# FIT5196-S1-2024 Assessment 2

***This is a group assessment and is worth 35% of your total mark for FIT5196.***

**Due date: Friday 24 May 2024, 11:55pm**

# Task 1. Data Cleansing (55%)

For this assessment, you are required to write Python code to analyse your dataset, find and fix the problems in the data. The input and output of this task are shown below:

**Table 1. The input and output of task 1**

| Input files | Submission | |
|---|---|---|
| | **Output files** | **Other Deliverables** |
| <group_id>_dirty_data.csv<br><group_id>_outlier_data.csv<br><group_id>_missing_data.csv<br>branches.csv<br>edges.csv<br>nodes.csv<br>suburb_info.xlsx | <group_id>_dirty_data_solution.csv<br><group_id>_outlier_data_solution.csv<br><group_id>_missing_data_solution.csv | <group_id>_ass2.ipynb<br><group_id>_ass2.py<br><group_id>_form.pdf |

**Note1: All files must be zipped into a file named** *<group_id>_ass2.zip (please use zip not rar, 7z, tar, etc.)*

**Note2: Replace <group_id> with your group id (do not include <>)**

**Note3: You can find your three input files here. Using the wrong files will result in zero marks.**

**Note4: Please strictly follow the instructions in the appendix to generate the .ipynb and .py files.**

Exploring and understanding the data is one of the most important parts of the data wrangling process. You are required to perform graphical and/or non-graphical EDA methods to understand the data first and then find the data problems. In this assessment, you have been provided with three data inputs along with 3 additional files: *branches.csv*, *edges.csv* and *nodes.csv* here. Due to an unexpected scenario, a portion of the data is missing or contains anomalous values. Thus, before moving to the next step in data analysis, you are required to perform the following tasks:

1. Detect and fix errors in *<group_id>_dirty_data.csv*
2. Impute the missing values in *<group_id>_missing_data.csv*
    - (impute the missing *delivery_fee* attribute only)
3. Detect and **remove** outlier rows in *<group_id>_outlier_data.csv*
    - (outliers are to be found with respect to *delivery_fee* attribute only)

As a starting point, here is what we know about the dataset in hand:

The dataset contains Food Delivery data from a restaurant in Melbourne, Australia[1]. The restaurant has three branches around the CBD area. All three branches share the same menu but they have different management so they operate differently.

Each instance of the data represents a single delivery order. The description of each data column is shown in Table 2.

**Table 2. Description of the columns**

| COLUMN | DESCRIPTION |
|---|---|
| **order_id** | A unique id for each order |
| **date** | The date the order was made, given in YYYY-MM-DD format |
| **time** | The time the order was made, given in hh:mm:ss format |
| **order_type** | A categorical attribute representing the different types of orders namely: Breakfast, Lunch or Dinner |
| **branch_code** | A categorical attribute representing the branch code in which the order was made. Branch information is given in the *branches.csv* file. |
| **order_items** | A list of tuples representing the order items: first element of the tuple is the item ordered, and the second element is the quantity ordered for that item. |
| **order_price** | A float value representing the order total price |
| **customer_lat** | Latitude of the customer coming from the *nodes.csv* file |
| **customer_lon** | Longitude of the customer coming from the *nodes.csv* file |
| **customerHasloyalty?** | A logical variable denoting whether the customer has a loyalty card with the restaurant (1 if the customer has loyalty and 0 otherwise) |
| **distance_to_customer_KM** | A float representing the shortest distance, in kilometres, between the branch and the customer nodes with respect to the *nodes.csv* and the *edges.csv* files. Dijkstra algorithm can be used to find the shortest path between two nodes in a graph. Reading materials can be found here. |

[1] The dataset is fictional

**Notes:**

1. The output *csv* files **must** have the exact same columns as the respective input files.
2. In the file *<group_id>_dirty_data.csv*, any row can carry no more than one anomaly. (i.e. there can only be one issue in a single row.)
3. All anomalies in dirty data have one and only one possible fix.
4. There are no data anomalies in the file *<group_id>_outlier_data.csv* except for outliers. Similarly, there are only coverage data anomalies (i.e. no other data anomalies) in *<group_id>_missing_data.csv*.
5. There are three types of meals:
    - Breakfast - served during morning (8am - 12pm),
    - Lunch - served during afternoon (12:00:01pm - 4pm)
    - Dinner - served during evening (4:00:01pm - 8pm)

