School of Computing, Engineering and Built Environment

Department of Computing

Artificial Intelligence and Machine Learning

Module Code: MMI226824

Coursework 1 Resit

Issued: 20th October 2025

This coursework comprises 50% of the overall mark for the module.

This coursework is to be submitted electronically via GCULearn, no later than:

Hand-in date: Friday 21st November 2025 (end of day)

An average student should be able to complete this assignment in about 20 hours of work.

Attention is drawn to the university regulations on plagiarism. Whilst discussion of the coursework between individual students is encouraged, the actual work has to be undertaken individually. Collusion may result in a zero mark being recorded for the coursework for all concerned and may result in further action being taken.

COURSEWORK 1

This module has two coursework components (CW1 and CW2), this is CW1 accounting for 50% of the module's mark. The pass mark for each coursework element is 50% (although a mark over the 45% threshold in one coursework can be compensated by a mark over 50% in the other, if the average results at least 50%).

In this coursework (CW) you will implement the first six steps of the Machine Learning Pipeline as shown in the book *"AI with Python"* by Artasanchez and Prateek [1], as shown in Figure 1.

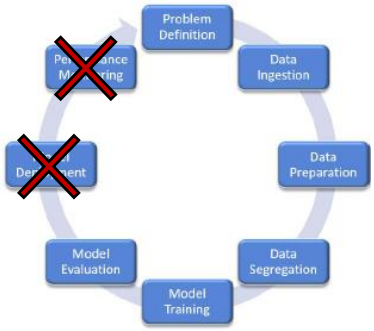

Figure 1: First six steps in a Machine Learning Pipeline [1]

Data

You will use a **modified** version of a Los Angeles Airbnb Data [2] which is posted on **GCU Learn** (***listings_cw.csv***). Please ensure to **use the modified version available on GCULearn**. The dataset includes 45,421 rental property listings located Los Angeles. Each entry typically contains:

- ***id***: Airbnb's unique identifier for the listing
- ***last_scraped***: UTC. The date and time of this listing was "scraped".
- ***host_since***: The date the host/user was created. For hosts that are Airbnb guests, this could be the date they registered as a guest.
- ***host_is_superhost***: Indicates whether the host is designated as a "Superhost"
- ***host_response_time***: The average time it takes for the host to respond to a new inquiry or request.
- ***host_response_rate***: The percentage of messages the host responds to.
- ***host_acceptance_rate***: The rate at which a host accepts booking requests.
- ***neighbourhood_cleansed***: The neighbourhood as geocoded using the latitude and longitude against neighborhoods as defined by open or public digital shapefiles.
- ***latitude***: Uses the World Geodetic System (WGS84) projection for latitude and longitude.
- ***longitude***: Uses the World Geodetic System (WGS84) projection for latitude and longitude.
- ***property_type***: Self-selected property type. Hotels and Bed and Breakfasts are described as such by their hosts in this field
- ***room_type***: All homes are grouped into the following three room types: entire place, private room and shared room
- ***accommodates***: The maximum capacity of the listing
- ***bedrooms***: The number of bedrooms
- ***beds***: The number of bed(s)
- ***bathrooms***: The number of bathrooms in the listing
- ***amenities***: A JSON list of amenities provided by the host.

- ***price***: daily price in local currency
- ***minimum_nights***: minimum number of night stay for the listing
- ***number_of_reviews***: The number of reviews the listing has
- ***reviews_per_month***: The average number of reviews per month the listing has over the lifetime of the listing.
- ***review_scores_rating***: Overall average review rating (typically on a 0–5 scale).

A full description of all variables is available in the official Inside Airbnb Data Dictionary [3], accessible at: Data Dictionary

Target tasks

This CW will require you to outline a detailed problem definition, ingest, prepare, and segregate the data, for subsequent training and evaluation of two **machine learning models** to **predict the review rating of Los Angeles Airbnb listings** based on a selection of relevant features.

You are required to submit a Jupyter Notebook based on the template available on GCU Learn (***CW1_template.ipynb***).

Jupyter Notebook Contents

The various sections in the notebook should include code, in-code comments and **appropriate Markdown cells** describing your approach chosen and discussing the results.

In detail, sections should include the following:

1. Introduction and Problem Definition

- A textual description providing an overview of the data.
- A discussion on why this problem is a regression problem.
- A detailed problem statement question as discussed in Lecture 3.

2. Data Ingestion

- Code to load the data into a suitable format to be used in the notebook
- A description of the statistical data types for each field in the file "*listings_cw.csv".* You should outline whether features are categorical or numerical and if they are ordinal, continuous, or discrete.
- **PLEASE NOTE:** Any version of this dataset which was found online will not align to the modified version used for this CW, and therefore marks may not be awarded!

3. Data Preparation

You should assume that exploratory data analysis has taken place and the following was concluded:

- No duplicated records detected.
- Missing data occur across multiple attributes: relatively small in ***host_since*** and ***host_is_superhost***; thousands of missing values in ***beds***, ***bedrooms***, ***bathrooms***, ***price***, and ***review_scores_rating***.
- Data types:
    - ***price*** stored as string due to currency symbols and commas.
    - ***last_scraped***, ***host_since*** stored as string rather than proper datetime formats.
    - ***host_response_rate***, ***host_acceptance_rate*** stored as string due to '**%**' sign.

