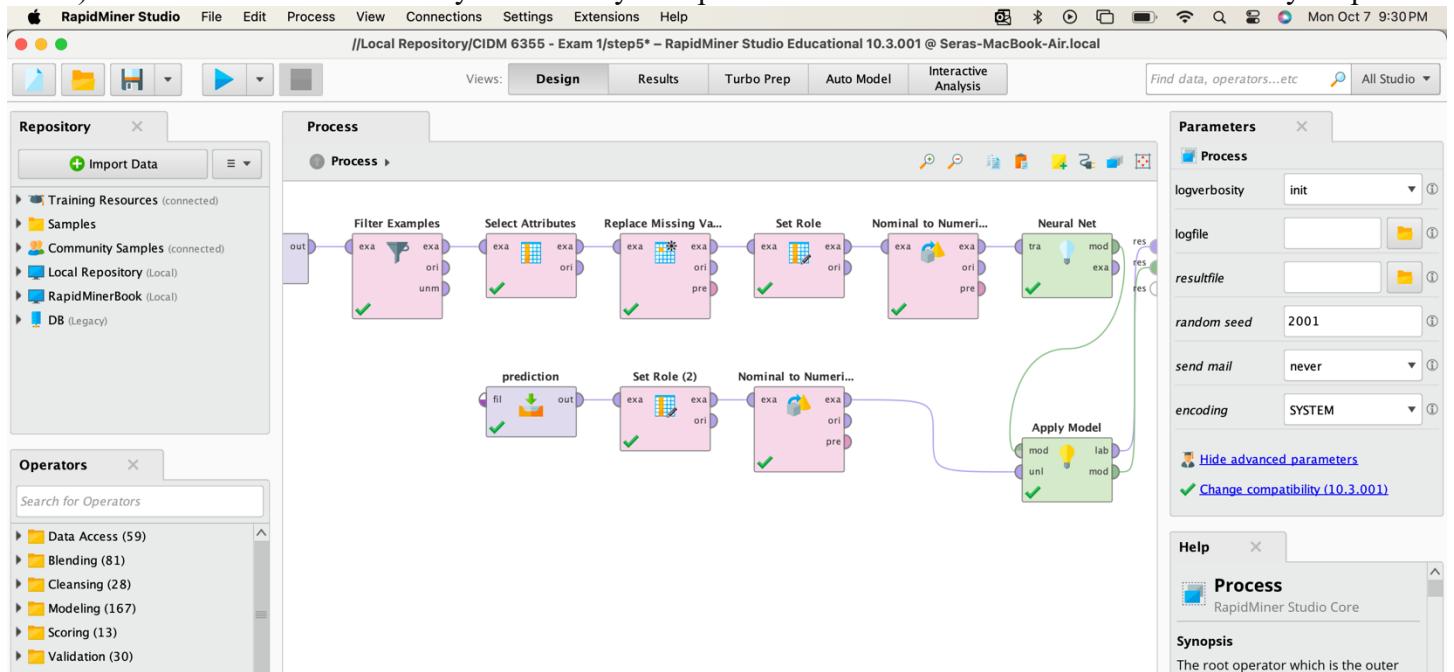
2 Screenshots in RM (6 points for each: 3 pts for your screenshot and 3 pts for your description/discussion).

Screenshot 1 with description (**6 pts**): A screenshot of your decision tree graph with date and time at Step 2.3 and briefly describe your model. Your description must include root node, split nodes, and leaf nodes.



The root node in this Decision Tree is “OpenPrice”. It has 5 split nodes, and 7 leaf nodes.

Screenshot 2 with your discussion (**6 pts**): A screenshot of your RapidMiner Process (the flow chart in your design mode) with date and time and briefly discuss why the operator Nominal to Numerical must be used in your process.



The Nominal to Numerical operator is necessary for this model to convert the categorical variables into numerical dummy variable values because Neural Networks are mathematical in nature and can't directly process non-numerical data.

5 Screenshots in R: Each screenshot must include complete and correct code description or comments (20 pts in total and 4 points for each).

Screenshot 3: A screenshot of your R codes with date and time to show how you import and prepare the data for modeling and prediction, that is, Steps 6.1-6.3.