   Each meal has a distinct set of items in the menu (ex: breakfast items can't be served during lunch or dinner and so on).
6. In order to get the item unit price, a useful python package to solve multivariable equations is numpy.linalg
7. Delivery fee is calculated using a different method for each branch.
   The fee depends linearly (but in different ways for each branch) on:
    a. weekend or weekday (1 or 0) - as a continuous variable
    b. time of the day (morning 0, afternoon 1, evening 2) - as a continuous variable
    c. distance between branch and customer

   It is recommended to use **sklearn.linear_model.LinearRegression** for solving the linear model as demonstrated in the tutorials.

8. Using **proper data** for model training is crucial to have a **good** linear model (i.e. $R^2$ score over 0.95 and very close to 1) to validate the delivery fee. The better your model is, the more accurate your result will be.
9. **If a customer has loyalty, they get a 50% discount on delivery fee**
10. The restaurant uses the Djikstra algorithm to calculate the shortest distance between customer and restaurant. (explore **networkx** python package for this or alternatively find a way to implement the algorithm yourself)
11. The branch and customer nodes are provided in *branches.csv*, *edges.csv* and *nodes.csv* here.
12. The below columns are error-free (i.e. don't look for any errors in dirty data for them):
    - **order_id**
    - **time**
    - **the numeric quantity in order_items**
    - **delivery_fee**
13. For missing data imputation, you are recommended to try all possible methods to impute missing values and keep the most appropriate one that could provide the best performance.
14. As EDA is part of this assessment, no further information will be given publicly regarding the data. However, you can brainstorm with the teaching team during tutorials or on the Ed forum.
15. **No libraries/packages restriction.**

# Task 2: Data Reshaping (15%)

You need to complete task 2 with the *suburb_info.xlsx* file **ONLY**. With the given property and suburb related data, you need to study the effect of different normalization/transformation(i.e.

standardisation, min-max normalization, log, power, box-cox transformation) methods on these columns: **number_of_houses, number_of_units, population, aus_born_perc, median_income, median_house_price**. You need to observe and explain their effect assuming we want to develop a **linear model** to predict the **"median_house_price**" using the 5 attributes mentioned above. When reshaping the data, we have two main criteria. First, we want our features to be on the same scale; and second, we want our features to have as much linear relationship as possible with the target variable (i.e., median_house_price). You need to first explore the data to see if any scaling or transformation is necessary (if yes why? and if not, also why?) and then perform appropriate actions and document your results and observations. Please note that you don't need to actually build a linear model and just need to prepare the data for a linear regression model.
This is **an exploratory task**, thus you **only need** to put your journey of exploration in proper documentation in your .ipynb file, **no other output file to be submitted for task 2.** We will mark based on the .ipynb content for task 2.

**Table3. Description of the suburb_info.xlsx file.**

| | |
|---|---|
| **suburb** | The suburb name, which serves as the index of the data |
| **number_of_houses** | The number of houses in the property suburb |
| **number_of_units** | The number of units in the property suburb |
| **municipality** | The municipality of the property suburb |
| **aus_born_perc** | The percentage of the Australian-born population in the property suburb |
| **median_income** | The median income of the population in the property suburb |
| **median_house_price** | The median 'house' price in the property suburb |
| **population** | The population in the property suburb |

# Methodology (15%)

The report **<group_id>_ass2.ipynb** should demonstrate the methodology (including all steps) to achieve the correct results.

You need to demonstrate your solution using correct steps.

- Your solution should be presented in a proper way including all required steps.
- You need to select and use the appropriate Python functions for input, process and output.
- Your solution should be an efficient one without redundant operations and unnecessary reading and writing the data.

# Documentation (10%)

The cleaning task must be explained in a well-formatted report (with appropriate sections and subsections). Please remember that the report must explain the complete EDA to examine the data, your methodology to find the data anomalies and the suggested approach to fix those anomalies.

