FIT5196-S2-2025 assessment 1 (35%)

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

Due date: 23:55, Sunday, 14 September 2025

Exploratory Data Analysis (EDA) plays a critical role in the broader data analysis lifecycle. It acts as the bridge between raw data and actionable insights, allowing analysts to assess data quality, detect errors, and identify potential transformations needed to prepare the data for modelling. In the context of data wrangling, EDA guides the cleaning and reshaping process by revealing inconsistencies, duplicates, and missing values that must be addressed. Moreover, the insights generated through EDA inform data-driven decision-making by highlighting key metrics and revealing patterns that support evidence-based strategies. Effective EDA ensures that subsequent analysis is grounded in a solid understanding of the dataset and increases the reliability and interpretability of any models or decisions derived from the data.

Assessment Objectives

In this group assessment, students will undertake an Exploratory Data Analysis (EDA) project as part of the Data Wrangling unit. The primary objective of EDA is to develop a comprehensive understanding of a dataset before engaging in further modelling or machine learning processes. Through this process, students will identify trends, patterns, anomalies, and relationships within the data that may inform or inspire meaningful machine learning research questions.

You are expected to gain knowledge, skills and experience on

- **Data Familiarisation**: Gain a deep and structured understanding of the dataset's structure, content, and quality.
- **Basic Data Quality Assessment**: Identify missing values, duplicates, invalid values or inconsistencies that may affect analysis, but no need to fix them.
- **Visual Exploration**: Use visualisation techniques to uncover hidden structures, correlations, and distributions in the data.
- **Research Insight Generation**: Translate EDA findings into potential research questions that could be explored further using machine learning approaches.

To achieve the above objectives, the following step-by-step guide is recommended:

Step 1: Load, parse and merge data files

Each group has unique input data files to work with. Please note that using wrong input data files may lead to wrong findings from EDA and result in ZERO marks for the assessment. Please double check that you have the correct input data files naming with your group number.

Input data files:

- Group<group_number>.xml
- Group<group_number>.json

These two files contain photo data collected from Flickr. Each photo is encapsulated in a record that contains 18 attributes. Please check with the sample input files (sample_input) for all the available attributes.

Your tasks:

- Use appropriate Python codes to load data from both files;
- Convert loaded data into suitable objects for ease of manipulation;
- Inspect the structure and schema of data from both files to understand their components and relationship;
- Merge data into a single dataset using appropriate methods;
- Ensure the merged data is well structured and formatted correctly. Any non-English (other language, digital numbers should be keeped) content needs to be appropriately handled with reasons and justifications.
    - All text data in these five (5) attributes: *Title*, *City*, *Country*, *Tags*, *Description*, must be transformed into lowercase ('NaN' excluded);
    - No XML or JSON tags, no emojis, only valid UTF-8 characters are allowed (only Regular Expression is allowed to process this task);
    - Non-English characters need to be completely removed from all attributes (only Regular Expression is allowed to process this task).
    - All null values need to be represented by 'NaN'.
- Output the valid merged data to a CSV file named Group<group_number>_dataset.csv (see sample output file)

Note: You need to identify appropriate **Regular Expressions** to clear the text data as required. You need to have the exactly **same attributes** in your CSV file as that in the sample output file to avoid mark redundancy. Your Python solution will be tested using a test dataset (not provided to students) to check its correctness. That is, your code should work for any datasets with the same structure. A solution without proper debugging leads to mark redundancy. **We do not fix your errors.**

Step 2: Apply EDA on the merged dataset with visualizations

The primary goal of Exploratory Data Analysis (EDA) is to **gain an in-depth understanding of the dataset** to inform subsequent decisions, such as data preprocessing, model design, or hypothesis generation.

EDA helps you:

- Identify **structure and patterns** within the data
- Detect **errors, outliers, or anomalies**
- Assess the **quality and completeness** of the data
- Generate **insights and hypotheses** to guide further analysis, such as machine learning tasks

Your tasks:

- Understand the overall structure and dimensions of the merged dataset
- Apply univariate analysis
- Apply bivariate analysis
- Apply multivariate analysis
- Visualise the findings

By the end of this EDA process, you should be able to

- Describe the key characteristics of the dataset
- Explain the relationships and trends observed
- Identify data issues that must be addressed before modelling
- Create appropriate visualizations to present your understanding of the data and valuable findings

Step 3: Summarise key insights and research questions

To identify key insights and high quality research questions, you need to review the findings from EDA and translate them into machine learning problems. You can consider a number of possible machine learning questions, including

- Supervised learning - classification and regression
- Unsupervised learning - clustering, dimensionality reduction
- Semi-supervised learning
- Others - time series/sequential pattern detection, anomaly detection, etc.

At least ten (10) key insights and five (5) high-quality ML questions are required.

A well-formulated machine learning research question should be data driven, which is rooted in actual EDA findings, not speculative or hypothetical (need to have evidence from your EDA and corresponding visualizations).

The ML question should be carefully clearly defined, which specifies the input features and output variables (for classification questions as an example). You need to avoid any vagueness. For example, “Can we use ML to learn more?” is a low-quality question. You need to link your research questions with the available data. Do not try to predict user emotion without sentiment or behavioural data in the dataset.