The screenshot shows the RStudio interface with the following details:

- Top Bar:** RStudio, File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, Help, Thu Oct 10 8:38PM
- Project:** Project: (None)
- Console Tab:** R 4.3.1 · ~ /
- Code Area:** The code block contains R script for training and predicting datasets, handling missing data, and summarizing results.
- Output Area:** The console output shows the structure of the datasets, summary statistics, and the final dataset after handling missing values.

```
1 # import training dataset
2 Exam1 <- read.csv(file.choose(), header = T, stringsAsFactors = T)
3 # import prediction dataset
4 Exam1_predict <- read.csv(file.choose(), header = T, stringsAsFactors = T)
5 # view structure of training and prediction datasets
6 str(Exam1)
7 str(Exam1_predict)
8 # only select records with US in the attribute currency
9 Exam1_2 <- Exam1[Exam1$Currency != 'GBP',]
10 # check whether GBP records are removed and missing data issues
11 summary(Exam1_2)
12 str(Exam1_2)
13 # remove the attribute Currency with only one level
14 Exam1New <- subset(Exam1_2, select = -Currency)
15 #check whether the attribute is removed
16 str(Exam1New)
17 # replace the missing data in OpenPrice using the minimum of OpenPrice
18 Exam1New$OpenPrice[which(is.na(Exam1New$OpenPrice))] <- min(Exam1New$OpenPrice)
19 # check if missing values are replaced
20 summary(Exam1New)
21
22
```

```
Music/Movie/Game :273   GBP: 0   Min. : 0   Min. : 1.00   Fri:204   Min. : 0.01   no :621
Toys/Hobbies    :181   US :1292   1st Qu.: 993   1st Qu.: 5.00   Mon:384   1st Qu.: 2.00   yes:671
Collectibles    :163   Median : 2577   Median : 7.00   Sat:303   Median : 6.46
Automotive      :155   Mean   : 4670   Mean   : 5.76   Sun:257   Mean   : 16.86
Antique/Art/Craft:152   3rd Qu.: 3613   3rd Qu.: 7.00   Thu: 16   3rd Qu.: 12.99
SportingGoods   : 85   Max.   : 37727   Max.   :10.00   Tue: 87   Max.   :999.00
(Other)          :283   NA's   :        NA's   :        Wed: 41   NA's   :8
```

```
> str(Exam1_2)
'data.frame': 1292 obs. of 7 variables:
 $ Category  : Factor w/ 18 levels "Antique/Art/Craft",...: 14 14 14 14 14 14 14 14 14 14 ...
 $ Currency   : Factor w/ 2 levels "GBP","US": 2 2 2 2 2 2 2 2 2 2 ...
 $ SellerRating: int 3249 3249 3249 3249 3249 3249 3249 3249 3249 3249 ...
 $ Duration   : int 5 5 5 5 5 5 5 5 5 5 ...
 $ endDay     : Factor w/ 7 levels "Fri","Mon","Sat",...: 2 2 2 2 5 5 5 5 5 5 ...
 $ OpenPrice   : num 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
 $ Competitive : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
> Exam1New <- subset(Exam1_2, select = -Currency)
> str(Exam1New)
'data.frame': 1292 obs. of 6 variables:
 $ Category  : Factor w/ 18 levels "Antique/Art/Craft",...: 14 14 14 14 14 14 14 14 14 14 ...
 $ SellerRating: int 3249 3249 3249 3249 3249 3249 3249 3249 3249 3249 ...
 $ Duration   : int 5 5 5 5 5 5 5 5 5 5 ...
 $ endDay     : Factor w/ 7 levels "Fri","Mon","Sat",...: 2 2 2 2 5 5 5 5 5 5 ...
 $ OpenPrice   : num 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 ...
 $ Competitive : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
> Exam1New$OpenPrice[which(is.na(Exam1New$OpenPrice))] <- min(Exam1New$OpenPrice, na.rm=TRUE)
> summary(Exam1New)
   Category   SellerRating   Duration   endDay   OpenPrice   Competitive 
Music/Movie/Game :273   Min. : 0   Min. : 1.00   Fri:204   Min. : 0.01   no :621
Toys/Hobbies    :181   1st Qu.: 993   1st Qu.: 5.00   Mon:384   1st Qu.: 2.00   yes:671
Collectibles    :163   Median : 2577   Median : 7.00   Sat:303   Median : 6.46
Automotive      :155   Mean   : 4670   Mean   : 5.76   Sun:257   Mean   : 16.86
Antique/Art/Craft:152   3rd Qu.: 3613   3rd Qu.: 7.00   Thu: 16   3rd Qu.: 12.99
SportingGoods   : 85   Max.   : 37727   Max.   :10.00   Tue: 87   Max.   :999.00
(Other)          :283   NA's   :        NA's   :        Wed: 41   NA's   :8
```

Screenshot 4: A screenshot of your R codes with date and time to show Step 6.4.1-6.4.3. Requirements: your screenshot must clearly include all the R codes for your decision tree model and the output of 6.4.3.

