Assessment 1 Brief

Module Title	Advanced Data Analytics and Machine Learning	
Module Code	MAN-40389	
Assessment Type	Report	
Assessment Title	Individual report	
Weighting (% of module mark)	40%	
Assessment Length (word count or equivalent)	2000 words (+/- 10%)	
Submission Deadline (date and time)	Friday, 12 December 2025	1pm
Format of Submission	Via Turnitin	
Type of GenAI Use Permitted in Assessment (please refer to the GenAI descriptor within the 'Use of AI' section	GenAI-assisted editing and proofreading Gen-AI-Assisted Reviewer GenAI-Assisted Idea Generation GenAI-Assisted Presentation	
Feedback Release Date	30/01/2026	
Staff contact details	Dr. Rotimi Ogunsakin r.s.ogunsakin@keele.ac.uk	

Assessment Details

Submission

The electronic version of your report (saved as a single .pdf file and a zipped copy of your code) is to be submitted through KLE/Turnitin.

Format

Conduct your coding and analysis using appropriate functionalities provided in Python (ideally the latest version). Then summarise your work using Microsoft Word. All main results have to be included in your report, as further detailed in the brief below. Keep your writing as brief as possible to around 2,000 words, excluding figures, tables, references and appendices.

Figures and tables must be numbered and referred to in the text. Copy and paste the Python code at the end of the report as an appendix (and submit a zipped copy on KLE). You can print the code to a PDF file and then append the PDF file to your final PDF report file. There is no limit on the length of your code. For the report, use a minimum font size of Arial 11pt. For the code, you can use any readable font, but the font size should be 11pt. Handwritten submissions will NOT be accepted.

While the brief below is broken down into questions/tasks, this is done to help guide your analysis and understanding of the tasks that must be undertaken. It does not determine the

organisation of your report. The report should not quote the questions/tasks in the brief. It should be a coherent piece of work, where you describe the problem at hand and explain the steps taken to analyse it and solve it. However, regardless of your report structure, it should contain an introduction section and a conclusion section.

Assessment

The final coursework mark will count 40% towards your overall module mark; the remaining 60% will be made up of the grades from your group coursework, of which your individual coursework is a fundamental part. Ensure that the material presented in your report is correct and reproducible, and that the Python code is correct and readable. In terms of professional presentation, pay attention to details such as layout, font size, adhering to the word-count limit, spelling, grammar, the use of tables and charts, and maintaining a consistent number of decimal places.

Assessment Task: Smart Energy Grid

The National Energy Consortium (NEC) must meet an electricity demand scenario every day. These can include, for example, high-peak weekday loads, volatile evening ramps, or holiday base-load profiles. For completing each demand scenario, NEC must choose one out of sixty-four (64) power plants (Coal, Gas, Hydro, Nuclear, Wind, Solar) that provide the generation required to serve the demand. The final cost per MWh of serving a demand depends on how effective the chosen plant is for that specific scenario. Unfortunately, estimating this cost in advance requires significant resources (engineering simulations, market forecasts, etc.). Specifically, running detailed engineering simulations to estimate these costs is a resource-intensive process. Therefore, NEC has commissioned your Analytics firm to develop a machine learning (ML) approach for selecting plants given a new demand scenario, and your line manager has assigned you to handle the task.

To complete your assignment, NEC has provided you with three datasets:

- **demand.csv** – a CSV file that contains one row per demand scenario and one column per demand feature (DF1, DF2, DF3, …, DF12), plus categorical descriptors (e.g., region/day type). Each scenario is uniquely identified by a Demand ID (D1, D2, D3, …).
- **plants.csv** – a CSV file that contains one row per plant and one column per plant feature (PF1, PF2, …, PF18), plus categorical descriptors (e.g., plant type/region).

Each plant is uniquely identified by a Plant ID given in the first column of the file (P1, …, P64).

- **generation_costs.csv** – a compressed CSV file that contains data collected and/or estimated by NEC about the cost of serving a demand scenario with each plant. Each row gives the cost value (in USD per MWh) of one scenario served by one plant.

Goal and Evaluation

Your goal is to provide NEC with an ML model that, given the features of a demand scenario, selects one plant among the 64 plants available. The selection error made by the ML model will be measured as:

$$\text{Error}(d) = \min\{\, c(p,d) \mid p \in P \,\} - c(p'_d, d) \qquad (1)$$

Where d is a demand scenario, P is the set of 64 plants, p'_d is the plant chosen by the ML model for demand d, and $c(p,d)$ is the cost if demand d is served by the plant p. That is, the **Error** is the difference in cost between the plant selected by the ML model and the **actual best** plant for this scenario.

