- Due May 31, 2023 by 11:59pm
- Points 100
- Submitting a file upload
- Available until Jun 3, 2023 at 11:59pm

This assignment was locked Jun 3, 2023 at 11:59pm.

A2 Decision Tree Classification and Evaluation

Instructions

Packages required: Install the packages: caret, C50, and rminer.

Input file (for use in code) - *CD_additional_balanced.csv* (https://utah.instructure.com/courses/882385/files/147060840/download?wrap=1) and data dictionary (not for use in code) - *CD_metadata.xlsx* (https://utah.instructure.com/courses/882385/files/147060897/download?wrap=1). The target variable is *y*.

Submission: Submit two files in canvas – A2_yourLastname_yourFirstname.Rmd, and A2_yourLastname_yourFirstname.html. (Please do not compress these two files into one zip file. Submit them as separate files.)

Note: If there is any discrepancy in the task descriptions in the assignment page and the short rubric descriptions, please follow the task descriptions in the assignment page.

Task I (95%)

(8 points) Create A2_yourLastname_yourFirstname.Rmd that includes code chunks to meet the requirements specified per chunk below. Use the RMD header to include assignment title, author name – you and the file creation date. Also include header specifications to generate a table of contents and section numbers of your code chunks. I suggest that you set your chunk_output_type to console. (See C5.0 tutorials for examples of the RMD header specification.)

1. Code chunk 1 (6 points) – Set up, data import and inspection code for the following:

If needed, load packages using library()

A.
- Remember to use getwd() and setwd() to set the working directory in your rmarkdown file. For example,

```
mydir <- getwd()

setwd(mydir)
```

- Import data using read.csv(). Do not coerce the character variables to factors automatically when loading the data. Examine the overall 'structure' of the input data.

B. Transform all of the character variables that include categorical values to factor variables. After this transformation, show the overall structure and summary of the input data.

2. Code chunk 2 (5 points) - Target variable

For each level of the target variable, show the count and the percentage of instances belonging to that level.

3. Code chunk 3 (20 points)- Data preparation

A. (12 points) Partition the data set for simple hold-out classification model building and evaluation – 70% for training and the other 30% for testing. (not required: Show the summary of train and test sets.)

B. (8 points) Show the distributions (i.e., percentages or proportions of "yes" and "n") of *y* in the train set and in the test set.

Note: Unless otherwise stated use the simple hold-out datasets for model building and evaluation. Only use the entire dataframe if explicitly requested. The Train set is the only set that should be used for training the model. Both the Train and Test sets are to be used for prediction.

4. Code chunk 4 (13 points) – Train and Test Decision Tree 1 to classify *y*

A. (8 points) Train a C5.0 model using the default setting to classify *y* with all other variables as predictors. Show this model and the summary of the model. Do **not** plot the tree at this point.
B. (5 points) Using the predict() and mmetric() functions, generate and compare this model's confusion matrices and classification evaluation metrics in the test and train sets.
    A. Are there significant differences between train and test? Discuss in text.

5. Code chunk 5 (15 points) – Train and Test Decision Tree 2 to classify *y*

A. (10 points) Build a simplified version of Decision Tree 1 by adjusting the confidence factor (CF) of Decision Tree 1. Show this model and the summary of the model. Plot the tree since it is simpler. Try to adjust the CF value between non-zero and 1 to come up with a tree that is simple enough to be plotted.
B. (5 points) Using the predict() and mmetric() functions, generate and compare this model's confusion matrices and classification evaluation metrics in the test and train sets.
    A. Are there significant differences between train and test? Discuss in text. How does performance compare to Chunk 4?

6. Code chunk 6 (13 points)– Train and Test Decision Tree 3 to predict *y*

A. (8 points) Remove the variable – *duration* from the predictors for Decision Tree 3. Using the default setting of C5.0 to build a model to classify *y* with the remaining predictors of Decision Tree 1. Show this model and the summary of the model. Do not plot the tree at this point.
B. (5 points) Using the predict() and mmetric() functions, generate and compare this model's confusion matrices and classification evaluation metrics in the test and train sets.
    A. Are there significant differences between train and test? Discuss in text. How does performance compare to Chunk 4,5?

Note: Be careful here to not accidentally include your target as a predictor. Check your decision tree carefully to ensure the target variable is not accidentally left in the X dataframe when training and is therefore used in the model as a predictor.

