Assessment Brief 2025-26

Module Leader: Dani Papamaximou		Level: 7	
Module Name: INTRODUCTION TO PROGRAMMING FOR BIG DATA		Module Code: 55-710252-BF-20256	
Assignment Title: Recommendation Engine and Exploratory Data Analysis			
Individual Assignment	Weighting: 100%	Magnitude: 20 min presentation and 3500 words report	
Submission date/time: Tuesday, 16th December 2025, 15:00	Blackboard submission: Yes Turnitin submission: Yes	Format: *.py, .mp4, .pdf. ,. Rmd*	
Planned feedback date: 20th of January 2026	Mode of feedback: Written feedback		
In this assessment students are asked to consider:	Inclusivity and accessibility		Yes
	Sustainability		Not applicable

Module Learning Outcomes

- LO1: Select appropriate programming techniques and data structures to design and implement effective software systems.
- LO2: Apply relevant program design strategies and follow the software development life cycle to the implementation of software applications
- LO3: Design and implement domain-specific software.

Indicative employability skills you will develop during this module:

1. Analytical and Problem-Solving Skills
2. Technical Programming Competence
3. Software Development and Project Management Skills

Assessment Brief

With the overwhelming volume of online content and increasing ubiquity of Internet-enabled devices, the pervasive use of the Web for content sharing and consumption has become our everyday routine. However, people seeking online access to content or items of interest are becoming more and more frustrated due to information overload. Deciding on the content to consume from a deluge of available alternatives becomes increasingly tricky. In this regard, an online content streaming company provides movies, TV shows, music and other services to millions of online users. The company makes profits from selling interesting content/ services

of all genres to their customers. In the last couple of years, the fortune of the company has suffered due to dwindling sales, possibly occasioned by the non-digitalisation of their recommendation services. The management decided to improve their sales by developing an intelligent service recommendation engine for their online platform. To achieve this, the company has hired you as their data engineer to design and develop an intelligent recommendation engine and report on the quality of the data used in this engine. One of the key contents provided by the company is an online music streaming service.

The company has provided you with a dataset containing music tracks of various artists. Your tasks are:

1. To deliver the recommendation engine, working with the dataset to analyse, design, and implement the system.
2. To deliver an Exploratory Data Analysis report for the Data Analysts of the company. This report will provide insights into the quality of the data and help the analysts collect data more efficiently.

The dataset

In the assignment folder, you will find the dataset, “data.csv” (See Figure 1) . You are only required to extract the following features: “acousticness”, “artists”, “danceability”, “energy”, “id”, “liveness”, “loudness”, “name”, “popularity”, “speechiness”, “tempo”, and “valence”.