The screenshot shows the RStudio interface with the following details:

- Top Bar:** RStudio, File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, Help.
- Project Bar:** Project: (None).
- Code Editor:** Untitled1.R, showing R code for data cleaning, reading CSV files, creating a new dataset, and building a decision tree (DT) using the party package.
- Console:** Displays the output of the R code, including the structure of the decision tree, node splits, and final statistics.
- Output:** Shows the summary of the decision tree, including levels and yes/no counts.
- Bottom Bar:** Environment, History, Connections, Tutorial, Files, Plots, Packages, Help, Viewer, Presentation.

```

1 Exam1 <- read.csv(file.choose(), header = T, stringsAsFactors = T)
2 Exam1_predict <- read.csv(file.choose(), header = T, stringsAsFactors = T)
3 # view structure of training and prediction datasets
4 str(Exam1)
5 str(Exam1_predict)
6 Exam1_2 <- Exam1[Exam1$Currency != 'GBP',]
7 # check whether GBP records are removed and missing data issues
8 summary(Exam1_2)
9 str(Exam1_2)
10 # remove the attribute Currency with only one level
11 Exam1New <- subset(Exam1_2, select = -Currency)
12 #check whether the attribute is removed
13 str(Exam1New)
14 # replace the missing data in OpenPrice using the minimum of OpenPrice
15 Exam1New[OpenPrice[which(is.na(Exam1New$OpenPrice))]] <- min(Exam1New$OpenPrice, na.rm=T)
16 # check if missing values are replaced
17 summary(Exam1New)
18 # install library 'party'
19 install.packages('party')
20 # invoke the party library
21 library(party)
22 # build the decision tree model and store it in DT - no pruning arguments
23 DT <- ctree(formula = Competitive~, data = Exam1New)
24 # view the properties of the decision tree: DT
25 DT
26 # generate the decision tree graph for DT
27 plot(DT)
28 # apply the model for prediction and store it in R_DT
29 R_DT <- predict(DT, Exam1_predict)
30 # view properties of the decision tree for the prediction dataset
31 R_DT
32 # view summary of prediction dataset to quickly see how many values are "yes" and how many are "no"
33 summary(R_DT)
34 R_DT
35 # view summary of prediction dataset to quickly see how many values are "yes" and how many are "no"
36 summary(R_DT)
37
38
37:1 (Top Level) : R Script

```

Screenshot 5: A screenshot of your R codes with date and time to show Step 6.5.1-6.5.3. Requirements: your screenshot must clearly include all the R codes for your NB model and the output of 6.5.3.

RStudio

Untitled1\* >

```

17 # replace the missing data in openPrice using the minimum of OpenPrice
18 Exam1New$OpenPrice[which(is.na(Exam1New$OpenPrice))] <- min(Exam1New$OpenPrice, na.rm=TRUE)
19 # check if missing values are replaced
20 summary(Exam1New)
21 # install library 'party'
22 install.packages('party')
23 # invoke the party library
24 library(party)
25 # build the decision tree model and store it in DT - no pruning arguments
26 DT <- ctree(formula = Competitive ~ ., data = Exam1New)
27 # view the properties of the decision tree: DT
28 DT
29 # generate the decision tree graph for DT
30 plot(DT)
31 # apply the model for prediction and store it in R_DT
32 R_DT <- predict(DT, Exam1_predict)
33 # view properties of the decision tree for the prediction dataset
34 R_DT
35 # view summary of prediction dataset to quickly see how many values are "yes" and how many are "no"
36 summary(R_DT)
37 # create naive bayes model using e1071 library
38 # install library "e1071"
39 install.packages('e1071')
40 # invoke e1071 library
41 library(e1071)
42 # build naieve bayes model
43 NB <- naiveBayes(Competitive ~ ., data = Exam1New)
44 # view NB model
45 NB
46 # apply the model for prediction and store it in R_NB
47 R_NB <- predict(NB, Exam1_predict)
48 # view NB model for the prediction dataset
49 R_NB
50 # view summary of prediction dataset to see how many values are "yes" and how many are "no"
51 summary(R_NB)
52

