MONASH University

Faculty of Information Technology

Semester 1, 2024

FIT5145: Foundations of Data Science

Assignment 4: Description

Sunday, Week 13 (June 2, 2024) 11:55 PM

Assessment Details:

- Assessment Type: Individual Assignment
- Total marks: 40%
- Due Date: Sunday, Week 13 (June 2, 2024) 11:55 PM. Please note that we do not accept submissions after June 9, 2024 (i.e., 7 days after the due date).

Hand in Requirements:

In this assignment, two files (PDF report and RMD file) should be submitted.

1. A **report in PDF** containing your ***(a) code, (b) answer,*** *and* ***(c) explanation*** used to answer each question. Please make sure that your answers to all the questions are numbered correspondingly.

   (a) ***code:*** Make sure to include all the shell commands for Task A and the R codes for Tasks B-D in the PDF report. For the shell commands, please **copy** your codes and **paste** into Word or other word processing software (**Please do NOT take the screenshots of your code**).

   For the R codes, please directly convert the RMD file including your codes into the PDF file (Note: Please Knit RMD to HTML and print the HTML as pdf). If you want to use Microsoft Word or other word processing software to format your submission, please **copy** your codes from the RMD file and **paste** into Word (Please do **NOT** take the screenshots of your code).

   You need to merge the shell PDF and R PDF files into a single pdf file.

   (b) ***answer:*** Please make sure to include **screenshots/images of the code outputs and written answers (not screenshot)** for each question of Tasks A-D in PDF.

   (c) ***explanation:*** Please explain how you answered each question (i.e., explaining your codes or summarising your work for each question).

   Marks will be assigned to reports based on their correctness and clarity. For instance, higher marks will be given to reports containing graphs with appropriately labelled

axes.

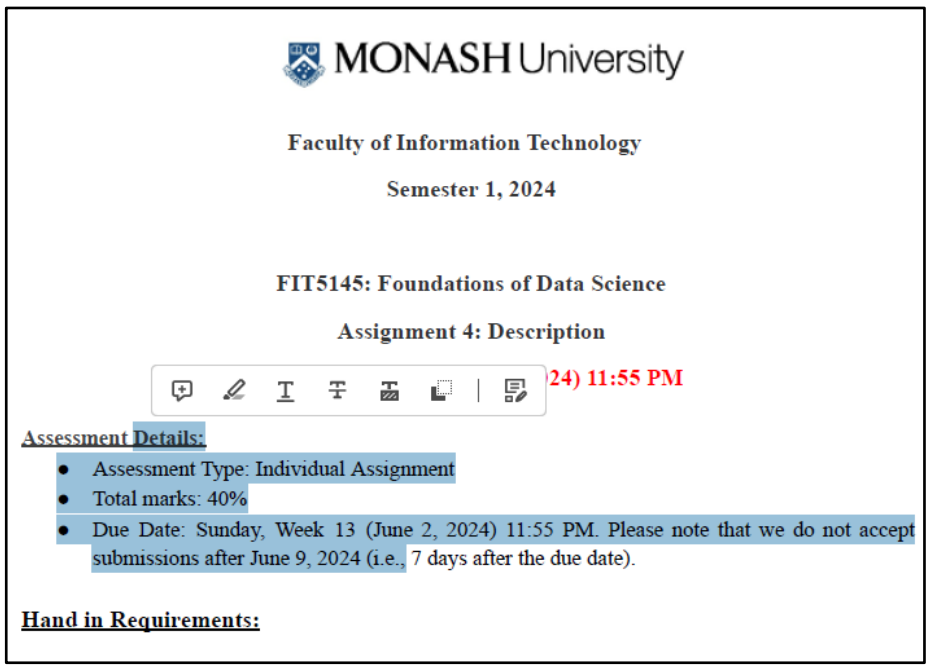
2. The **RMarkdown** file: Please submit the RMarkdown file that contains your R codes for Tasks B-D of this assignment. Your file should contain all the codes, proper comments, and any instructions of libraries that need to be installed.

Notes:

1. Whenever a question asks for a certain value, your code should produce the value. For example, when a question asks for the number of rows contained in a table, your code should print out the answer. Extraction of the answer manually will not earn any marks.
2. Assignment should be submitted in two files (PDF report and RMD file):

   (a) An RMD file that generates errors when running will not be considered

3. **Please make sure that you can select and highlight texts in your PDF,** as shown below then the turnitin score can be generated properly for your PDF file (we just need the Turnitin score for the PDF file, not the RMD file).