valence	year	acousticness	artists	danceability	duration_ms	energy	explicit	id	instrumentalness	key	liveness	loudness	mode	name	popularity	release_date	speechiness	tempo
0.0594	1921	0.982	['Sergei Rach	0.279	831667	0.211	0	4BJqT0PrAfrx	0.878	10	0.665	-20.096	1	Piano Concerto No. 3 in D Minor, O	4	1921	0.0366	80.954
0.963	1921	0.732	['Dennis Day'	0.819000000	180533	0.341	0	7xPhfUan2yN	0	7	0.16	-12.441	1	Clancy Lowered the Boom	5	1921	0.415	60.936000000
0.0394	1921	0.961	['KHP Kridhar	0.328	500062	0.166	0	1o6I8BglA6ylDMD	0.913	3	0.101	-14.85	1	Gati Bali	5	1921	0.0339	110.339
0.165	1921	0.967	['Frank Parke	0.275	210000	0.309	0	3ftBPoC5vPa	2.77e-05	5	0.381	-9.316	1	Danny Boy	3	1921	0.0354	100.109
0.253	1921	0.957	['Phil Regan']	0.418	166693	0.193	0	4d6HGyGT8e	1.68e-06	3	0.229	-10.096	1	When Irish Eyes Are Smiling	2	1921	0.038	101.665
0.196	1921	0.579	['KHP Kridhar	0.697000000	395076	0.346000000	0	4pyw9DVHG	0.168	2	0.13	-12.506	1	Gati Mardika	6	1921	0.07	119.824
0.406	1921	0.996	['John McCor	0.518	159507	0.203	0	5uNZnEtqOS	0	0	0.115	-10.589	1	The Wearing of the Green	4	1921	0.0615	66.221
0.0731	1921	0.993	['Sergei Rach	0.389	218773	0.088	0	02GDntOXex	0.527	1	0.363	-21.091	0	Morceaux de fantaisie, Op. 3: No. 2	2	1921	0.0456	92.867
0.721	1921	0.996	['Ignacio Cors	0.485	161520	0.13	0	05xDjWH9ut	0.151	5	0.104	-21.508	0	La Ma√ ¤anita - Remasterizado	0	20/03/1921	0.0483	64.678
0.770999999	1921	0.982	['Fortuga√©']	0.684	196560	0.257	0	08zfJvRLp7pj	0	8	0.504	-16.415	1	Il Etait Syndiqu√©	0	1921	0.399	109.378
0.826	1921	0.995	['Maurice Ch	0.463	147133	0.26	0	0BMkRpQtDo	0	9	0.258	-16.894	1	Dans La Vie Faut Pas S'en Faire	0	1921	0.0557	85.146
0.578	1921	0.994	['Ignacio Cors	0.378	155413	0.115	0	0F36WMBqR	0.906	10	0.11	-27.039	0	Por Que Me Dejaste - Remasterizad	0	20/03/1921	0.0414	70.37
0.493	1921	0.99	['Georgel']	0.315	190800	0.363	0	0H3k2CvJvH	0	5	0.292	-12.562	0	La Vip√ ºre	0	1921	0.0546	174.532
0.212	1921	0.912	['Mehmet Ker	0.415	184973	0.42	0	0LcXzABeA8	0.89	8	0.108	-10.766	0	Ud Taksimi	0	1921	0.114	70.758
0.493	1921	0.0175	['Zay Gatsby']	0.527	205072	0.691	1	0MJZ4hh60z	0.384	7	0.358	-7.298	1	Power Is Power	0	27/03/1921	0.0326	159.935
0.282	1921	0.989	['Sergei Rach	0.384	221013	0.171	0	0NFeJgmTAV	0.82	7	0.116	-20.476	0	10 Pr√ ©ludes, Op. 23: No. 5 in G Mi	4	1921	0.0319	107.698
0.218	1921	0.957	['Phil Regan']	0.259	186467	0.212	0	0Nk5f07H3Ja	0.000222	2	0.236	-13.3	1	Come Back To Erin	1	1921	0.0358	85.726
