FIT5196-S2-2025 Assessment 2

This is a group assessment and is worth 40% of your total mark for FIT5196.

Due date: Monday 21 Oct 2025, 11:55pm

Task 1. Data Cleansing (21/40)

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<group_id>_dirty_data.csv Group<group_id>_outlier_data.csv Group<group_id>_missing_data.csv warehouse.csv	Group<group_id>_dirty_data_solution.csv Group<group_id>_outlier_data_solution.csv Group<group_id>_missing_data_solution.csv	Group<group_id>_ass2_task1.ipynb Group<group_id>_ass2_task1.py

Note1: All files must be zipped into a file named Group<*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 all your input files from the folder with your group number 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 the additional file: warehouse.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*
3. **Detect** and **remove** outlier rows in *<group_id>_outlier_data.csv*
   - (w.r.t. the *delivery_charges* attribute only)

Project Background

As a starting point, here is what we know about the dataset in hand:
The dataset contains transactional retail data from an online electronics store (DigiCO) located in Melbourne, Australia[1]. The store operation is exclusively online, and it has three warehouses around Melbourne from which goods are delivered to customers.

Each instance of the data represents a single order from DigiCO store. 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
customer_id	A unique id for each customer
date	The date the order was made, given in YYYY-MM-DD format
nearest_warehouse	A string denoting the name of the nearest warehouse to the customer
shopping_cart	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 denoting the order price in AUD. The order price is the price of items before any discounts and/or delivery charges are applied.
customer_lat	Latitude of the customer's location
customer_long	Longitude of the customer's location

[1] The dataset is fictional

coupon_discount	An integer denoting the percentage discount to be applied to the order_price.
distance_to_nearest_warehouse	A float representing the arc distance, in kilometres, between the customer and the nearest warehouse to him/her. (radius of earth: 6378 KM)
delivery_charges	A float representing the delivery charges of the order
order_total	A float denoting the total of the order in AUD after all discounts and/or delivery charges are applied.
season	A string denoting the season in which the order was placed. Refer to this link for details about how seasons are defined.
is_expedited_delivery	A boolean denoting whether the customer has requested an expedited delivery
latest_customer_review	A string representing the latest customer review on his/her most recent order
is_happy_customer	A boolean denoting whether the customer is a happy customer or had an issue with his/her last order.

Notes:

1. The output *csv* files **must** have the exact same columns as the respective input files. Any misspelling or mismatch will lead to a malfunction of the auto-marker which will in turn lead to losing marks.
2. In the file *Group<group_id>_dirty_data.csv*, any row can carry no more than one anomaly. (i.e. there can only be up to 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<group_id>_outlier_data.csv* except for outliers. Similarly, there are only coverage data anomalies (i.e. no other data anomalies) in *Group<group_id>_missing_data.csv*.
5. The retail store has three different warehouses in Melbourne (see *warehouse.csv* for their locations)
6. The retail store focuses only on 10 branded items and sells them at competitive prices
7. In order to get the item unit price, a useful python package to solve multivariable equations is numpy.linalg
8. The distance is calculated as Haversine Distance (with radius of earth = 6378 KM) like here.
9. The store has different business rules depending on the seasons to match the different demands of each season. For example, *delivery charge* is calculated using a linear model which differs depending on the season. The model depends linearly (but in different ways for each season) on:
    - Distance between customer and nearest warehouse
    - Whether the customer wants an expedited delivery
    - Whether the customer was happy with his/her last purchase (if no previous purchase, it is assumed that the customer is happy)
10. It is recommended to use **sklearn.linear_model.LinearRegression** for solving the linear model as demonstrated in the tutorials.

11. Using **proper data** for model training is crucial to have a **good** linear model (i.e. R^2 score over 0.97 and very close to 1) to validate the delivery charges. The better your model is, the more accurate your result will be.
12. To check whether a customer is happy with their last order, the customer's latest review is classified using a sentiment analysis classifier. ***SentimentIntensityAnalyzer*** from nltk.sentiment.vader is used to obtain the polarity score. A sentiment is considered positive if it has a 'compound' polarity score of 0.05 or higher and is considered negative otherwise. Refer to this link for more details on how to use this module.
13. If the customer provided a coupon during purchase, the coupon discount percentage will be applied to the order price before adding the delivery charges (i.e. the delivery charges will never be discounted).
14. The below columns are error-free (i.e. don't look for any errors in dirty data for them):
    - **coupon_discount**
    - **delivery_charges**
    - **The ordered quantity values in the shopping_cart attribute**
    - **order_id**
    - **customer_id**
    - **latest_customer_review**
15. 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.
16. 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.
17. **No libraries/packages restriction.**

Methodology (3/21)

The report **Group<group_id>_ass2_task1.ipynb** should demonstrate the methodology (including all steps) to achieve the correct results for all three files.