MONASH University

Faculty of Information Technology

Semester 1, 2024

FIT5145: Foundations of Data Science

Assignment 4: Description

24) 11:55 PM

Assessment Details:

- Assessment Type: Individual Assignment
- Total marks: 40%
- Due Date: Sunday, Week 13 (June 2, 2024) 11:55 PM. Please note that we do not accept submissions after June 9, 2024 (i.e., 7 days after the due date).

Hand in Requirements:

Task A: Shell commands

Are you interested in buying a property in Melbourne? Have you realised that the rent and home prices have seen significant increases over the past year? In this task, you are required to explore and wrangle the data in the file "***property_transaction_victoria.csv***", which contains most of the property transactions that took place in Victoria in 2021. The data was collected from https://www.domain.com.au/. The file contains different variables to describe each collected transaction record, as described below.

Column Name	**Description**
ID	The unique ID of a transaction record, which usually consists of 8 digits
postcode	The postcode of a property
suburb	The suburb where a property locates in
sold_time	The date on which the transaction was made
sold_type	The type of transaction, i.e., whether a property was sold in an auction or via private sale
sold_price	The price for which the property was sold
address	Address of the property
beds	The number of bedrooms that a property has
baths	The number of bathrooms that a property has
parking	The number of parking spaces that a property has
area	The area size of a property (measured in square metres)
property_type	Whether a property is a *House*, *Townhouse*, etc.
features	A textual description specifying the features that a property has, e.g., whether it has a dishwasher or a shed.
description	A textual description that real estate agents used to describe the property and attract potential buyers before the transaction was made.

Please note that you are only allowed to use shell commands as you would run in Linux shell, Mac terminal, or Cygwin, to tackle this task. Using other utilities or tools such as PowerShell is NOT allowed.

1. What is the *sold_time* range of the records? Please note that the file is not guaranteed to be sorted and Nulls (NA and empty values) should not be considered. Hint: you can change the delimiter of the dataset to sort the *sold_time*.
2. We want to preprocess the ID and sold_time columns.
   a. **Count** lines with an *id* that is not a number of 6 digits long, i.e., *id* values that contain anything other than numbers OR are of a length more/less than 6.
   b. **Remove** the lines mentioned in Q2-a and **remove** time values in the *sol_time* column. *For example,* the *sold_time* column will contain "25/05/2021", instead of having "25/05/2021 12:03".
   c. **Display** the first 5 lines of the dataset that was filtered in Question 2-b. Store the filtered dataset in a file named "***filtered_property.csv***" and **use this file for the following questions in Task A.**
3. When was the first and last mention of the term "*Mount Dandenong*" in the column *address*? Please note that the first and last mention of a term refers to the earliest and latest transaction whose *address* contains the term in the dataset and the term to be searched is case sensitive.
4. Let's investigate the *suburb* column.
   a. How many unique values are there in the *suburb* column?
   b. Can you write commands to list the top 5 most frequent *suburb* values in the dataset (i.e., the top 5 suburbs with the largest number of transaction records in 2021)?
5. Let's investigate the *description* column.
   a. How many transaction records contain both "*Alfresco*" and "*Renovation*" in their *description* value in the dataset? (Note: Please ignore cases).
   b. How many transaction records contain the property size information (e.g.: 1249m2, 758 sq metres) in their *description* value in the dataset? (Note: Please ignore cases).
6. In the following, please generate the dataset filtered according to the following conditions:
   a. Keep these columns: *ID, sold_time, sold_type, sold_price, address, beds, area, property_type, and description*
   b. Keep the transaction records satisfying the following: (i) whose *sold_time* belongs to odd months (e.g.: January, March, …, November); (ii) whose *property_type* is Townhouse; and (iii) whose area is larger than 300 square metres.

   Then, print out the first and last sold time of the filtered dataset (Please include a header).

Task B: Data Collection and Exploratory Data Analysis Using R

There are many ways to collect data from different sources. One of them is web scraping. In this task, you are required to scrape data from websites, wrangle data scrapped if required, and visualise them.

Task B1:

Please extract the following table, "Historical rankings" from the web, ICC Men's T20I Team Rankings in Wikipedia (Note: please extract the entire table).

(https://en.wikipedia.org/wiki/ICC_Men%27s_T20I_Team_Rankings).

Historical rankings [edit]

This table lists the teams that have historically held the highest rating since the T20I rankings was introduced.[citation
decided to grant full T20I status to all its members. As a result, ratings of leading teams since 2018 have been consi
directly compared to those before that date.