0.664	1921	0.996	['Hector Berl	0.541	250747	0.283000000	0	0POO8Xa1gV	0.898	9	0.393	-14.808	1	R√ ºk√ ¤czy March	0	1921	0.0477	108.986
0.0778	1921	0.148000000	['THE GUY']	0.604	204957	0.418	1	0QQmUf4aPI	0.0382	4	0.102	-11.566	0	When We Die	0	11/09/1921	0.0417	80.073000000
0.527	1921	0.971	['Christopher	0.54	122000	0.0848	0	0QU5xT6Mik	0.00196	5	0.0887	-16.055	0	A Ballynure Ballad	0	1921	0.075	100.296
0.672	1921	0.994	['Fortuga√©']	0.67	191333	0.113	0	0QvUUTHYUC	0	10	0.213	-16.57	0	Je Suis Toujours La	0	1921	0.196	87.162
0.24	1921	0.994	['John McCor	0.4	187333	0.155	0	0RPKAq5yhbC	4.33e-05	4	0.103	-13.976	1	Mother Machree	0	1921	0.0873	170.250999
0.422	1921	0.995	['Ignacio Cors	0.648	154240	0.0995	0	0SK1upzAP6f	0.846	11	0.112	-22.429	1	Flor Marchita - Remasterizado	0	20/03/1921	0.105	71.978000000
0.381	1921	0.995	['Hanende f∞	0.223	158908	0.245	0	0UqiUmGDIh	0.875999999	7	0.354	-14.387	1	f∞mtidadƒ¤ A¤ükƒ¤ma	0	18/12/1921	0.0387	142.136
0.41	1921	0.97	['Mehmet Ker	0.269	86988	0.143000000	0	0VhIFYGSpY	0.469	4	0.166	-9.003	1	Ney Taksimi	0	1921	0.0413	141.386
0.0731	1921	0.993	['Sergei Rach	0.389	218773	0.088	0	0eQsdik7GTE	0.527	1	0.363	-21.091	0	Morceaux de fantaisie, Op. 3: No. 2	0	1921	0.0456	92.867
0.677999999	1921	0.996	['Maurice Ch	0.5	181733	0.274	0	0i7MdVu0tNI	0	2	0.302	-14.001	1	Je M'donne	0	1921	0.041	79.218
0.723	1921	0.388	['Mehmet Ker	0.685	155063	0.698	0	0osXBirvQzP	1.97e-06	4	0.0768	-8.184	0	Korkma S√ ¶nmez	0	1921	0.0421	133.951
0.792	1921	0.99	['Fortuga√©']	0.665	203800	0.231	0	0qt9gBzfZZiC	0	2	0.287	-14.116	1	La Victoire De La Madelon	0	1921	0.155	129.671
0.425	1921	0.992	['Ignacio Cors	0.286000000	130200	0.177	0	0slGoVO1yul	0.932	7	0.11	-25.772	1	Por Ella - Remasterizado	0	20/03/1921	0.0431	167.352
0.403	1921	0.992	['Ignacio Cors	0.332	166987	0.179	0	0y3F8OUDFu	4.41e-06	0	0.432000000	-20.937	1	A Mi Palomita - Remasterizado	0	20/03/1921	0.0443	75.704
0.614	1921	0.991	['Ignacio Cors	0.484	159013	0.215	0	0yKcdafy64x	0.000546	4	0.136999999	-22.223	0	Amurado Me Dejaste - Remasteriza	0	20/03/1921	0.106	76.277
0.185	1921	0.505	['KHP Kridhar	0.233	686664	0.00817	0	0yxeAe7MlSC	0.000203	8	0.109	-37.311	1	Raja Manggala	5	1921	0.0305	108.143
0.47	1921	0.765999999	['Mehmet Ker	0.453	337607	0.558999999	0	107JTiqeqxjS	0.0104	9	0.213	-8.072	1	E¤üin Var √ e¤üiy√ e Var	1	1921	0.0329	156.077

