

# CIDM 6355 Data Mining Methods Exam 1 Part 2 Instruction

**Requirements: Please read, understand, and comply with the following requirements in this exam.**

- Part 2 is open book, open slides, and open notes, but you are not allowed to collaborate nor discuss with anyone else during the exam period. Any question about the exam should be addressed to the instructor.
- Part 2 is not timed, but you have to submit all the required responses by the deadline to be accepted; a late submission without any acceptable excuse will not be acceptable and a zero point will be assigned.
- This is an individual exam, so sharing your RM processes, R script, screenshots, or answers with other students or parties is considered as cheating, which will be reported to the university authority.
- Please compile all your screenshots in the specified template and then submit it via [Exam 1 Part 2 Submission](#) on WTCClass.

## Instructions & Business Understanding:

- Please download the training dataset eBayAuction.csv and prediction set eBayPrediction.csv from WTCClass.
- The training set contains information about real auctions and their related attributes: product category, currency, seller ratings, auction duration (in number of days), day of week that auction ends, opening price (set by seller). The class label (**Competitive** = “yes” or “no”) indicates if the **auction attracted any bids or not**. In eBay, it is common to have auction listings with no bids.
- You are expected to **build classification models that can be used to predict whether or not the 20 new auctions in the prediction set will attract bids**.

## 1. Data Understanding and Preparation in RM

- 1.1. Import your data using the operator Read CSV. At the third step, format your columns, please change the type of Competitive to binominal because it has only two classes, yes and no.

**Format your columns.**

☐ Replace errors with missing values ⓘ

	SellerRating <i>integer</i>	Duration <i>integer</i>	endDay <i>polynomial</i>	OpenPrice <i>real</i>	Competitive <i>polynomial</i>
	3249	5	Mon	0.010	no
	3249	5	Mon	0.010	no
	3249	5	Mon	0.010	no
	3249	5	Mon	0.010	no
	3249	5	Mon	0.010	no
	3249	5	Mon	0.010	no
	3249	5	Mon	0.010	no

Change Type  
Change Role  
Rename column  
Exclude column

polynomial  
**binominal**  
real  
integer  
date\_time  
date  
time

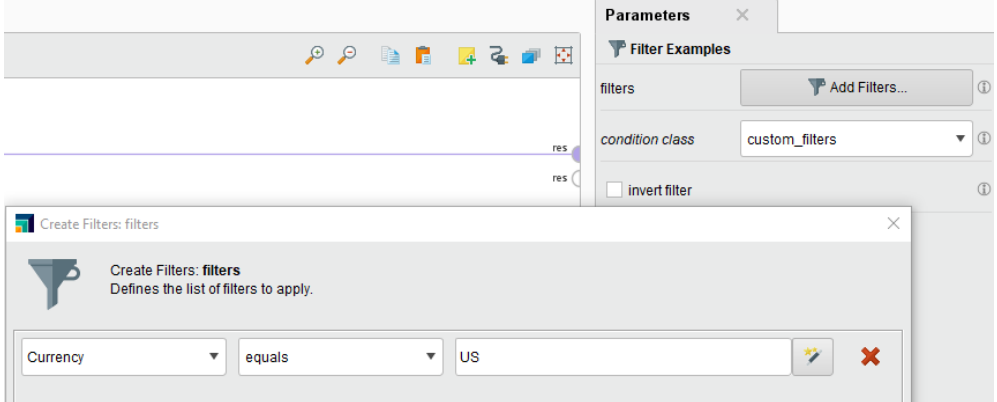
- 1.2. Then, run this process. Observe your data under the Statistics view and then check if the dataset has any quality issue. After careful observation, it has come to our attention that the "Currency" attribute can have values of either US dollars (USD) or British pounds (GBP). In light of this, we have made the decision to focus exclusively on auctions conducted in US dollars (USD). Therefore, we will be excluding

observations related to auctions conducted in British pounds (GBP) from the dataset (Hint: Use the operator Filter Examples” and set "condition class" as "attribute\_value\_filter", "parameter string" as Currency=US).