7. Code chunk 7 (15 points)– Training and Testing Decision Tree 4 to classify *y*

A. (10 points) Build a simplified version of Decision Tree 3 by adjusting the confidence factor (CF) of Decision Tree 3. Show this model and the summary of the model. Plot the tree.
B. (5 points) Using the predict() and mmetric() functions, generate and compare this model's confusion matrices and classification evaluation metrics in the test and train sets.
    A. Are there significant differences between train and test? Discuss in text. How does performance compare to Chunk 4,5,6?

Task II: Reflections (5%)

What have you learned from building each of these models and the modeling impact of your adjustments to the hyperparameters or dataset? If you were explaining the results of these models to a supervisor what would you say about them? Attempt to do more than just state facts here, interpret the results. Coding is great, interpretation of output is even more important. Discuss each model.

Write at least 150 words.

Suggestions for each chunk:

- Add some simple descriptive text in the text area before the code chunk.
- Add a name or description of each code chunk in {r}. Be sure that you *allow code and output from executing code to be included in the file from rendering A3_yourLastname_yourFirstname.Rmd.*
- Feel free to add comment lines with the requirement item numbers (e.g., # 3.A or # 3.B) to your code cell to help TAs and instructors easily identify your code that addresses a particular requirement.

Render A2_yourLastname_yourFirstname.Rmd to an HTML output file. Submit both files to Canvas. Do not submit a zipped file.

A1

Criteria	Ratings		Pts
Task I Rmd header, chunks and formatting.	**8 pts** **Full Marks**	**0 pts** **No Marks**	8 pts
1.1 Set up, data import and inspection code	**6 pts** **Full Marks**	**0 pts** **No Marks**	6 pts
1.2 Target variable	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
1.3.A Partition the data set for simple hold-out classification model building and evaluation	**12 pts** **Full Marks**	**0 pts** **No Marks**	12 pts
1.3.B Show the distributions (i.e., percentages of “yes” and “n”) of y in the train set and in the test set.	**8 pts** **Full Marks**	**0 pts** **No Marks**	8 pts
1.4.A Train a C5.0 model using the default setting to classify y with all other variables as predictors. Show this model and the summary of the model.	**8 pts** **Full Marks**	**0 pts** **No Marks**	8 pts
1.4.B Using the predict() and mmetric() functions, generate and compare this model’s confusion matrices and classification evaluation metrics in the test and train sets.	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
1.5.A Build a simplified version of Decision Tree 1 by adjusting the confidence factor (CF) of Decision Tree 1. Show this model and the summary of the model. Plot the tree since it is simpler. Try to adjust the CF value from non-zero to 1 to come up with a tree that is simple enough to be plotted.	**10 pts** **Full Marks**	**0 pts** **No Marks**	10 pts
1.5.B Using the predict() and mmetric() functions, generate and compare this model’s confusion matrices and classification evaluation metrics in the test and train sets.	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
1.6.A Remove the variable – duration from the predictors for Decision Tree 3. Using the default setting of C5.0 to build a model to classify y with the remaining predictors of Decision Tree 1. Show this model and the summary of the model. Do not plot the tree at this point.	**8 pts** **Full Marks**	**0 pts** **No Marks**	8 pts
1.6.B Using the predict() and mmetric() functions, generate and compare this model’s confusion matrices and classification evaluation metrics in the test and train sets.	**5 pts** **Full Marks**	**0 pts** **No Marks**	5 pts
1.7.A Build a simplified version of Decision Tree 3 by adjusting the confidence factor (CF) of Decision Tree 3. Show this model and the summary of the model. Plot the tree.	**10 pts** **Full Marks**	**0 pts** **No Marks**	10 pts
1.7.B	**5 pts**	**0 pts**	5 pts

Criteria	Ratings		Pts
Using the predict() and mmetric() functions, generate and compare this model's confusion matrices and classification evaluation metrics in the test and train sets.	**Full Marks**	**No Marks**	
Task II What have you learned from building each of these models and the modeling impact of your adjustments to the hyperparameters or dataset? If you were explaining the results of these models to a supervisor what would you say about them? Attempt to do more than just state facts here, interpret the results. Coding is great, interpretation of output is even more important. Discuss each model. Write at least 150 words.	**5 pts Full Marks**	**0 pts No Marks**	5 pts
			Total Points: 100