Given a set of demand scenarios for validation, we can assign a score to the ML model by computing the root mean squared error (RMSE) over the scenarios:

$$\text{Score} = \sqrt{\frac{1}{D}\sum_{d=1}^{D} \text{Error}\,(d)^2} \qquad (2)$$

Where D is the number of scenarios and $\text{Error}(d)$ is given in Eq. (1). This score should be minimised.

Task Description

1. Data Preparation

In the data preparation stage, please adopt a comprehensive approach, addressing the following areas:

1.1. Handle any missing data by identifying and managing incomplete records or missing values to ensure the dataset is ready for further analysis.

1.2. Perform relevant feature selection by determining which features are most important for your goal. Additionally, apply feature scaling to normalise the data and ensure all features contribute equally to the model, avoiding biases due to differing ranges.

1.3. Focus on identifying the top-performing plants for each demand by analysing the cost data. Since not all plants perform equally well, you should remove the worst-performing plants from the dataset to make the following tasks more manageable.

2. Exploratory Data Analysis (EDA)

Consider using various EDA techniques to explore the distribution of feature values, cost data and plant performance:

2.1. Use EDA methods to analyse the distribution of demand features and interpret emerging patterns, such as identifying correlations, clusters, or outliers in demand characteristics

2.2. Analyse the cost data to identify patterns and good combinations of demand scenarios and plants, such as investigating whether specific plant types or regions are more cost-effective under certain demand contexts.

2.3. Explore how errors (Eq. 1) are distributed across different choices of plants, identifying key trends or outliers in their performance. Calculate the RMSE (Eq. 2) of each plant for all scenarios and use them to motivate the need for ML models.

3. ML Model Fitting and Scoring (Train/Test Split)

In this step, you will need to split the data into input features and output values. In addition, you will create training and testing datasets. One particularity of this case study is that the same demand appears in both training and testing, thus data must be grouped by Demand ID when creating training and validation sets.

3.1. Combine the demand features, plant features and costs into a single dataset with the following columns (the plant data may need to be transposed):

Demand ID	DF1	…	DFn	PF1	…	PFm	Cost_USD_per_MWh

Each **Demand ID** (and its corresponding feature values) will appear multiple times (once for each plant). The number of rows of this dataset must be the same as the number of rows in ‘generation_costs.csv’. Then split the dataset

into **X**(DF1,…,DFn, PF1,…,PFm), **y** (Cost_USD_per_MWh), and Groups (Demand ID). Note that there will be **fewer demand features (DFi)** than in the original data because of the features removed in Step 1 above.

3.2. Randomly select 20 unique 'Demand ID' values from Groups (let's call this subset TestGroup). Split the dataset created in Step 3.1 into four different datasets (X_train, X_test, y_train and y_test) as follows. X_train, X_test contain only the columns DF1,…,DFn, PF1,…, PFm; whereas y_train, y_test contain only the column Cost_USD_per_MWh. Also, X_test, y_test contain only rows whose Demand IDs are in TestGroup; whereas X_train, y_train contain the remainder rows.

3.3. Train a regression ML model of your choice using X_train and y_train. Score the performance of the model on X_test and y_test using the score function of the model. When using a regression model, the ML model predicts the cost of each plant for a given demand; thus, the plant selected by the ML model in Eq. (1) is the one with the lowest predicted cost. Note that Eq. (1) uses the actual cost of the selected plant, not the predicted cost.

3.4. Using Eq. (1) calculate the Error of the trained model for each demand in TestGroup and using Eq. (2), calculate the RMSE score.

3.5. Compare Errors and RMSE to the values obtained in Step 2.3 and summarise your conclusions.

4. Cross-Validation (Leave-One-Demand-Out)

4.1. In Step 3, we have done a simple train-test split grouped by Demand ID. Now, you will perform a Leave-One-Out cross-validation, grouped by Demand ID. Fortunately, Scikit-learn already provides a Leave-One-Group-Out cross-validation (LOGO) type. Starting from the datasets created in Step 3.1, perform a Leave-One-Group-Out cross-validation using the Groups dataset - to define the groups (read the documentation and examples of Leave-One-Group-Out cross-validation to learn how to specify the groups).