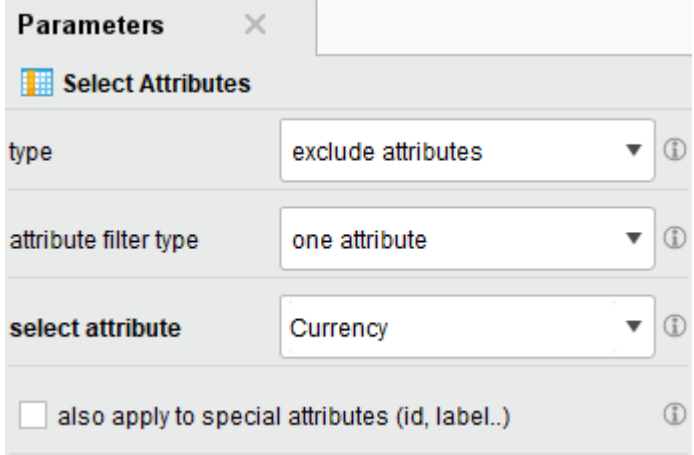
1.3. Now, because Currency has only one value (US) only, it has no prediction power and thus we decide to remove it. Please use the operator Select Attributes to remove Currency.

1.4. In addition, you notice that the attribute OpenPrice has some missing values. Replace these missing values with the **minimum** "OpenPrice" value in the data using the operator Replace Missing Value.

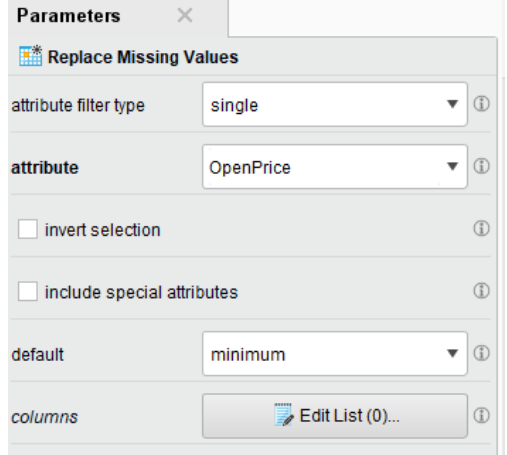
Step 1.2 Filter Examples



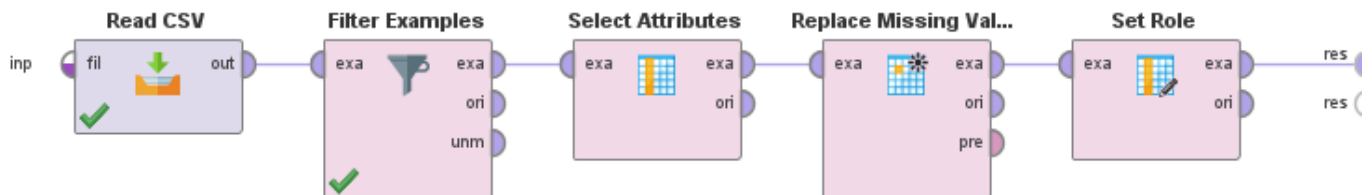
Step 1.3 Select Attributes



Step 1.4 Replace Missing Values



1.5. Set your target attribute. Your process will be like the following:



1.6. Run your process and then think about the following questions based on your Data view and Statistics view (You do not need to submit your answers to these questions).

1.6.1. Among all the auctions left in your dataset, how many of them have attracted bids (Competitive = yes)?

1.6.2. How many percent of auctions have attracted bids?

1.6.3. This percentage above is the estimated probability that a random auction attracts bids. Based on this probability, how many of the new auctions in the prediction dataset are expected to attract bids?

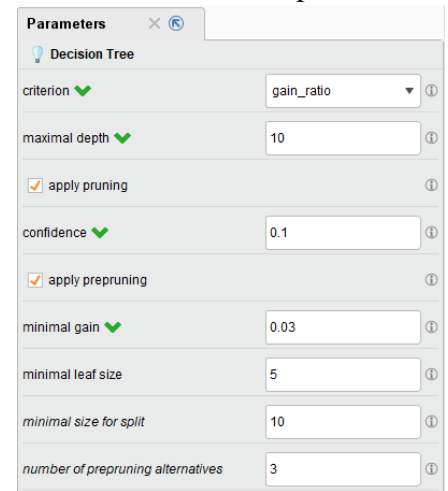
## 2. Decision Tree Model in RM

2.1. Use the Decision Tree operator to generate a decision tree model for this case. We use the parameters shown as below.