The report should be organised in a proper structure to present your solutions with clear and meaningful titles for sections and subsections or sub-subsection if needed.

- Each step in your solution should be clearly described and justified. For example, you can write to explain your idea of the solution, any specific settings, and the reason for using a particular function, etc.
- Explanation of your results including all intermediate steps is required. This can help the marking team to understand your solution and give partial marks if the final results are not fully correct.
- All your codes need proper (but not excessive) commenting.

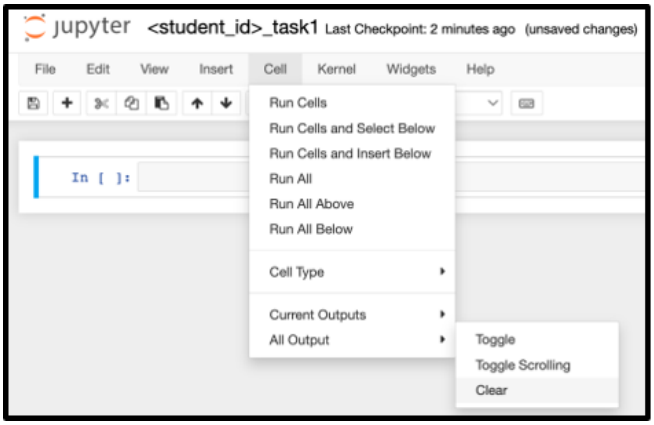
# Submission Requirements

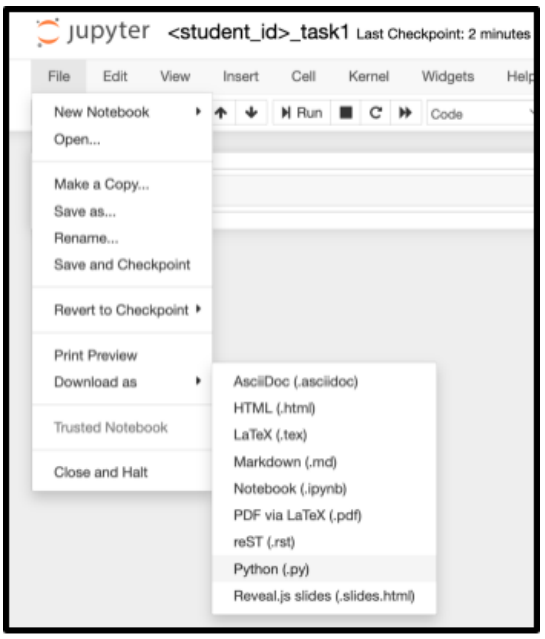
You need to submit the following 6 files:

- A file named **<group_id>_dirty_data_solution.csv**: This file should contain the data records with all identified errors fixed.
- A file named **<group_id>_missing_data_solution.csv**: This file must contain the data records with all missing values imputed.
- A file named **<group_id>_outlier_data_solution.csv**: This file should include the remaining data records with all outliers removed.
- A Python notebook named **<group_id>_ass2.ipynb**: In this notebook, provide a well-documented report that comprehensively demonstrates your solutions for the 'dirty,' 'missing,' and 'outlier' data files of Assessment 2. You need to clearly present your methodology through a step-by-step process, accompanied by relevant comments and explanations. Your approach can vary, but clarity is paramount. Please keep this notebook easy-to-read, as you will lose marks if we cannot understand it. (make sure the cell outputs are NOT cleared)
- A file named **<group_id>_ass2.py**: This file will be used for plagiarism check. (make sure the cell outputs are cleared before exporting)
- A file named **<group_id>_form.pdf:** This is the SIGNED contribution form for the group assessment.