- ***price*** vs ***review_scores_rating***: No clear linear relationship. Higher prices do not systematically yield higher ratings. The EDA shows high ratings (4.5–5.0) at low–to–medium prices (below $500) → *value-for-money* appears more influential than absolute price.
- The Rating System is Skewed to Perfection: The target variable, ***review_scores_rating***, is heavily left-skewed, with a median of 4.91 and a massive number of perfect 5.0 scores.

Your data preparation steps should therefore include the following:

- Drop the columns that you won't need for your analysis (**reasoning why**).
- Convert the ***price*** column from string to numeric by removing currency symbols and commas, then casting to float.
- Convert ***last_scraped*** and ***host_since*** into datetime format for further use, e.g., it may be useful to derive new temporal attributes such as ***host_tenure_days*** (the number of days the host has been active on the platform).
- Convert ***host_response_rate*** and ***host_acceptance_rate*** to numeric proportions by removing the percentage symbol and dividing by 100.
- Automatically fill in the missing values in the other columns that you are going to use, with values that would most closely mirror the actual missing values (**explain why you took a particular approach**).
- Construct a new feature to express *value-for-money*, for example: ***price_per_person*** = price / accommodates, or ***price_per_room*** = price / bedrooms (handling division by zero safely).
- A **justification (and potential application)** of whether you should use data binning or not
- Suitable encoding of the data

4. Data Segregation

- Code and justification for the selection and application of a suitable data split

5. Model Training

- Select two different Regression models and justify why they are suitable for the task.
- Only one of those models should be Tree-Based (e.g. Random Forest or Decision Tree).
- Application of those models with default parameters as a baseline on the data
- Utilisation of manual **or** automatic hyperparameter optimisation and justification of your choices to create "optimised" versions of each regression model

6. Model Evaluation

- Selection of appropriate metrics and a written outline why they are suitable for this task.
- A comparison of the baseline models to the "optimised" versions and an evaluation of the results.

7. Conclusion

- A conclusion and interpretation of the results and suggestion of potential improvements

You should provide **sufficient written documentation** in the notebook **Markdown cells** to show that you understood and have justified the steps that have been implemented. Marks may not be awarded if the code has insufficient explanations and the information provided is just contained within a few code cells.

The maximum word limit for all Markdown cells (excluding inline code comments or the "Sources" Markdown cell) is **1500 Words**.

You should use machine learning libraries such as ***scikit-learn*** to assist your implementation and training of algorithms. However, **copying code steps** from external sources which have used the same, or similar dataset is not permitted and may lead to zero marks being awarded for this particular section. You should write the code yourself and **demonstrate your understanding** using the textual explanations you provide.

Plagiarism

You should also pay attention to the university's codes and practices[1] as well as their plagiarism regulations[2]. Any kind of content (images, text, code) that was copied from any source and used in your coursework **without acknowledging of the source** is bad academic practice and could fall under plagiarism.

The discussion of coursework between students is encouraged, but **the work must be undertaken individually**. Collusion (copying work between students) may result in a zero mark being recorded for everyone involved and further action being taken.

Sources

To avoid plagiarism or poor academic practice, you need to ensure that you specify where you obtained any material you use and how you have modified it.

This MUST include **specific web addresses** (not just *google.co.uk* or similar).

A template for this is included on at the very bottom of the supplied template notebook and shown in Figure 2.

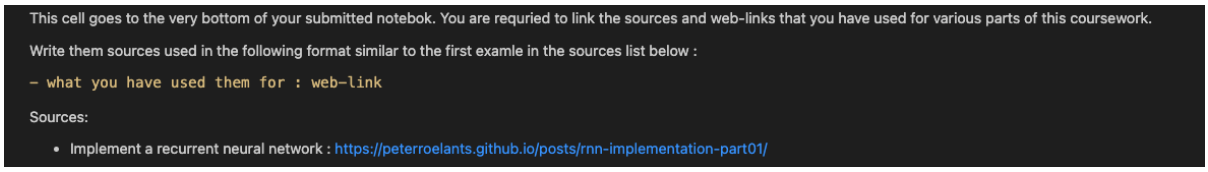

Figure 2: "Sources" markdown cell to be included at the very end of the submission

Coursework Submission

You are expected to submit your Jupyter Notebook with your code and markdown cells, using the following **naming** scheme:

CW1_LastName_FirstName_StudentID.ipynb

For instance, if your name is Nicola Sturgeon with the StudentID S1234567, name the file:

CW1_sturgeon_nicola_S1234567.ipynb

Coursework files are submitted using GCU Learn.

Marking Scheme

The marking scheme which will be used to assess the coursework is appended at the end of this document.