Figure 1: This figure shows the features in the “data.csv” file.

Task 1: Deliver a recommendation engine (60% of your total mark)

Programming language required: Python

Your task is to design and develop five modules, namely load_dataset_module, statistics_module, similarity_module, the user_interface_module and the main module function that imports and uses the rest of the modules.

You will design and implement your solution using object-oriented programming (OOP) concepts. You should implement all parts of your solution using classes, methods, inheritance, etc. You will demonstrate your understanding of key object-oriented programming concepts that you have learnt throughout the entire implementation.

This part of the assignment is split in the following sub-tasks/steps to facilitate your work. For each module developed, you should check that you have implemented an OOP structure (classes and modules) and that you have applied exception handling techniques so that your code will exit gracefully.

Loading and parsing music dataset file (load_dataset_module)

In this task, you will develop a suitable module for retrieving data from the provided dataset. The data to be retrieved will be temporarily stored in memory (for additional processing) using one dictionary for artists, their tracks, and features.

The module retrieves “acoustiness”, “artists”, “danceability”, “energy”, “id”, “liveness”, “loudness”, “name”, “popularity”,” speechness”, “tempo”, and “valence” from the data.csv dataset. This module should return a dictionary (artist_music). The artist_music dictionary contains artists, track names, and corresponding features.

Statistical functions (statistics_module)

This module should implement functions for statistical analysis of the dataset features. The statistical features should include some (or all) of the following: mean, mode, standard deviation, minimum, maximum, and variance. The statistical functions included in this module must be used to compute the similarity between artists and tracks.

Computing similarity (similarity_module)

In this task, you will design and implement 4 functions in the similarity module (similarity_module), for computing similarity scores between two items based on the features retrieved from the load_dataset_module. You will also evaluate these functions by explaining and justifying which of the similarity measures you consider the best in your presentation.

These functions are: euclidean_similarity (Euclidean distance), cosine_similarity (Cosine distance), pearson_similarity (Pearson correlation), and manhattan_similarity (Manhattan distance). You may need to research the mathematical formulas for these similarity metrics. Each of these functions will accept 3 parameters, namely: the artist_music dictionary and two identifiers, which should be a pair of artist names, track names, or music IDs. The functions should be able to use features, e.g. loudness, liveness, valence, etc., to compute the similarity.

Computing similarity between music tracks and similarity between artists (similarity_module)

In this task, you will design and implement 2 further functions in the similarity_module. The first function will compute similarity scores between two music tracks using any of the 4 functions in (c). Similarly, the second function will compute the similarity scores between two artists using any of the 4 functions in (c). The functions should accept appropriate parameters (a pair of artist names, track names, or track ids), and one of the similarity functions in (c). Hint: your functions should be able to use any of the features retrieved from the dataset to compute the similarity.

Ranking artists and music (similarity_module)

In this task, you are to implement a method in the similarity_module that returns the first 5 similar artists using any of the implemented similarity metrics. It should be possible to use the same method to generate the first 5 similar music tracks.

Artist recommendations (similarity_module)

In this task, you are to implement a method in the similarity_module that generates recommendations for any artist or track based on their similarity scores.

The user interface (user_interface_module)

In this task you will create a module that produces a user interface. You should use the tkinter library in accordance with the relevant tutorial in codio. The user interface should prompt the user to select a metric to calculate similarity. It should also enable the user to enter a pair of track IDs, or artists' names, or track names (all three options of pairs should be made available to the user). Marks will be assigned for successful exception handling and display of relevant messages. The interface should display the similarity score between the two artists or between the two tracks, the top 5 similar artists for each artist in the comparison (if the user has selected to compare artists) or the top 5 similar tracks for each track in the comparison (if the user has selected to

compare tracks or track IDs).

The interface should give the user the option to quit the application at any time.

The main module (main.py)

This module should import the rest of the modules and start the application. The user should be able to launch your application by navigating to the file tree and running "python3 main.py" in the terminal.

Video Demonstration

You must record a video demonstrating your deliverables' functionality and how it meets the assessment criteria. The length of this video should be 20 minutes maximum. In your video, introduce yourself, the problem you solved, your solution (design and implementation of all the above tasks), how your solution meets the assessment criteria, and a walkthrough of the application.

You should also demonstrate the structure of your code, the implementation of the object-oriented programming principles and demonstrate and explain cases of error handling.

For the first task, you should submit a folder with the following files:

1. load_dataset_module.py
2. statistics_module.py
3. similarity_module.py
4. user_interface_module.py
5. main.py
6. recorded video presentation (preferably an .mp4 file).

Failure to submit any of the files above will result in the appropriate mark reduction.

Task 2: Deliver an Exploratory Data Analysis Report (40% of your total mark)

Programming language required: R

In this task, you should perform an **Exploratory Data Analysis (EDA)** on the given music dataset and present findings in an **RMarkdown report** using R. The focus is to understand the dataset's structure, clean and pre-process the data, explore relationships between features, and visualise key trends.