```

50:34 (Top Level) : R Script

Environment History Connections Tutorial

Console Terminal Background Jobs

```

R 4.3.1 · ~/r
Y Home/Garden Jewelry Music/Movie/Game Photography Pottery/Glass SportingGoods
no 0.025764895 0.057971014 0.161030596 0.003220612 0.017713366 0.024154589
yes 0.020864382 0.017883756 0.257824143 0.011922504 0.007451565 0.104321908
Category
Y Toys/Hobbies
no 0.144927536
yes 0.135618480

SellerRating
Y [,1] [,2]
no 4686.155 6470.358
yes 4655.298 7508.914

Duration
Y [,1] [,2]
no 5.835749 1.650633
yes 5.690015 1.587038

endDay
Y Fri Mon Sat Sun Thu Tue Wed
no 0.19806763 0.21256039 0.29307568 0.22222222 0.01127214 0.06280193 0.00000000
yes 0.12071535 0.37555887 0.18032787 0.17734724 0.01341282 0.07153502 0.06110283

OpenPrice
Y [,1] [,2]
no 21.85427 55.81729
yes 12.02918 35.89468

> R_NB <- predict(NB, Exam1_predict)
> R_NB
[1] no yes yes yes no yes no no no yes yes no no yes yes no yes yes no
Levels: no yes
> summary(R_NB)
no yes
10 10
>

```

Files Plots Packages Help Viewer Presentation

Screenshot 6: A screenshot of your R codes with date and time to show Step 6.6.1-6.6.4. Requirements: your screenshot must clearly include all the R codes for your logistic regression model and the output of 6.6.4.

RStudio

Untitled1\* >

```

17 # view properties of the decision tree: DT
28 DT
29 # generate the decision tree graph for DT
30 plot(DT)
31 # apply the model for prediction and store it in R_DT
32 R_DT <- predict(DT, Exam1_predict)
33 # view properties of the decision tree for the prediction dataset
34 R_DT
35 # view summary of prediction dataset to quickly see how many values are "yes" and how many are "no"
36 summary(R_DT)
37 # create naive bayes model using e1071 library
38 # install library "e1071"
39 install.packages('e1071')
40 # invoke e1071 library
41 library(e1071)
42 # build naieve bayes model
43 NB <- naiveBayes(Competitive ~ ., data = Exam1New)
44 # view NB model
45 NB
46 # apply the model for prediction and store it in R_NB
47 R_NB <- predict(NB, Exam1_predict)
48 # view NB model for the prediction dataset
49 R_NB
50 # view summary of prediction dataset to see how many values are "yes" and how many are "no"
51 summary(R_NB)
52 # build a logistic regression model using glm function
53 # create LR model and store it in LR
54 LR <- glm(Competitive ~ ., family = "binomial", data = Exam1New)
55 # view the summary of the LR model
56 summary(LR)
57 # apply model to the prediction dataset and store it in R_LRP
58 R_LRP <- predict(LR, Exam1_predict, type="response")
59 # convert the probabilities to prediction class and then convert it to a factor
60 R_LRP <- as.factor(ifelse(R_LRP > 0.5, "yes", "no"))
61 # view summary of prediction
62 summary(R_LRP)
62:1 (Top Level) : R Script
```

52:1 (Top Level) :

Environment History Connections Tutorial

Console Terminal Background Jobs

```

R 4.3.1 · ~/r
CategoryElectronics 4.905e+01 5.576e-01 0.880 0.379067
CategoryEverythingElse -2.029e+00 6.823e-01 -2.974 0.002939 **
CategoryHealth/Beauty -1.802e+00 3.862e-01 -4.667 3.05e-06 ***
CategoryHome/Garden -4.780e-01 4.289e-01 -1.115 0.264994
CategoryJewelry -1.175e+00 4.001e-01 -2.935 0.003332 **
CategoryMusic/Movie/Game -3.027e-01 2.389e-01 -1.267 0.205123
CategoryPhotography 9.672e-01 8.247e-01 1.173 0.240865
CategoryPottery/Glass -6.872e-01 5.902e-01 -1.164 0.244308
CategorySportingGoods 1.561e+00 3.824e-01 4.083 4.45e-05 ***
CategoryToys/Hobbies 5.611e-02 2.506e-01 0.224 0.822828
SellerRating -5.093e-05 1.206e-05 -4.222 2.42e-05 ***
Duration -7.143e-02 4.264e-02 -1.675 0.093922 .
endDayMon 1.017e+00 2.103e-01 4.838 1.31e-06 ***
endDaySat -2.591e-01 2.163e-01 -1.198 0.231018
endDaySun 3.468e-02 2.172e-01 0.160 0.873167
endDayThu 4.505e-01 5.530e-01 0.815 0.415352
endDayTue 5.421e-01 2.850e-01 1.902 0.057188 .
endDayWed 1.687e+01 3.699e-02 0.046 0.963623
OpenPrice -2.199e-03 1.948e-03 -1.129 0.259048
---
Signif. codes: 0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1789.2 on 1291 degrees of freedom
Residual deviance: 1522.9 on 1265 degrees of freedom
AIC: 1576.9

Number of Fisher Scoring iterations: 15

> R_LRP <- predict(LR, Exam1_predict, type="response")
> R_LR <- as.factor(ifelse(R_LRP > 0.5, "yes", "no"))
> summary(R_LR)
no yes
10 10
>

```

Files Plots Packages Help Viewer Presentation

Screenshot 7: A screenshot of your R codes with date and time to show Step 6.7.1-6.7.4. Requirements: your screenshot must clearly include all the R codes for your NN model and the output of 6.7.4.

