MATHS 7107 Data Taming Assignment Five

Due date: 5pm, Wednesday 12th April 2023.

Some rules about your submissions:

- **You must complete this assignment using R Markdown**;
- Your assignment must be submitted as **pdf only** on MyUni;
- You must include **units** when providing solutions;
- Include any working when providing solutions;
- Provide all numerical answers to **3 decimal places**;
- Make sure you include both your code and R output / plots in your answers;
- Make sure any tables or plots included have captions;
- Do not write directly on the question sheet;
- You can submit more than once if you find errors and your latest submission will be marked;
- Make sure you only upload one document for your final submission. If you submit multiple pages (i.e. one per question) you will be deducted 10% per page submitted;
- Penalties for late submission - within 24 hours 40% of final mark. After 24 hours, assignment is not marked and you get zero; and
- Finally, make sure you check your submitted assignment is the correct one, as we cannot accept other submissions after the due date.

The purpose of this assessment is to work through all the relevant steps of constructing a predictive model on a given data set. This will include data cleaning, exploratory analysis, preprocessing, model building and tuning, and finally model evaluation and predicting on new data. This will assess a variety of skills developed in the course.

Label all question answers clearly. Include any relevant code in your answers, as there are marks awarded for code.

The data we are looking at today concerns extramarital activities of readers of *Psychology Today* in 1969. Our aim is to build a predictive model that will determine whether an individual is likely to engage in extramarital affairs based on various aspects of their lives. The variables in this dataset are:

- affair - An indicator of whether the participant had engaged in an affair, categorical.
- sex - the sex of the participant, categorical.
- age - the age in years of the participant, continuous.
- ym - the number of years the participant had been married, continuous.
- child - do they have a child? Categorical.
- religious - how religious are they, ranging from 1 = anti-religious to 5 = very religious.
- education - years of education, ranging from 9 = primary school to 20 = PhD.
- occupation - Job status, ranging from 1-7 according to the Hollinghead classification (reverse numbering so 7 is a better level job).
- rate - How do they rate their marriage, ranging from 1 = unhappy to 5 = very happy.

Further information regarding this data can be found in the original paper:

Fair, R. C. (1978). A theory of extramarital affairs. Journal of Political Economy, 86(1), 45-61.

This includes a more in depth discussion of the data and variables involved.

Data Cleaning

The questions in this section will involve making sure the data is all good to work with. Data cleaning processes include checking for missing data, making sure variables have been read in correctly, and making sure the values of our data are reasonable.

1. Read the data into R, making sure it is a tibble. Display the first 6 rows of the dataset to make sure it has read in correctly (`head` is a good function for this). [2]
2. What is the outcome variable, and what are the predictor variables? [2]
3. Skim the data. Is there any missing data? How many observations and variables do we have? Have any variables been read in incorrectly? [4]
4. Convert the `affair` variable to a yes/no response (the function `ifelse` or `case_when` will be useful). Change all character variables to factors. [3]
5. Skim the data again and answer the following. [5]

a. How many people responded as having had an affair? How many people responded to having children?
b. What is the mean age of respondents? What is the mean response on the religious scale?

Section Total: [16]

Exploratory analysis

This section is concerned with the exploratory analysis of the data. Exploratory analysis is an important part of any model building process. This is where we will examine possible relationships in our data that may help inform our model. We will look at some of the relationships in our dataset using summary statistics and data visualisation.

1. Of the participants who responded "no" to an affair, what proportion of them are female? How about for those who responded "yes" to having an affair? Does there appear to be a difference in the proportion of females who will have an affair opposed to those who will not? (Hint: the function `count` will be useful for this) [3]
2. What proportion of participants who responded "yes" to having an affair had children? How about those participants who responded "no"? Based on this, are you more likely to have children if you have an affair? [3]

Usually we would look at plots to compare the relationships in our dataset using summary statistics and data visualisation. To reduce the length of the assignment, we won't do that this week

Section Total: [6]

Split and preprocess

In this section, we will look at data splitting and preprocessing in TidyModels. Data slitting is an important step in the model building process that will help with avoiding over fitting and the evaluation of models. Preprocessing is the generalised term for performing mathematical operations on your data before modelling it to improve the predictive power of the model.