The report should comprise the following sections, and you should include the following information in each section:

Introduction

Explain the scope of this report and its main outcomes.

Data Import and setup

Load the libraries required and explain their main functionality. Explain which methods of these libraries you will use in the report and why. Load the dataset data.csv and show a sample of the set (first few rows). Describe the dimensions and the structure of the dataset.

Data Cleaning and Preparation

Select only the required features listed in the "dataset" section above. Check for missing values and handle them appropriately (e.g., remove, impute, or replace), remove any duplicates and convert data types (e.g. are the numbers in the dataset imported as strings? Should they be converted to a different type?).

Descriptive statistics

Summarise the data using the appropriate statistical methods for numerical and categorical data.

Data Visualisation

Create clear, well-labelled plots to explore the data and interpret your findings. Use Distribution plots (histograms or density plots) for selected features, Boxplots to compare distributions, e.g., popularity vs. energy, scatter plots to visualise relationships between continuous variables, and Bar charts to investigate features relevant to artists or tracks.

Relationship and Trend Analysis

Investigate correlations between the features (use the Correlation heatmaps to show relationships among numeric features). Explore how features relate to popularity and identify trends (e.g., are more "danceable" songs generally more popular?).

Insights and Interpretation

Interpret key findings from visualisations and correlations and highlight interesting patterns or outliers. Suggest possible explanations or implications (e.g., popular songs tend to have higher energy and danceability).

Knit the report into PDF format.

The word count of the report should be a maximum of 3500 words, excluding the code in R, the images, and the appendix with the Transparency declaration statement (see below).

For the second task, you should submit a folder with the following files:

1. Report file that includes the code and markdown. This should be an .Rmd file. No other file types will be accepted.
2. A .pdf file of this report.

Failure to submit any of the files above will result in the appropriate mark reduction.

Pay attention to the following requirements:

This assignment is an individual piece of work, and your submission must be in the form detailed below. We should be able to open and run your modules on a standard campus computer.

You **must submit on Blackboard a .zip folder** with the subfolders Task_1 and Task_2. An example of the file structure and naming for a student with ID: 1234567 is as follows:

Main folder	**subfolder**	**File name**
1234567_intro_programming.zip	1234567_Task1	load_dataset_module.py
		statistics_module.py
		similarity_module.py
		user_interface_module.py
		main.py
		1234567_presentation.mp4
	1234567_Task2	1234567_EDA.Rmd
		1234567_EDA.pdf

Table 1. Submitted file structure for Blackboard

You must also submit your EDA report on Turnitin. In the example above, the student should submit the report named 1234567_EDA.pdf on Turnitin.

Any evidence of collusion/plagiarism will be penalised if appropriate! If there is some doubt about the authenticity of a particular piece of work, then the person submitting it will be expected to defend such work, including reasons for the programming decisions taken.

Appropriate use of variable names for clearer understanding is desirable.

Adequate commenting of your code for easier understanding during your presentation/grading is also desirable.

Submission Process and Deadline

Your assignment should be submitted electronically through the module's Blackboard site as a single ZIP file that contains all the information described above. Check your upload to ensure you have submitted the correct files successfully, as any issues will not be considered after the deadline. **The submission deadline is Tuesday, 16th December 2025, 15:00 pm.**

You must also submit your EDA report (the .pdf file only) on Turnitin.

Check your upload to ensure you have submitted the correct files successfully, as any issues will not be considered after the deadline.

Note that any wrong submission or submission of the wrong file or deliverable will be graded as 0. Please do not send your deliverable file via email.

Please ensure that your correct files are submitted, as we will not entertain any request for mistaken submission after the deadline.

Note that late submission will attract a penalty. The penalty is a cap on your mark to 50% or 0, depending on how late the submission is.

Who do I contact if I have a question?

Please use the form shared (Questions about the assignment in Introduction to Programming – Fill in form) to write down any questions you have. All questions submitted via this form will be addressed promptly in Blackboard.