Country	Start	End	Duration	Cumulative	Highest Rating
England	24 October 2011[5]	7 August 2012 [6]	289 days	289 days	140
South Africa	8 August 2012	11 September 2012	35 days	35 days	137
England	12 September 2012	21 September 2012	10 days	299 days	130

Then, please print out the following summarised table. Note: You might need to pre-process the data that you extracted from the web.

Country	Earliest_start	Latest_end	Average_duration
Pakistan	2017-11-01	2020-04-30	294.67
Sri Lanka	2012-09-29	2016-02-11	212.80
New Zealand	2016-05-04	2018-01-27	191.33
England	2011-10-24	2022-02-20	187.00
India	2014-03-28	2024-06-02	164.50
Australia	2020-05-01	2020-11-30	106.00
South Africa	2012-08-08	2012-09-28	21.00
West Indies	2016-01-10	2016-01-30	21.00

The summarised table shows the earliest start, latest end, and average duration for each country.

Task B2: Please choose a different website that you are interested in and follow the instructions below:

1. Scrape data contained in table format in a website
2. Wrangle data if required.
3. Create a plot for the scraped data.
4. Discuss the information or insights that can be drawn from the chart.

Note: Please refer to Week 3 lab activity material for web scraping.

Task C: Exploratory Data Analysis using R

Are you a fan of sports? Have you watched the Olympic games held in Tokyo - Japan in 2020? In this task, you are required to explore and wrangle the data in the file "Olympics_tweets.csv", which contains tweets related to the Tokyo Olympic 2021 that were collected from Twitter. The file contains different variables to describe the collected tweets, e.g., the id of a tweet, the text content of a tweet, the screen name of a user who posted a tweet (i.e., user_screen_name), and so on. *Please use R studio to perform the following analyses.*

1. When analysing data collected from Twitter, it is often of equal importance to analyse the tweets (to know people's viewpoints and opinions) and the creators of those tweets (to understand who has those viewpoints and opinions). Now, let's first focus on the creators of the tweets.

    1.1. Write code to produce a bar chart to visualise the number of Twitter accounts created across different years (Note: Please create the "***year***" column).

    1.2. For users whose accounts were generated after 2010, what is the average number of "***user_followers***" of these users for each year? Write code to produce a bar chart to visualise these average "***user_followers***" numbers across different years.

    1.3. Based on the two bar charts generated in Question 1.1 and Question 1.2, what observations can you make? Any potential explanations for your observations?

    1.4. In addition to *when* those Twitter accounts were created, it might be worth further exploring *where* those Twitter users located. Please write code to count the occurrences of different location values (i.e., the column "*user_location*") and display the top 10 most frequent location values. Are there any odd values observed in the top 10 most frequent locations? How many tweets are associated with these top 10 most frequent location values?

2. Now let's move on to analyse the tweets.

    2.1. Please write code to produce a bar chart to visualise the number of tweets posted in different dates (e.g., "25/7/2021") (Note: Please create the "***date***" column). Which date has the lowest number of tweets?

    2.2. Please write code to calculate the length of the *text* contained in each tweet (measured in characters) and produce a bar chart to show the number of tweets of the following length:

    - [1, 40]
    - [41, 80]
    - [81, 120]
    - [121, 160]
    - [161, 200]
    - [201, 240]
    - >= 241

    2.3. In Twitter, people often interact with one another by mentioning another account's username, which is preceded by the "@" symbol (e.g., "Hello @TwitterSupport!"). How many tweets contain another account's username in the dataset? Among the tweets containing another account's username, how many of them contain at least three different accounts' usernames?

    2.4. What are the top 20 most frequent terms in all tweets in the "***text***" column? Are there any stopwords among them? If yes, could you please identify the top 20 most frequent terms which are not stopwords?

Task D: Predictive Data Analysis using R

Do you think the FLoRA chatbot powered by GPT-4 is useful for solving Assignment 1? In this task, you will be asked to analyse the conversational data generated by students in this unit when interacting with the chatbot and perform predictive data analysis to characterise the *usefulness* of a dialogue. Please download the conversational data files from Moodle. All data has been anonymised.