1. Using `initial_split`, create an `rsplit` of the affairs data. How many observations are in the training set and how many are in the testing set? Do not forget to set a seed for reproducibility using `set.seed(1234)`. [3]
2. Use the functions `training` and `testing` to obtain the test and training sets. Display the first 6 rows of the training set to make sure this has worked properly. [3]
3. What does `step_downsample` from the `themis` package do? Why might we want to down sample our data? [4]
4. In tutorial 3 we saw how to use recipes. Create a recipe, based off of our training data, that will: [4]

- Down sample our data on `affair`. Do this using `themis::step_downsample( affair )` in your recipe,
- Convert all our categorical predictors to dummy variables, and
- Normalise all of our predictor variables to have mean 0 and standard deviation 1.
- Print out the recipe to make sure it has worked using `prep()`.

5. Complete the following:
   a. Use the function `juice` (on the recipe) to get your preprocessed training set. [1]
   b. Use the function `bake` (on the recipe and testing split) to get your preprocessed testing set. This can be both be done in the one function. [1]
6. Skim the preprocessed training data. Explain if the 3 preprocessing steps have done what you expect. [4]

Section Total: [20]

Tune and fit a model

This section is concerned with the tuning and fitting of the model. We will be looking at a k-nearest neighbours model. When considering a k-nearest neighbours model, we need to choose a suitable value for k.

1. Make a model specification for a k-nearest neighbours model. In the model specification, define that we would like to `tune()` the `neigbors` parameter. [2]
2. Create a 5-fold cross validation set from the preprocessed training data. Be sure to set a seed for reproducibility using `set.seed(1234)`. [3]
3. Use `grid_regular` to make a grid of k-values to tune our model on. Using `levels` get 25 unique values for k. You also need to set your `neighbors` to range from 5 to 75. [2]
4. Use `tune_grid` to tune your k-nearest neighbours model using your cross validation sets and grid of k-values. [2]
5. What is the value of k that gives the best accuracy based on our tuned model? (Hint: the function `select_best` will be useful with tuned model as the first parameter and "accuracy" as the second parameter) [2]
6. Finalise the k-nearest model using your results from question 6. Print the model specification to make sure it worked. (Hint: the using `finalize_model()` function is useful here) [2]
7. Fit your finalised model to the preprocessed training data and save it with the variable name `affairs_knn`. [1]

Section Total: [14]

Evaluation

We will now evaluate how well our model is at predicting outcomes on new data. This is vital when you have built a model, so that you can have an accurate understanding of how reliable your predictions will be.

1. Obtain class predictions using your finalised model from the preprocessed **test** set using `predict`. Print the first 6 rows to make sure it worked. [2]
2. Add the true value of `affair` from the testing data to your predictions (Hint: you could use `bind_cols( select( preprocessed_test_data, affair) )`. You will need to change the variable names. Print the first 6 rows to make sure this worked. [2]
3. Get a confusion matrix from your predictions. [2]
4. From your confusion matrix, calculate the sensitivity and specificity of your model. Interpret these values in context. [4]
5. I have a friend: let's call him Bono. Bono is a large alpha **male** from Liverpool. He is **47** years old, has been married for **15** years and has **no** children. He places his religious beliefs at a **2**, his occupation at a **6**, his education at a **20**, and he rates his marriage at an astounding **5**.

a. Make a tibble containing Bono's information. Be careful when you do this. You need to name your variables *exactly* how they appear in the `affairs` dataset (i.e. `sex`, `age`, etc). Remember, case matters. [1]

b. Use `bake` to preprocess Bono's information with your recipe. [2]

c. Using the `predict()` function, obtain a predicted probability (i.e. with `type = "prob"`) that Bono will have an affair. [2]

d. Given what we have done, would you be comfortable going to Bono's partner with your prediction of whether Bono will have an affair or not? [2]

Section Total: [17]

Assessment Total: [73]