The screenshot shows an RStudio interface with the following details:

- File Menu:** RStudio, File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, Help.
- Console Tab:** Shows R version 4.3.1 and the output of the R code. The output includes iterative values from 90 to 200, a final value of 688.657841, and a message indicating convergence. It also shows an error in installing NeuralNetTools and the download of the package.
- Code Editor Tab:** Contains R code for building a logistic regression model (LR) and a neural network model (NN). The code includes loading libraries (glm, nnet), fitting models, predicting, and summarizing results. It also handles factor conversion and seed setting.
- Environment Tab:** Shows the current environment variables.
- History Tab:** Shows the history of R commands run.
- Connections Tab:** Shows the connections tab.
- Tutorial Tab:** Shows the tutorial tab.
- Bottom Navigation:** Files, Plots, Packages, Help, Viewer, Presentation.

## Step 7: Comparative Analysis (18 points)

7.3. Please include the following deliverables in your submission:

7.3.1. Please copy and paste the provided table into your submission (your table must be accessible; screenshot or image is not accepted). Ensure that the table includes the predicted results of 20 records using 8 different methods (8 pts in total and each column is worth 1 point).

RM_DT	RM_NB	RM_LR	RM_NN	R_DT	R_NB	R_LR	R_NN
no	no	no	no	no	no	no	no
yes	yes	yes	yes	yes	yes	yes	no
no	no	yes	yes	yes	yes	yes	yes
yes	yes	yes	yes	yes	yes	yes	yes
yes	yes	no	yes	no	no	no	yes
no	no	yes	no	yes	yes	yes	no
no	no	no	no	no	no	no	no
yes	yes	yes	yes	yes	no	no	yes

no	no	no	yes	no	no	no	no
yes	yes	yes	yes	yes	no	no	yes
no	no	yes	yes	yes	yes	yes	yes
yes	yes	yes	yes	no	yes	yes	no
yes	yes	no	no	no	no	no	yes
no	no	no	yes	no	no	no	yes
yes							
yes	no						
no							
yes	yes	yes	yes	no	yes	yes	yes
no	no	yes	yes	yes	yes	yes	yes
no	no	no	yes	no	no	no	yes

7.3.2. Discuss the number of records predicted to be "yes" or "no" by each method in the RM and R datasets. For example, among the 20 records, RM\_DT and R\_DT jointly predict "yes" for 6 records (ID = 2, ....), and jointly predict "no" for 6 records (ID = 1, .....). (8 pts: 2 pts for each pair of methods)

For the 20 records, RM\_DT and R\_DT jointly predicted "yes" for 6 records, IDs = 2, 4, 8, 10, 15, 16. They jointly predicted "no" for 6 records, IDs = 1, 7, 9, 14, 17, 20.

For the 20 records, RM\_NB and R\_NB jointly predicted "yes" for 6 records, IDs = 2, 4, 12, 15, 16, 18. They jointly predicted "no" for 6 records, IDs = 1, 7, 9, 14, 17, 20.

For the 20 records, RM\_LR and R\_LR jointly predicted "yes" for 10 records, IDs = 2, 3, 4, 6, 11, 12, 15, 16, 18, 19. They jointly predicted "no" for 8 records, IDs = 1, 5, 7, 9, 13, 14, 17, 20.

For the 20 records, RM\_NN and R\_NN jointly predicted "yes" for 11 records, IDs = 3, 4, 5, 8, 10, 11, 14, 15, 18, 19, 20. They jointly predicted "no" for 4 records, IDs = 1, 6, 7, 17.

7.3.3. Finally, provide an analysis of the number of records that all eight models predict as "yes" and the number of records that all eight models predict as "no." For example, all eight models jointly predict as "yes" for 2 records (ID = 4.....), and "no" for 4 records (ID = 1,.....). (2 pts).

All eight models predicted "yes" for 2 records, IDs = 4, 15.

All eight models predicted "no" for 3 records, IDs = 1, 7, 17.