2.2. Save your process as Exam1.DT and run it.

2.3. Examine your decision tree model. Take a screenshot of your decision tree graph with date and time (**Screenshot 1**) and briefly describe your model (Your description must include root node, split nodes, and leaf nodes).

2.4. Import the prediction dataset and then use the decision tree model to predict whether or not the 20 new auctions in the prediction dataset will receive a bid.



Parameter	Value
criterion	gain_ratio
maximal depth	10
apply pruning	<input checked="" type="checkbox"/>
confidence	0.1
apply prepruning	<input checked="" type="checkbox"/>
minimal gain	0.03
minimal leaf size	5
minimal size for split	10
number of prepruning alternatives	3

2.5. Copy and paste the prediction column (the column highlighted in light green) to an Excel spreadsheet. **Attention: Make sure that the Row No. in RM is at the ascending order; the demonstration here is not based on this lab, so it is okay if you got different results.** Name that column as RM\_DT.

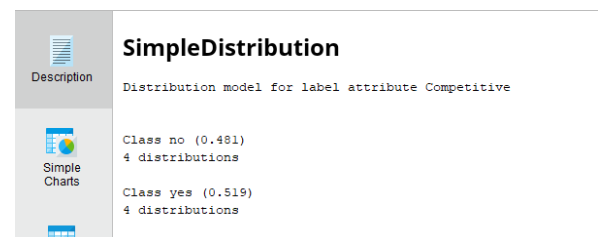
## 3. Naïve Bayes Model in RM

3.1. Use the Naïve Bayes operator to build another classification model using the training set. Make sure that you check Laplace correction for your Naïve Bayes operator. You may simply replace the Decision Tree operator by the Naïve Bayes using the tip provided in Week 5 lab.

3.2. Use the Naïve Bayesian model to predict whether or not the 20 new auctions in the prediction dataset will receive a bid.

3.3. Copy and paste the prediction column (the column highlighted in light green) to an Excel spreadsheet. **Attention: Make sure that the Row No. in RM is at the ascending order.** Name that column as RM\_NB.

**Note: Check the Description view of your Naïve Bayesian model: the number after “class yes” is the decimal format of the result you computed at Step 1.7.2.**



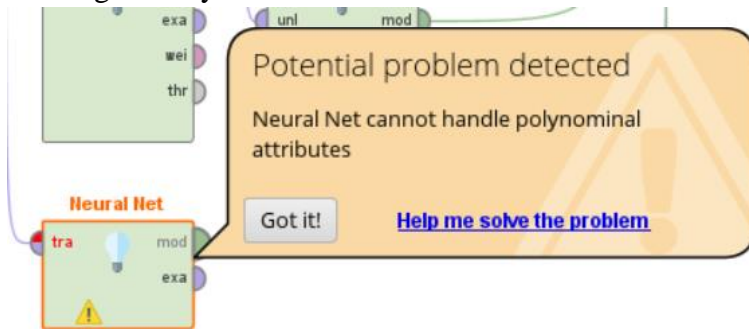
SimpleDistribution	
Distribution model for label attribute Competitive	
Class no (0.481)	4 distributions
Class yes (0.519)	4 distributions

#### 4. Logistic Regression Model in RM

- 4.1. Please build a logistic regression model using the operator Logistic Regression with the specified parameters in the right.
- 4.2. Run your process.
- 4.3. Use the Logistic regression model to predict whether or not the 20 new auctions in the prediction dataset will receive a bid.
- 4.4. Copy and paste the prediction column (the column highlighted in light green) to an Excel spreadsheet. **Attention: Make sure that the Row No. in RM is at the ascending order.** Name that column as RM\_LR.