[1] https://www.gcu.ac.uk/academicquality/regulationsandpolicies/universityassessmentregulationsandpolicies/
[2] https://www.gcu.ac.uk/library/smile/plagiarismandreferencing/

References

[1] Artasanchez, Alberto, and Prateek Joshi. Artificial Intelligence with Python: Your complete guide to building intelligent apps using Python 3. x. Packt, (2020)

[2] Inside Airbnb, 2025, "Los Angeles, California, United States: Listings", Inside Airbnb. [Online]. Available: https://insideairbnb.com/get-the-data/. [Accessed: 14-Oct-2025].

[3] Airbnb, “Inside Airbnb: Get the Data,” Inside Airbnb, 2022. [Online]. Available: http://insideairbnb.com/get-the-data.html. [Accessed: 14-Oct-2025].

Artificial Intelligence and Machine Learning (MMI226824) CW1 Marking Rubric

Criteria	Exceptional (80-100%) Demonstrates ***exceptional*** ability, skills and behaviours across specified characteristics.	Excellent (70-79%) Demonstrates ***mostly excellent*** ability, skills and behaviours across specified characteristics.	Good (60-69%) Demonstrates ***overall good*** ability, skills and behaviours across specified characteristics.	Satisfactory (50-59%) Demonstrates ***overall satisfactory*** ability, skills and behaviours across specified characteristics.	Marginal Fail (40-49%) Demonstrates ***overall poor*** ability, skills and behaviours across specified characteristics with ***some satisfactory*** elements.	Clear Fail (0-39%) Demonstrates ***consistently poor*** ability, skills and behaviours across specified characteristics with ***limited or no satisfactory*** elements.
Introduction and problem definition (10 marks) *Clear introduction and problem definition. Clear explanation of why each task is a regression problem.*	Provides an exceptionally clear and comprehensive overview of the data and presents a detailed and insightful problem statement, demonstrating deep understanding.	Provides a clear and thorough overview of the data and presents a detailed problem statement, demonstrating strong understanding.	Provides an adequate overview of the data with a clear problem statement, meeting the basic requirements of the task.	Provides a basic overview of the data and a basic problem statement as described in the booklet, with some additional contribution but lacking depth.	Provides an incomplete or unclear overview of the data and problem statement, with some relevant content but significant omissions.	Fails to provide an adequate overview of the data or problem statement; lacks clarity and relevant content.
Data ingestion, preparation and segregation (35 marks) *Data ingestion, preparation steps and segregation have been included and clearly explained.*	Demonstrates thorough checks and cleaning, handles missing values and encoding logically yet creatively, and applies feature engineering that clearly improves model potential. Provides a well-justified strategy for splitting the data (e.g. train/test, cross-validation).	Performs correct data loading and basic cleaning, deals with missing values in a suitable manner, shows a sound approach to feature engineering (e.g. encoding/binning), and presents a clear rationale for train/test splitting.	Loads and mostly cleans the data, with some basic feature engineering. Carries out train/test splitting with a brief explanation.	Takes incomplete steps to process and clean data, implements minimal feature engineering, and gives an unclear or perfunctory splitting method.	Shows poor or inconsistent data handling, leaves missing/invalid values unaddressed, and lacks a clear strategy for splitting the data.	Fails to perform meaningful data ingestion or cleaning, conducts no feature engineering, and shows no evidence of any data splitting.
Model Training and Evaluation (40 marks) Select at least two models, including one tree-based method (e.g. Random Forest or Decision Tree). Compare baseline and optimised versions using appropriate metrics and a clear performance analysis.	Chooses at least two models, both highly appropriate for the task, including one tree-based approach. Provides a strong comparison between baseline and optimised versions, using highly relevant metrics and offering thorough interpretations of results.	Selects two models that suit the data and problem, compares baseline versus optimised models, and applies the chosen metrics correctly with clear explanations.	Picks models that are mostly appropriate. Shows some effort to compare baseline and improved versions. Uses standard evaluation metrics, though the discussion is somewhat brief.	Offers an unclear or poorly justified model selection, makes little distinction between baseline and tuned models, and provides insufficient discussion of metrics.	Chooses models unsuited to the task, with no real optimisation or comparison. Employs inappropriate or missing metrics.	Fails to attempt valid modelling or evaluation. Does not provide meaningful metrics or results.
Discussion and Conclusion (15 marks) *Detailed and meaningful concluding statements.*	Provides exceptionally detailed conclusions for each task, with insightful interpretation of the results and thoughtful suggestions for potential improvements. The analysis demonstrates critical reflection and a deep understanding of the implications of the findings.	Provides detailed conclusions for each task, with clear interpretation of the results and suggestions for improvements. The analysis is well-thought-out and demonstrates a solid understanding of the results.	Provides general conclusions for each task, with some interpretation of the results. Suggestions for improvements are basic or implied, with minimal critical discussion.	Summary of results for each task with little critical discussion. The suggestions for improvements are vague or minimal.	Conclusions are incomplete or lack clear interpretation of the results. No possible proposed improvements.	Fails to provide meaningful conclusions, interpretation of results, or suggestions for improvements.