4.2. In addition, we wish to use our own score function (Eq. 1) within the cross-validation for scoring. You will need to use the scikit-learn function 'make_scorer', as shown in the lectures, to create a 'score' function that can be passed to scikit-learn's cross-validation function.

4.3. Report the scores returned by cross-validation and compute the RMSE of the scores returned. Compare these results with those obtained in Steps 2.3 and 3.5.

Hint: If your computer has multiple CPUs, use the n_jobs parameter to use all CPUs in parallel.

5. Hyperparameter Optimisation

5.1. Looking at the documentation for the ML model of your choice, devise a hyper-parameter grid (`param_grid`) appropriate for your ML model and apply `GridSearch` to identify the best hyper-parameter configuration starting from the datasets from Step 3.1. As in Step 4, you must set up `GridSearch` to use Leave-One-Group cross-validation and the scoring function from Eq. (1).

5.2. Report the best hyper-parameter configuration found by GridSearch and its RMSE. Compare the results with those obtained in Steps 2.3, 3.5 and 4.

6. Model Comparison

6.1. Explore additional regression models from scikit-learn and select one that you believe could perform well (justify your choice in the report).

6.2. Apply the selected model and repeat Steps 3 to 5. Compare the results between the models and discuss the strengths and weaknesses of each approach, such as accuracy, computational cost, stability, and interpretability.

6.3. Feel free to experiment with more regression models for comparison, but keep the word limit in mind.

Hints:

A. Implement all steps in Python. Use Python to compute statistics, generate tables and plots. Do NOT use Excel or any other software.

B. Use different Python files to implement independent steps that read/write intermediate data files. For example, step1-preparation.py will create a new set of files that will be read by step-2-exploratory.py and step-3-train-test.py. This is only an example; it is part of the coursework to think and decide on the best way to divide tasks between Python files.

C. All steps must be reproducible without requiring edits to the Python files. That is, if we provided you with new datasets with the same filenames, you should be able to execute all Python files in order and obtain new results without needing to modify the code.

D. This coursework requires doing your own research**.** The coursework can be completed using only the Python packages: Numpy, Pandas, Matplotlib and scikit-learn. However, you DO need to look up in the documentation of the relevant package how to perform the various steps and understand the documentation using the knowledge acquired in the lectures and labs.

E. Document your code.

Deliverables

- **Report (~2,000 words)**: Clear documentation of all stages (data preparation, EDA, modelling, evaluation, comparison). Include tables, charts, and visualisations where appropriate.
- **Code repository**: Well-structured, documented Python scripts (e.g., `step1_preparation.py`, `step2_eda.py`, etc.) that reproduce results end-to-end when provided with the datasets.
- **Appendix**: Hyperparameter grids, additional results, and error analyses (optional, not included in word count).

Marking Scheme (40% of module)

- Q1 Data Preparation — 5%
- Q2 EDA — 5%
- Q3 Train/Test Modelling — 15%
- Q4 Cross-Validation — 15%
- Q5 Hyperparameter Optimisation — 15%
- Q6 Model Comparison — 15%
- Professional presentation, writing and clarity — 15%
- Conciseness, correctness and clarity of code — 15%

Late submission and academic integrity rules follow the School policy.

Module Learning Outcomes:

ILO1: critically evaluate machine learning algorithms for different types of business problems, selecting the most appropriate method for various datasets.

ILO2: analyse complex datasets to identify patterns and relationships using advanced data analytics tools and techniques.

ILO3: develop predictive models that solve real-world business challenges, justifying the selection of specific algorithms and parameters.

ILO4: evaluate the performance of data-driven models through statistical measures, refining models to enhance accuracy and efficiency.

ILO5: design and implement machine learning pipelines to solve complex business problems, integrating multiple models and tools.

Assessment Criteria:

MARKING CRITERIA

Your work will be marked using the following criteria and assessment expectations outlined within the session.