#### 5. Neural Network Model in RM

- 5.1. Please build a Neural Network Model using the operator Neural Net with the specified parameters in the right. In this case, you do not need to apply “remove highly-correlated attributes” here.
- 5.2. When you receive the following error message, just click “Help me solve the problem” and you will find a new operator “Nominal to Numerical” is added to your process. This operator will convert all the nominal (categorical) attributes to numerical attributes by creating dummy variables.



Parameters	
Logistic Regression	
solver	AUTO
<input type="checkbox"/> reproducible	
<input type="checkbox"/> use regularization	
<input checked="" type="checkbox"/> standardize	
<input type="checkbox"/> non-negative coefficients	
<input checked="" type="checkbox"/> add intercept	
<input checked="" type="checkbox"/> compute p-values	
<input checked="" type="checkbox"/> remove collinear columns	
missing values handling	MeanImputation
max iterations	0
max runtime seconds	0
hidden layers	Edit List (0)...
training cycles	200
learning rate	0.01
momentum	0.9
<input type="checkbox"/> decay	
<input checked="" type="checkbox"/> shuffle	
<input checked="" type="checkbox"/> normalize	
error epsilon	1.0E-4
<input type="checkbox"/> use local random seed	

- 5.3. Use the neural network model to predict whether the 20 new auctions in the prediction dataset will receive a bid. If you may receive an error message indicating that the attributes in the training set does not match with the ones in the prediction set, you need to add the operator, “**Nominal to Numerical**”, the same one that RM added for you at the previous step, between your prediction set and the Apply Model operator.
- 5.4. Take a screenshot of your RapidMiner Process (the flow chart in your design mode) with date and time (**Screenshot 2**) and briefly discuss why the operator **Nominal to Numerical** must be used in your process.
- 5.5. Copy and paste the prediction column (the column highlighted in light green) to an Excel spreadsheet. **Attention: Make sure that the Row No. in RM is at the ascending order.** Name that column as RM\_NN.

## 6. Data Mining in R

**Attention:** when writing your R scripts, please make sure that you briefly describe what each line of R code means. You can find examples in this instruction and previous R lab instructions. If you do not include complete and correct code description or comments, you will lose 50% of points in your screenshots. If your screenshot does not have acceptable dates, you will lose 50% of points in your screenshots.

6.1. Import your datasets and save them as Exam1 and Exam1\_predict.

```
# import training and prediction datasets
Exam1<-read.csv(file.choose(),header=T,stringsAsFactors = T)
Exam1_predict<-read.csv(file.choose(),header=T,stringsAsFactors = T)
```

6.2. Check the structure of your datasets using str().

```
> str(Exam1)
'data.frame': 1972 obs.
 $ Category      : Factor \
 $ Currency      : Factor \
 $ SellerRating  : int 324
 $ Duration      : int 51
 $ endDay        : Factor \
 $ OpenPrice     : num 0.0
 $ Competitive   : Factor \

> str(Exam1_predict)
'data.frame': 20 obs.
 $ Category      : Factor \
 $ Currency      : Factor \
 $ SellerRating  : int 31
 $ Duration      : int 81
 $ endDay        : Factor \
 $ OpenPrice     : num 0.0
 $ Competitive   : logi N
```

6.3. Select records and deal with missing values.

```
# only select those records with US in the attribute currency
Exam1_2<-Exam1[Exam1$Currency!='GBP',]
# check whether GBP records are removed and missing data issues
summary(Exam1_2)
str(Exam1_2)
# remove the attribute, Currency, with only one level
Exam1New <-subset(Exam1_2, select=-Currency)
#check check whether the attribute is removed
str(Exam1New)
# replace the missing data in OpenPrice using the minimal of OpenPrice
Exam1New$OpenPrice[which(is.na(Exam1New$OpenPrice))]<-min(Exam1New$OpenPrice,na.rm=T)
# check if missing values are replaced
summary(Exam1New)
```

Take 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, with date and time (Screenshot 3).

6.4. Decision Tree model

6.4.1. Build a decision tree model using the party library and store your tree in DT (you do not need to set the pruning arguments).