The ML questions should be able to generate value, that is, by studying these questions, one can contribute meaningfully to the research domain or business context.

To evaluate the quality of your ML questions, you can ask yourself the following:

- What variables do I want to predict or understand?
- What predictors do I have available?
- Is there a meaningful relationship supported by the EDA?
- What ML technique is best suited to this problem?
- What would success look like in this task (e.g., high accuracy, good clustering)?

Note: You need to consider all the attributes in EDA, identifying findings and ML questions.

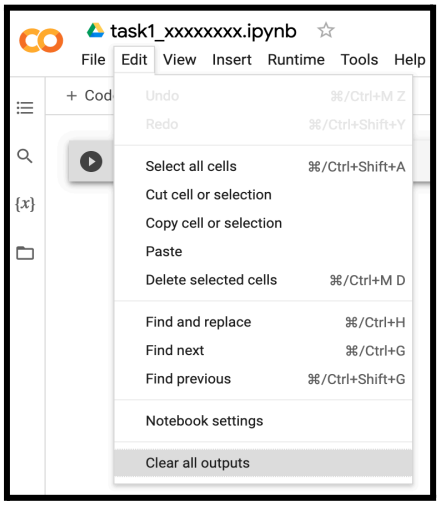
Submission Requirements

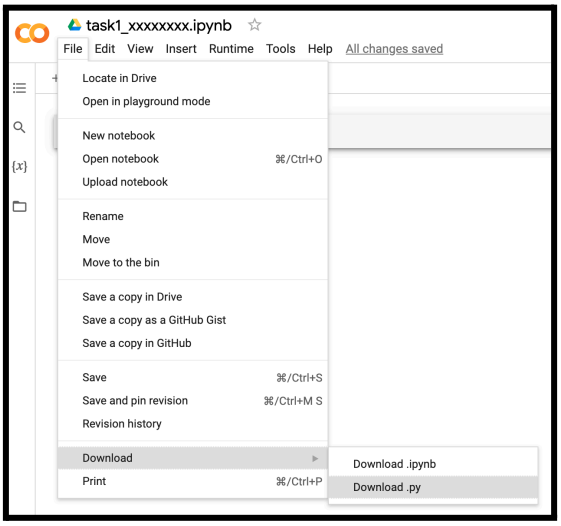
- **EDA methodology - 35%**
  - A **Group<group_number>_solution.ipynb** file that contains your EDA process
  - A **Group<group_number>_solution.py** file. This file will be used for plagiarism check (make sure you clear your cell output before exporting).
  - A **Group<group_number>_dataset.csv** file that contains the merged dataset
    Note: Your python codes should be able to produce the same merged dataset as submitted to avoid mark redundancy.
- **EDA report - 50%**
  - A **Group<group_number>_EDA.pdf** file that contains your EDA report including your design of the EDA, your findings, insights and ML questions with justifications and visualizations.
- **Video presentation - 10%**
  - A **Group<group_number>_presentation.mp4** file that is a video presentation (6-10 minutes) to effectively communicate your findings from EDA and the ML questions with justification and visualization. All group members need to present a part of the presentation.
- **Generative AI tools declaration and history documentation - 5%**
  - A **Group<group_number>_AI_declaration.pdf** file that contains the Generative AI Tools Declaration Form. (Note: This is a hurdle that you must provide the declaration. Failure to provide this declaration leads to failure of this assessment.)
  - A **Group<group_number>_AI_record.pdf** file that contains a complete AI conversation record (if applicable).

Note: You can use the **templates** provided for these submission files. Failure to submit a Generative AI Tools Declaration Form or the AI conversation record (if applicable) may lead to a suspension of Academic Integrity case. All submissions will be put through a plagiarism detection software that automatically checks for the 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.

Submission Checklist:

- Please zip all the submission files into a single file named **<group_number>_A1.zip** except the EDA report file **<group_number>_EDA.pdf**. That is, your submission should have two files:
  - **Group<group_number>_A1.zip** (6 files in this zip file)
  - **Group<group_number>_EDA.pdf** (if the report pdf file is placed inside the zip file, 5% penalty will be applied)
- Make sure both members of your group click the '**Submit**' button on Moodle
- Please strictly follow the file naming standard.
- Please make sure that your **.ipynb file** contains printed output, while your **.py file** does not include any output. You can use the options in the menu (in Colab as example):


task1_xxxxxxxx.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Undo ⌘/Ctrl+M Z
Redo ⌘/Ctrl+Shift+Y
Select all cells ⌘/Ctrl+Shift+A
Cut cell or selection
Copy cell or selection
Paste
Delete selected cells ⌘/Ctrl+M D
Find and replace ⌘/Ctrl+H
Find next ⌘/Ctrl+G
Find previous ⌘/Ctrl+Shift+G
Notebook settings
Clear all outputs


task1_xxxxxxxx.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
All changes saved
Locate in Drive
Open in playground mode
New notebook
Open notebook ⌘/Ctrl+O
Upload notebook
Rename
Move
Move to the bin
Save a copy in Drive
Save a copy as a GitHub Gist
Save a copy in GitHub
Save ⌘/Ctrl+S
Save and pin revision ⌘/Ctrl+M S
Revision history
Download
Print ⌘/Ctrl+P
Download .ipynb
Download .py