You are required to build machine learning models to predict the usefulness of a dialogue represented in numerical scores. In total, we collected and pre-processed a total of 199 dialogues, and you can access 70% of these dialogues (shared in the data files "*dialogue_utterance_**train**.csv*" and "*dialogue_usefulness_**train**.csv*"), which are randomly selected as the training set that you can use to build machine learning models. Among the remaining 30% dialogues, 15% of them are randomly selected as the validation set and can be

accessed via the files "*dialogue_utterance_**validation**.csv*" and "*dialogue_usefulness_**validation**.csv*". The other 15% are used as the test set and can be accessed via the files "*dialogue_utterance_**test**.csv*" and "*dialogue_usefulness_**test**.csv*". Please refer to Table 1 and Table 2 to know the meaning of each feature/column.

Table 1: Description of columns in the data files "*dialogue_utterance_train/validation/test.csv*"

Column Name	**Description**
Dialogue_ID	The unique ID of a dialogue
Timestamp	When an utterance contained in the dialogue was made
Interlocutor	Whether the utterance was made by the student or the chatbot
Utterance_text	The text of the utterance

Table 2: Description of columns in the data file "*dialogue_usefulness_train/validation/test.csv*"

Column Name	**Description**
Dialogue_ID	The unique ID of a dialogue
Usefulness_score	This score is given by a student to indicate their perceived usefulness of the FLoRA chatbot when answering the post-task questionnaire Question 3 (i.e., "*To what extent do you think the GPT-powered chatbot on FLoRA is useful for you to accomplish the assignment?*"). The value range of this feature is [1,5], with 1 representing "*very unuseful*", 2 representing "*unuseful*", 3 representing "*neutral*", 4 representing "*useful*", and 5 representing "*very useful*".

If the dialogue you generated is included as part of the training set, you need to first exclude it before answering the following questions. The *Dialogue_ID* of your dialogue will be shared with you via email.

1. What features can you engineer to empower the training of a machine learning model? You may propose as many as you believe are useful. Please note that the number of the features should not exceed the number of the dialogues contained in the training set. Otherwise, the constructed machine learning models are prone to have overfitting issues. Select two features that you propose and try to use boxplots to visualise the feature value between the following two groups of dialogues in the training set: (i) <u>those</u>

with *Usefulness_score* of 1 or 2; and (ii) those with *Usefulness_score* of 4 or 5. Is there any difference between the two groups of dialogues? How can you tell whether the difference is statistically significant?

2. Build a machine learning model (e.g., polynomial regressions, regression tree) by taking all the features that you have proposed and evaluate the performance of the model on the validation set using the relevant evaluation metrics you learned in class. The best-performing model here is denoted as *Model 1*.

3. Now we want to improve the performance of *Model 1* (i.e., to get a more accurate model). For example, you may try some of the following methods to improve a model:

    - Select a subset of the features (especially the important ones in your opinions) as input to empower a machine learning model.
    - Deal with errors (e.g.: filtering out data outliers).
    - Rescale data (i.e., bringing different variables with different scales to a common scale).
    - Transform data (i.e., transforming the distribution of variables).
    - Try other machine learning algorithms that you know.

    Please build the predictive models by trying some of the above methods or some other methods you can think of and evaluate the performance of the models and report whether *Model 1* can be improved.

    You need to explain how you have improved your model by including code, output, and explanations (explaining the code or the process) and **justify why you have chosen some of the above methods or some other methods to improve a model** (e.g., why this subset of the variables are chosen to build a model). Marks will be given, based on the depth of investigation required to improve a model, as well as the sufficient justification provided for the proposed approaches.

4. What is the *Dialogue_ID* of the dialogue you generated? Please copy and paste the whole dialogue text that you generated with the chatbot here. With the best-performing model constructed from Question 2&3, what is the prediction value for the dialogue you generated? Is the prediction value close to the groundtruth value? If yes, what features do you think play important roles here to enable the model to successfully make the prediction? If not, what might be the reasons? For students whose dialogues are included in the test set, you may randomly select a dialogue from the validation set to analyse and answer this question.

5. Please notice that the groundtruth *Usefulness_score* values in the file "*dialogue_usefulness_test.csv*" are withheld for now, but they will be shared after the due date of this assignment. Here, your task is to use the best-performing model constructed from Question 2&3 to predict the usefulness of the dialogues contained in the test set. You need to populate your prediction results (i.e., the predicted *Usefulness_score* values) into the file "*dialogue_usefulness_test.csv*" and upload it to

Moodle to measure the overall performance of your model. Please name the submission file using the following format:

LastName_StudentNumber_dialogue_usefulness_test.csv.

The mark you receive for this question will be dependent on the performance level of your model (measured by RMSE).