Artificial Intelligence and Academic Integrity – AI&AI

It is important that if you use AI tools to generate an assignment, you do not submit it as if it were your own work. You must check the AI Transparency Scale (AITS) to determine what level of AI use is allowed.

Our regulations state:

Contract cheating/concerns over authorship: This form of misconduct involves another person (or artificial intelligence) creating the assignment, which you then submit as your own. Examples of this sort of misconduct include: buying an assignment from an 'essay mill'/professional writer; submitting an assignment which you have downloaded from a file-sharing site; acquiring an essay from another student or family member and submitting it as your own; attempting to pass off work created by artificial intelligence as your own. These activities show a clear intention to deceive the marker and are treated as misconduct.

Further guidance is available here: https://blogs.shu.ac.uk/assessment4students/preparing-to-submit-work/#AI

AI Transparency Scale (AITS)

For this assessment, the permitted use of Artificial Intelligence is highlighted in the table below. All students are required to include a transparency declaration statement which can be added as an appendix to the assessment (not included in the word count)

SHU AI Transparency Scale (AITS)				
AITS	**Descriptor**	**Transparency Statement**	**AI Contributions**	**Human Contribution**
1	No AI	Artificial Intelligence (AI) has not been used for any part of the activity.	AI is not used for any part of the activity.	All aspects of the activity are human generated, created, edited, and developed.

Transparency declaration statements

Your statement should:

- Indicate the AITS descriptor you are using.
- Confirm no AI was used.

Please include this statement as an appendix in your EDA report. Failure to comply with this requirement may be considered a breach of academic integrity under our Academic Conduct Policy.

Assessment Criteria

Criteria	Fail (<50%)	Pass (50–59%)	Merit (60–69%)	Distinction (70% +)
1. Understanding and Implementation of Programming Concepts (OOP) (/10)	No or minimal evidence of understanding and application of Object-Oriented Programming (OOP) concepts (classes, objects, methods, inheritance, etc.). Poor or no modular design. No or poor exception handling. Code may not run.	Clear but basic understanding and use of OOP principles in some modules. Classes and methods are implemented, but may lack consistency or full integration. Limited or partial exception handling.	Strong and consistent implementation of OOP principles across all modules. Demonstrates good modularity, reusability, and correct use of inheritance and encapsulation. Exception handling is present and mostly effective.	Excellent and creative use of OOP principles across all modules. Highly modular, robust, and reusable structure. Effective and elegant exception handling. Shows deep understanding of software design and architecture.
2. Implementation of Modules and Functionality (/15)	One or more required modules are missing or incomplete. Functions do not perform as described. No integration between modules. The application does not run successfully.	All required modules were implemented with some errors or missing functionality. Basic integration achieved, but results are inconsistent or incomplete.	All required modules were implemented with good integration and functionality. Minor issues may exist, but overall logic and data flow are correct. Outputs are mostly accurate.	All modules are fully implemented, well integrated, and function seamlessly. Excellent modular interaction and clear data flow. The program executes perfectly with robust outputs.
3. Use of Python Libraries (/10)	No or incorrect use of relevant Python libraries (e.g., numpy, pandas, matplotlib, tkinter, scikit-learn). Poor or no data manipulation or similarity computation.	Some evidence of using Python libraries appropriately (e.g., loading data, basic similarity calculation). Some outputs are correct but may have inconsistencies.	Good and consistent use of relevant Python libraries. Functions implemented for similarity measures and statistical analysis produce correct outputs. Some justifications were provided.	Excellent use and integration of multiple Python libraries. Implements and justifies all similarity functions effectively (Euclidean, Cosine, Pearson, Manhattan). Highly accurate and efficient implementation with clear justifications.
4. User Interface Design and Interaction (/10)	No interface or non-functional interface. Poor or no user input handling. No feedback or error messages.	Functional but limited user interface using tkinter. Allows some user interaction but lacks	Well-structured, user-friendly interface allowing user input, selection of metrics, and displaying similarity results.	Professional and intuitive GUI design. Smooth user interaction and clear presentation of results. Excellent error handling and user feedback. Demonstrates