Criterion	Excellent (70–100%)	Good (60–69%)	Satisfactory (50–59%)	Needs Improvement (<50%)
Q1. Data Preparation (5%)	Missing data handled systematically with justification; feature selection and scaling are rigorous; plant pruning well-reasoned and supported by evidence.	Most missing values handled; basic feature scaling/selection; plant pruning attempted with some justification.	Minimal treatment of missing data; feature scaling/selection superficial; plant pruning arbitrary or weakly justified.	Little/no handling of missing data; no clear feature selection or pruning; dataset not fit for modelling.
Q2. Exploratory Data Analysis (5%)	Thorough analysis of demand features and cost patterns; insightful visualisations; clear identification of baselines.	Good analysis with some visualisations; baselines computed but discussion limited.	Basic descriptive stats; visualisations limited or poorly explained; baselines incomplete.	Minimal or no EDA; baselines missing or incorrect.
Q3. Train/Test Modelling (15%)	Train/test split grouped correctly; model choice appropriate and justified; results critically evaluated against baselines; discussion shows deep insight.	Grouped split implemented; model appropriate; results compared to baselines but limited analysis.	Split attempted but may not fully respect groups; model choice acceptable but poorly justified; limited comparison to baselines.	Incorrect or missing grouped split; model choice inappropriate; no comparison to baselines.
Q4. Cross-Validation (15%)	LOGO CV implemented correctly with custom scorer; results clearly reported; strong comparison to train/test and baselines; robust insights.	LOGO CV implemented; results reported but analysis limited; some comparison to baselines.	CV attempted but with errors (e.g., wrong grouping); results incomplete or weakly analysed.	No evidence of LOGO CV; results missing or invalid.
Q5. Hyperparameter Optimisation (15%)	Well-designed param grid; GridSearch with LOGO CV implemented correctly; best config reported with strong justification; results critically compared to previous steps.	Param grid sensible; GridSearch run with minor issues; best config reported; some comparison to earlier results.	Param grid limited or poorly justified; GridSearch attempted but flawed; results weakly compared to earlier steps.	No attempt at hyperparameter optimisation or implementation incorrect.

Q6. Model Comparison (15%)	Two or more models implemented; strong justification for selection; results compared rigorously (accuracy, cost, interpretability, stability); critical discussion.	Two models implemented; justification present; results compared but discussion limited.	Two models mentioned but not fully implemented; weak justification; minimal comparison.	Only one model attempted; no justification or comparison.
Presentation & Writing (15%)	Report exceptionally clear, concise, and professional; excellent visuals/tables; logical structure; correct technical writing and referencing.	Report well-structured and clear; good use of visuals; minor issues in clarity or referencing.	Report understandable but uneven; limited visuals; technical writing weak.	Report unclear or incomplete; poor structure; visuals missing; significant writing errors.
Code Quality & Reproducibility (15%)	Code modular, concise, well-documented; reproducible end-to-end; efficient use of NumPy/pandas/scikit-learn.	Code runs; some documentation; minor duplication or inefficiency.	Code partially runs; little documentation; inefficient or repetitive.	Code not working or irreproducible; little/no documentation.

Feedback to Students:

You will receive feedback via Turnitin with 15 working days. Comments will be provided both in the overall feedback box. You can access you feedback by viewing your original submission on Turnitin when the marks and feedback have been released. Please contact the module leaders if you have any issues accessing your feedback.

Inclusive Practice:

The assessments for this module have been designed in accordance with the Keele Inclusivity Education Framework

The assessments for this module provide you with the opportunity to demonstrate your learning in various ways and across two assessment opportunities. You will be provided with an assessment brief and dedicated assessment briefing sessions for each assessment to outline key tasks, expectations, and marking criteria. You will have the opportunity for feedback on assessment plans and extracts of work produced to support you in successfully completing assessments.

Use of Artificial Intelligence (AI):

You can use assistive AI tools to check spelling, grammar and punctuation (i.e., use of in-built spell checkers) in all written assessments excluding examinations and class tests, unless advised otherwise. You are able to use certain AI tools with appropriate referencing requirements as outlined below to help complete assessments and all use of AI should be used in line with Keele's Code of Practice on Academic Misconduct (see link in the following

section below). You should not submit large chunks of work that has been developed by AI tools unless otherwise instructed by the module leader.