6.4.2. Apply the model to the prediction dataset and store it in R\_DT.

```
#apply the model for prediction and store it in R_DT
R_DT<-predict(DT,Exam1_predict)
```

6.4.3. How many auctions are predicted to attract bids? You may use `summary()` to quickly get the answer.

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

## 6.5. Naïve Bayesian Model

6.5.1. Build a naïve Bayesian model using the `e1071` library and store your model in `NB`.

6.5.2. Apply the model to the prediction dataset and store it in `R_NB`.

```
#apply the model for prediction and store it in R_NB  
R_NB<-predict(NB,Exam1_predict)
```

6.5.3. How many auctions are predicted to attract bids?

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

## 6.6. Logistic Regression Model

6.6.1. Build a logistic regression model using the `glm` function and store your model in `LR`.

6.6.2. Apply the model to the prediction dataset and store it in `R_LRP`.

6.6.3. Convert probabilities to prediction class, convert it to a factor, and store it in `R_LR`.

```
#Apply the model for prediction and store it in R_LRP  
R_LRP<-predict(LR,Exam1_predict,type="response")  
#Convert probabilities to prediction class and then convert it to a factor.  
R_LR<-as.factor(ifelse(R_LRP > 0.5, "yes", "no"))
```

6.6.4. How many auctions are predicted to attract bids?

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

## 6.7. Neural Network Model

6.7.1. Build a neural network model using the `nnet` library with the same seed number and other parameters as Week 6 Lab, and store your model in `NN` (You do not need to remove highly-correlated attributes in this case to be consistent with other models).

6.7.2. Apply the model to the prediction dataset and store it in `R_NNP`.

6.7.3. Convert probabilities to prediction class, convert it to a factor, and store it in `R_NN`.

```
#apply the model for prediction and store it in R_NNP  
R_NNP<-predict(NN,Exam1_predict)  
#Convert probabilities to prediction class and then convert it to a factor.  
R_NN<-as.factor(ifelse(R_NNP > 0.5, "yes", "no"))
```

6.7.4. How many auctions are predicted to attract bids?

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

6.8. Combine all the prediction results.

```
# Convert all the predictions to vector and then combine all of them using rbind
R_Predict <-rbind(as.vector(R_DT),as.vector(R_NB),as.vector(R_LR),as.vector(R_NN))
# Transpose the combination and then convert it to a data frame
R_DF<-as.data.frame(t(R_Predict))
# Rename all the columns
colnames (R_DF) <-c("R_DT", "R_NB", "R_LR", "R_NN")
#write R_DF to a CSV file
write.csv(R_DF,"R_DF.csv")
```

## 7. Comparative Analysis

We are going to combine and compare all the prediction results in R and RM.

7.1. Open R\_DF.csv and then copy and paste all the four columns into the spreadsheet you got in Step 5.4.

Your new spreadsheet will look like the one below (Your results may be different from the one below):

ID	RM_DT	RM_NB	RM_LR	RM_NN	R_DT	R_NB	R_LR	R_NN
1	no	no	no	no	no	no	no	no
2	yes	yes	yes	yes	yes	yes	yes	no
3	no	yes	yes	yes	yes	yes	yes	yes
4	yes	yes	yes	yes	yes	yes	yes	yes
5	yes	no	no	yes	no	no	no	no
6	no	yes	yes	no	yes	yes	yes	yes

7.2. Now, you can compare the prediction results by the same method in RM and R. Usually if a record is predicted to being in the same class by multiple methods and/or tools, it shows the robustness of the prediction. Note: you can use AND function in Excel to compare whether multiple columns have the same result. Please refer to this link <https://exceljet.net/formula/multiple-columns-are-equal> for more details.

7.3. **Please include the following deliverables in your submission:**

7.3.1. Please copy and paste the provided table into your submission. Ensure that the table includes the predicted results of 20 records using 8 different methods.

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, 4, 8, 10, 15, 16), and jointly predict "no" for 6 records (ID =1, 7, 9, 14, 17, 20).

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, 15), and "no" for 3 records (ID =1, 7, 17).

-----The end of Exam 1 Part 2 -----