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 (1.5/21)

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.

Task 2: Data Reshaping (9/40)

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 normalisation/transformation (e.g. standardisation, min-max normalisation, 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 normally 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 the aim for this task is to prepare the data for a linear regression model, it's not building the linear model.** That is, you need to record all your steps from load the raw data to complete all the required transformations if any.

Input files	Submission
suburb_info.xlsx	Group<group_id>_ass2_task2.ipynb

You could consider the scenario of task 2 to be **an open exploratory project**: Jackie and Kiara have got some funding to do an exploratory consulting project on the property market. We wish to understand any interesting insights from the relevant features in different suburbs of Melbourne. Before we step into the final linear regression modelling stage, we wish to hire you to prepare the data for us and tell us if any transformation/normalisation is required? Will those data satisfy the assumptions of linear regression? How could we make our data more suitable for the latter modelling stage.

As **an exploratory task**, 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

Documentation (1.5/9)

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.

Task 3: Declaration and Interview(10/40)

Submission
Group<group_id>_ass2_task3.pdf Group<group_id>_AI_Records.docx/pdf (if applicable)

3.1 Generative AI Tools Declaration Form(Hurdle)

Task Details: For this task, all students **must** complete the Generative AI Tools Declaration Form and include the following statement clearly in the submission (you may refer to here for monash GenAI declaration format and generate your own one). After completing the form, download it as a PDF and include it in the submission.

If your group uses Generative AI tools, you must clearly document and **attach all conversation records** with these tools as part of your submission(in either pdf or docx format).

Requirement:

- This task **must be completed (HURDLE)**
- Conversation records should be **all English**
- **All** conversations need to be recorded. If you are unable to download or use multiple conversations, copy and paste them into **1 single file**.

Consequences of Missing Generative AI tools Declaration

Failure to submit a complete **Generative AI Tools Declaration Requirement** and **(if applicable) AI conversation records**, will result in not meeting the **hurdle** requirement for Assessment 2.

3.2 Interview(10/40 + Hurdle)

There will be an interview for your A2. The aim for the interview is to check your understanding of your entire A2 work and make sure all submissions are compliant with the academic integrity requirements of Monash.

Task Details:

- Time/Date: Week 12, during your allocated Applied sessions
- Form: 1-on-1, i.e., one TA interview 1 Group
- 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: You will be asked questions related to your A2 submission (code, methodology, specific functions, etc)
- Criterion: Please refer to A2 marking rubrics

Requirement:

- Mandatory attendance **(HURDLE)**
- All members need to show and sign the attendance sheet
- All members need to answer questions

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 applied session
- Forget to prepare for the interview, i.e. forget your own solution
- Too limited time to answer the questions properly
- Be too nervous to talk in English thus not properly answering the questions
- Direct use online resources without proper reference and do not understand the submitted work

Submission Requirements

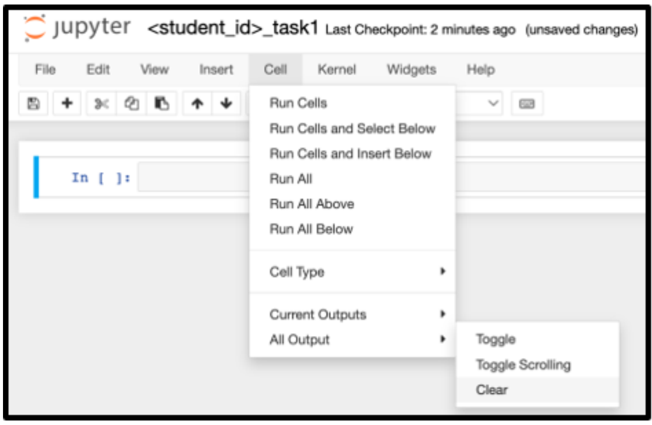
You need to submit the following **7-8 files**:

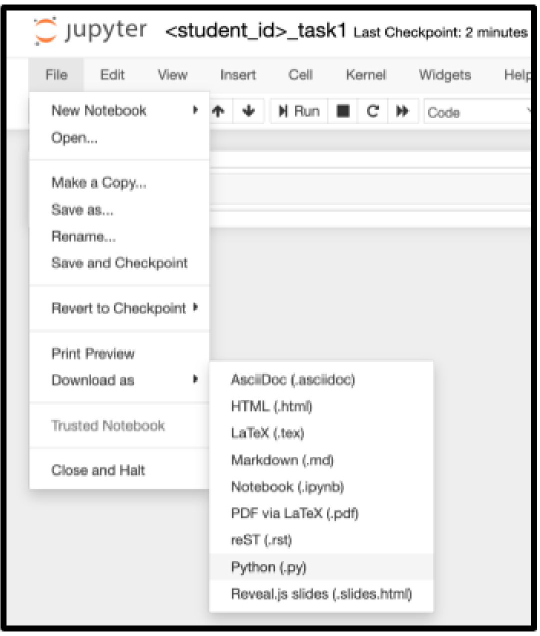
- A file named **Group<group_id>_dirty_data_solution.csv**: This file should contain the data records with all identified errors fixed.
- A file named **Group<group_id>_missing_data_solution.csv**: This file must contain the data records with all missing values imputed.
- A file named **Group<group_id>_outlier_data_solution.csv**: This file should include the remaining data records with all outliers removed.
- A Python notebook named **Group<group_id>_ass2_task1.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<group_id>_ass2_task1.py**: This file will be used for plagiarism check. (make sure the cell outputs are cleared before exporting)
- A Python notebook named **Group<group_id>_ass2_task2.ipynb**: In this notebook, provide a well-documented report that comprehensively demonstrates your solutions for task2. You need to clearly present your investigation, your EDA and other methodology through a well-organised reporting format, accompanied by relevant justifications, explanations and references (if any). As an open exploratory task, we expect to see you reach out to check the dataset, the model assumption and relevant research to enrich the report. 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<group_id>_ass2_task3.pdf:** This is the declaration form you need to submit.
- A file named **Group<group_id>_AI_Records.pdf (if applicable):** This is the AI communication history if you declare you used the generative AI in your A2.

Appendix

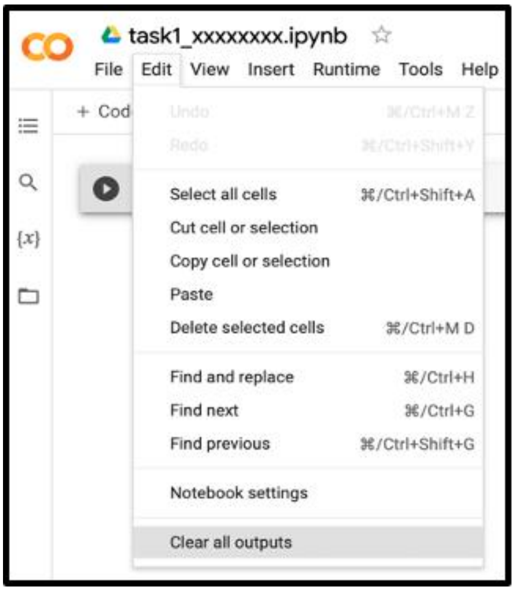
To generate a .py file, you need to clear all the cell outputs, and then download it.

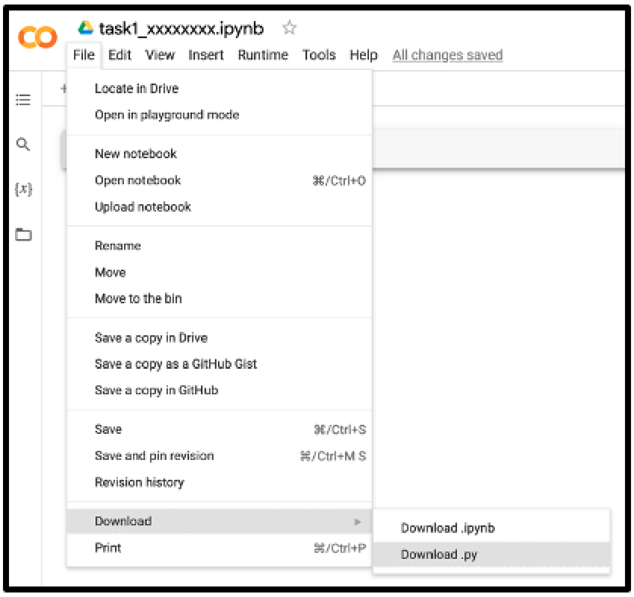
In Jupyter notebook:


jupyter <student_id>_task1 Last Checkpoint: 2 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
In [ ]:
Run Cells
Run Cells and Select Below
Run Cells and Insert Below
Run All
Run All Above
Run All Below
Cell Type
Current Outputs
All Output
Toggle
Toggle Scrolling
Clear



In Google colab:





Submission Checklist

- Please zip all the submission files for assessment 2 into a single file with the name **Group<group_id>_ass2.zip.** (any other format e.g. rar or 7z will be penalised)
- There are **7-8 files** in your compressed zip file
- **<group_id>** should be replaced with **your group id (without <>)** (it has 0 paddings ie.001,010…).'
- Make sure **BOTH** members of your group click the **'Submit'** button on Moodle
- 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 provide the declaration form and attend your allocated applied sessions on Week 12 to meet the HURDLE requirements

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.