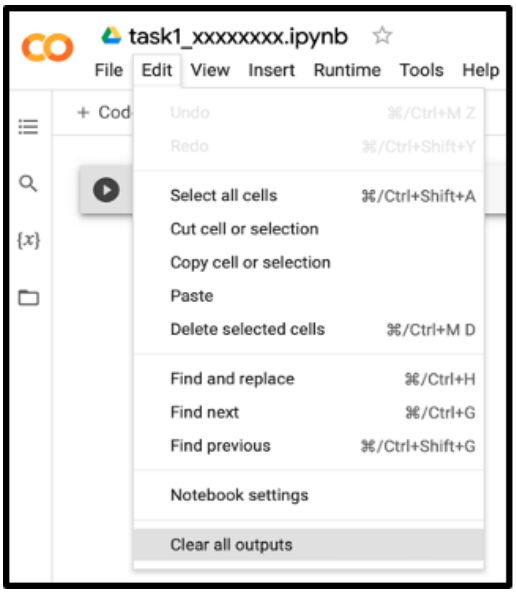
To generate a .py file, you need to clear all the cell outputs, and then download it.

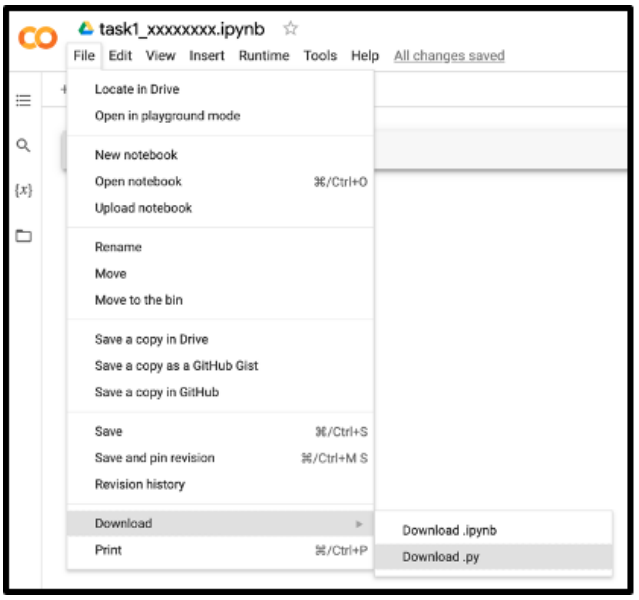
In Jupyter notebook:




In Google colab:



# Task 3: Presentation (5%)

## Group Interview Presentation (Hurdle)

There will be a mandatory presentation for your A2. The aim for the presentation is to check your understanding of your A2 project and make sure all submissions are compliant with the academic integrity requirements of Monash.

**Details:**

- Time/Date: Week 12, during your allocated Applied sessions
- Duration: Approximate 5-10 minutes per group
- Location: Normal location of allocated applied sessions in your Allocate+ records
- Arrangement: We will provide a time schedule for every group during their allocated session, please arrive at your allocated time slot. If you arrive earlier, please wait patiently outside the room.
- Content: Please briefly describe your logic of A2 (at least for task 1) and answer questions if any
- Criterion: Please refer to A2 marking rubrics

**Attendance Requirement:** Mandatory attendance (HURDLE)

**Consequences of Non-Attendance:**
Failure to attend the presentation or inability to satisfactorily demonstrate your work will result in not meeting the hurdle requirements for Assignment A2. Consequently, you will **receive ZERO for Assessment 2**.

The following excuses will not be accepted:

- Forget to come to the tutorial
- Forget to prepare for the presentation, i.e. forget your own solution
- Too limited time to prepare your presentation
- Be too nervous to talk in English
- Direct use online resources without proper reference

# Submission Checklist

- Please zip all the submission files for assessment 2 into a single file with the name **<group_id>_ass2.zip.** (any other format e.g. rar or 7z will be penalised)
- There are **6 files** in your compressed zip file
- **<group_id>** should be replaced with **your group id (without <>)**.
- Please strictly follow the file naming standard. Any misnamed file will carry a penalty.
- Please make sure that your **.ipynb file** contains printed output, while your **.py file** does not include any output.
- Please ensure that all your files are parsable and readable. You can achieve this by re-reading all your generated files back into python. (e.g. using `read_csv` for CSV files). These checks are only sanity checks and hence should not be added to your final submission.
- Make sure to attend your allocated tutorial sessions on Week 12 for mandatory presentation

Note: All submissions will be put through a plagiarism detection software which automatically checks for their similarity with respect to other submissions. Any plagiarism found will trigger the Faculty's relevant procedures and may result in severe penalties, up to and including exclusion from the university.