Type of AI Use	Description	Referencing Required
GenAI-assisted editing and proofreading	You may use GenAI to edit your work (e.g. enhancing phrasing, tone and sentence structure). You are not permitted to use GenAI to generate new content.	At the end of your work and before your reference list you must acknowledge your use of GenAI. For example: "*I acknowledge the use of Microsoft Copilot (version GPT-4, Microsoft, https://copilot.microsoft.com/) to edit my sentence structure and phrasing".*
Gen-AI-Assisted Reviewer	You may use GenAI to act as a reviewer to give academic feedback on a draft of your assessment (e.g. identifying areas for improvement). You are not permitted to use GenAI to generate new content.	At the end of your work and before your reference list you must acknowledge your use of GenAI. For example: "*I acknowledge the use of Microsoft Copilot (version GPT-4, Microsoft, https://copilot.microsoft.com/) to review a draft of my work and provide feedback".*
GenAI-Assisted Idea Generation	You may use GenAI to assist you to develop ideas (e.g., headings, bullet points, outline plans, basic structure), identify potential themes or relevant journal articles to assist you in the preparation of your assessment. However, you are not permitted to include GenAI generated content in your final assessment submission.	At the end of your work and before your reference list you must acknowledge your use of GenAI. For example, "*I acknowledge the use of Microsoft Copilot (version GPT-4, Microsoft, https://copilot.microsoft.com/) to identify key themes relevant to my presentation".*
GenAI-Assisted Presentation	You may use GenAI to assist you to create images, diagrams, or other media to include in your assessment submission, including in presentations. However, presentations must be your own work and not the direct output of GenAI software.	You must clearly reference any AI-generated parts of the assessment submission as per the guidance on referencing Generative AI via Cite them Right. At the end of a presentation you must clearly state how and where GenAI has been used in that presentation. For example, "*I acknowledge the use of Microsoft Copilot (version GPT-4, Microsoft, https://copilot.microsoft.com/) to create images used on slides xx and xx in this presentation"*

Academic Misconduct:

Academic misconduct is doing something that could give you an unfair advantage in an assessment. It includes, but is not limited to, the following: plagiarism; collusion; contract cheating; cheating in an examination; falsification of data or sources; falsification of official documents or signatures. The University treats academic misconduct very seriously and penalties will be given for proven cases, including termination of studies in serious cases. It is therefore very important that you understand how to prepare and take assessments honestly. In order to assist you with this there are various resources and help available both as part of your programme of study and also centrally. For more information please visit: https://www.keele.ac.uk/students/academiclife/appeals-complaints-conduct/studentacademicconduct/

Academic Skills Support:

The Academic and Digital Skills team provide a range of additional online resources (e.g., study guides, Sways, Podcasts, workshops etc) to help you with your academic work and assessments. You can find more information here.

Presentation, Referencing And Content

- The report is a maximum of 1000 (+/- 10%) words in length, excluding executive summary and final reference list. Images, diagrams and small tables (i.e. tables stating basic figures or short bullet points) do not count towards the word count. Extensive tables, in-text references, headings and subheadings do count towards the word count.

- You must use a report format for the assignment. A report format consists of the key headings and subheading which are normally numbered.

- Use a combination of academic sources and credible websites to inform your judgement and assessment.

- Make use of supporting visuals, diagrams and charts to communicate your knowledge and idea and make sure they are clear to read and correlate with any discussion points

- Put your student number as the assessment title on your submitted work. Do not put your name on any part of your submission.

- Harvard referencing format must be used. All reference sources cited in the reference list must be included in the main body of the presentation (guidance and examples provided on the KLE).

- University policy will apply where cases of copying and plagiarism are suspected

- Please note that the inappropriate use of a proof-reader, as outlined in Section 5 of the University's proofreading guidance (http://www.keele.ac.uk/studentacademicconduct/) could be classed as academic misconduct.

Submission

- You must submit your work in a single document. You are strongly advised to submit earlier, if possible, to avoid any last-minute problems or complications.

- You must submit the correct version of your work as you can make only one submission.

- You must submit in electronic form to the relevant Turnitin Assessment Dropbox. Submit your work whilst logged in to the KLE with your own Keele Student identity. Instructions on how to submit can be found at Turnitin Student Guides (sharepoint.com).

- It is your responsibility to download the digital receipt as proof of your submission (see step 4 at Turnitin Student Guides (sharepoint.com).

- Work may be submitted at any time before the deadline.

Problems With Electronic Submission

- If the Turnitin service fails on the day of the deadline, the Keele Business School Office will inform you of the details by email message sent to your Keele Student email account.

- You should continue to monitor your email account for a message informing you that the Turnitin service has been restarted (the '2nd Message').

All submissions received no later than 12 hours after the time at which the 2nd Message was sent (as shown in the Keele Business School Office) will be deemed to be within the submission deadline.

END.