Criteria	Fail (<50%)	Pass (50–59%)	Merit (60–69%)	Distinction (70% +)
		robustness. Error handling is minimal or ineffective.	Effective error handling and messaging.	innovation (e.g., dynamic menus, interactive results).
5. Statistical Computation and Evaluation of Similarity (/10)	No statistical functions implemented or incorrect calculations. No similarity evaluation or interpretation.	Some statistical functions are implemented, but with errors or incomplete evaluation. Limited explanation of similarity measures.	Most statistical functions are correctly implemented (mean, std, variance, etc.). Good evaluation and justification of chosen similarity measures.	All statistical functions are correctly implemented and well-integrated. Excellent justification and evaluation of similarity methods. Demonstrates deep understanding of data and metrics.
6. Code Quality, Documentation, and Video Demonstration (/5)	Poorly structured or commented code. No video demonstration or report. Weak or no explanation of implementation.	Code is structured but inconsistent in commenting or naming conventions. Adequate video demonstration showing basic functionality.	Well-organised and documented code with consistent naming and clear commenting. Very good video demonstration covering key functionalities.	Exceptionally well-documented, clean, and readable code. Excellent video demonstration with clear explanations of design, OOP principles, and results. Shows reflection, innovation, and professional presentation.

Table 2. Assessment Criteria for Task 1 (60% of the total grade)

Criteria	Fail (<50%)	Pass (50–59%)	Merit (60–69%)	Distinction (70% +)
1. Data Import, Setup, and Cleaning (/10)	No or poor loading of the dataset. Missing values or duplicates not handled. Wrong data types or no preprocessing.	Dataset imported and basic cleaning performed. Some issues with handling missing or incorrect data types.	Data imported, cleaned, and pre-processed appropriately. Few issues remain. Data types were correctly converted.	Excellent and complete cleaning and preprocessing. Handles missing data and duplicates professionally. Data ready for analysis with clear justifications.
2. Descriptive Statistics and Summaries (/10)	No or incorrect summary statistics. No explanation or poor interpretation.	Basic descriptive statistics presented (mean, min, max, etc.) with limited interpretation.	Comprehensive descriptive statistics were provided and correctly interpreted.	Excellent statistical summary with insightful interpretation of dataset features.
3. Data Visualisation (/10)	No or poor visualisations. Incorrect or unreadable plots. No interpretation.	Some correct plots provided (histograms, scatter plots, etc.), but limited interpretation or poor labelling.	Good range of clear and correctly labelled plots. Appropriate interpretation and visual storytelling.	Excellent, professional-quality plots. Clear, well-labelled, and insightful visualisations. Demonstrates strong analytical storytelling.
4. Relationship and Trend Analysis (/5)	No or incorrect analysis of relationships. No correlation matrix or trend explanation.	Some correlation analysis was done, but poorly interpreted.	Good correlation and trend analysis with correct interpretation.	Excellent correlation and trend analysis with strong insights and evidence-based interpretation.
5. Insights and Interpretation (/5)	Lacks meaningful insights. No connection between analysis and interpretation.	Provides basic interpretations but lacks depth or linkage to findings.	Good interpretation of results with relevant insights.	Exceptional interpretation and synthesis of findings. Insightful conclusions supported by strong evidence and visualisations.

Table 3. Assessment Criteria for Task 2 (40% of the total grade)

University Grade Descriptor (UGD)

Class	Category	Grade	Mark range	%	General Characteristics
Distinction	Exceptional Distinction	16	93 - 100	96	Exceptional breadth and depth of knowledge and understanding evidenced by own independent insight and critical awareness of relevant literature and concepts at the forefront of the discipline; evidence of extensive and appropriate independent inquiry operating with advanced concepts, methods and techniques to solve problems in unfamiliar contexts; Cogent arguments and explanations are consistently provided using a range of media demonstrating an ability to communicate effectively in a variety of formats using a sophisticated level of the English language in an eloquent and professional manner to both technical and non-technical audiences; a sustained academic approach to all aspects of the tasks is evidenced; academic work extends boundaries of the disciplines and is beyond expectation of the level and may achieve **publishable or commercial standard**.
Distinction	High Distinction	15	85 - 92	89	Excellent knowledge and understanding evidenced by some **clear independent insight and critical awareness of relevant concepts some of which are at the forefront of the discipline**; evidence of appropriate independent inquiry operating with core concepts, methods and techniques to solve complex problems in mostly familiar contexts; Arguments and explanations are provided that is well-supported by the literature and in some cases uses a range of media demonstrating an ability to communicate effectively in a limited number of formats using own style that is suited to both technical and non-technical audiences; a sustained academic approach to most aspects of the tasks is evidenced; **one or more aspects of the academic work is beyond the prescribed range** and evidences a competent understanding of all of the relevant taught content.
	Mid Distinction	14	78 - 84	81	
	Low Distinction	13	70 - 77	74	
Merit	High Merit	12	67 - 69	68	Very good knowledge and understanding is evidenced as the student is **typically able to independently relate taught facts/concepts together some of which are at the forefront of the discipline**; evidence of some competent independent inquiry operating with core concepts, methods and techniques to solve familiar problems; Arguments and explanations are provided that are typically supported by the literature and in some cases may challenge some received wisdoms; competently uses all taught media and communication methods to communicate effectively in a familiar settings; an academically rigorous approach applied to some aspects of the tasks is evidenced; some beyond the prescribed range, may rely on set sources to advance work/direct arguments; demonstrates autonomy in approach to learning.
	Mid Merit	11	64 - 66	65	
	Low Merit	10	60 - 63	62	
Pass	High Pass	9	57 - 59	58	Satisfactory knowledge and understanding of the area of study **balanced towards the descriptive rather than critical or analytical and mostly confined to concepts that are not at the forefront of the discipline**; evidence of some independent reading and research to advance work and inform arguments and approaches; Arguments and explanations are limited in range and depth although some are adequately supported by the literature albeit descriptively rather than critically; competently uses at least one taught media and communication method to communicate appropriately in familiar settings; although the approach applied to some aspects of the tasks may lack academic rigour, there are some clear areas of competence within the prescribed range. Relies on set sources to advance work/direct arguments and communicated in a way which shows clarity but structure may not always be coherent.
	Mid Pass	8	54 - 56	55	
	Low Pass	7	50 - 53	50	
Fail	Borderline Fail	6	40 - 49	45	Knowledge and understanding is insufficient as the **student only evidences an understanding of small subset of the taught concepts and techniques; fails to make sufficient links between known concepts and facts** to adequately solve relevant aspects of the brief/problem; little ability to independently select and evaluate reading/research with almost total reliance on set sources and unsubstantiated arguments/methods; communication/presentation may be competent in places but fails to demonstrate
	Mid Fail	4	30 - 39	35	

Class	Category	Grade	Mark range	%	General Characteristics
	Low Fail	2	20 - 29	25	structure, clarity and/or focus; inability to adequately define problems and make reasoned judgements; the general approach to tasks lacks rigor and competence.
Fail	Very Low Fail	1	6-19	10	Knowledge and understanding is highly insufficient as the student is **unable to evidence any meaningful understanding of taught concepts or methods**; very limited evidence of reading and research to advance work; inadequate technical and practical skills as the student is unable to use and apply such skills to address problems or make judgements; limited or lack of understanding of the boundaries of the discipline and does not question received wisdom; approach to learning lacks autonomy and approach to tasks is not sustained; inability to communicate coherently.
Zero	Zero	0	0-5	0	Work not submitted, work of no merit, penalty in some misconduct cases.

Table 4: Level 7 - Generic Grade Descriptor