Department of Computer Science
Summative Coursework Set Front Page

Module Title	Applied Data Science with Python
Module Code	CSMAD21
Lecturer responsible	Dr Carmen Lam, Dr Todd Jones
Type of Assignment (e.g., technical report, portfolio exercise, in-class test)	Coursework
Individual or Group Assignment	Individual
Weighting of the Assignment	100%
Word count/page limit	Approximately 3,000 words, excluding captions and tables
Expected hrs spent for the assignment (set by lecturer)	40
Items to be submitted	A single **`.zip`** or **`.tar.gz`** archive, **containing**: 1. One Jupyter Notebook file (.ipynb) containing code, figures, and explanations (as Markdown) 2. One HTML file (.html) exported from the above Jupyter notebook file (File --> Download as --> HTML)
Work to be submitted on-line via Blackboard Learn by	**12:00 noon, Friday, 8th December 2023**
Work will be marked and returned by	**Tuesday, 9th January 2024**

Note

By submitting this work, you are certifying that you have read the assessment guidelines, which are displayed at the top of the Assessment Folder on the Blackboard course for this module, and that you have conformed to the associated policies and practises, including those on:

- Submitting your own work, not that of other people or systems, and the associated penalties for Academic Misconduct
- Submitting by the specified deadline, and the penalties associated with late submission (if allowed)
- The exceptional circumstances system (for applying for extensions)
- The use of a green sticker for students with relevant needs

ASSESSMENT CLASSIFICATIONS

This coursework assesses your ability to:

- understand and use appropriate Python syntax and ecosystem;
- understand statistical and machine learning methods for data analytics and mining in Python;
- apply appropriate statistical and machine learning techniques for data science tasks.

You will gain credit for:

- preparing and submitting required files as requested;
- successful implementation of the requested coding tasks;
- writing efficient, functional code;
- providing thoughtful, clear, well-structured written analysis.

Your assignment will be marked according to the marking schemes provided below. The schemes are designed so that the collectively weighted assignment mark will correspond to the following qualitative Master's degree classification descriptions:

The table below shows what is typically expected of the work to obtain a given mark.

Classification Range	**Typically, the work should meet these requirements**
Distinction (>=70%)	Outstanding/excellent work with correct codes and results. An outstanding work should demonstrate coding proficiency with high efficiency and based on advanced techniques.
Merit (60-69%)	Good work with mostly correct results: most work has been carried out correctly. Some tasks have not been carried out or are not completely correct. Coding with average efficiency.
Pass (50-59%)	Achievement of the minimum requirements. Some significant part of the assignment is missing and/or has partially correct results. Coding lacks efficiency.
Fail (<50%)	Incomplete solutions to limited part of the assignment. Most tasks have not been carried out with sufficient accuracy. Results may not be correct or technically sound. Coding is inefficient.

ASSIGNMENT DESCRIPTION

Major Coursework (100% of module assessment)

This assignment consists of **two tasks**. Each of these will be used to assess your implementation of elements of the Data Science Process, using Python as the main tool.

A detailed breakdown of the Marking Scheme is provided later in this document.

Data Description

The data are available on Blackboard, under the **Assessment** heading. These files contain comma-separated values (CSV), some with a header that briefly describes each column. More detail about these is provided below. You **MUST** use these versions of the datasets.

Bicycle Journey Data: `metro.csv`

This dataset contains anonymised bicycle trip data from the Los Angeles Metro Bike Share (Source: https://bikeshare.metro.net/about/data/).

Each **row** corresponds to a single bicycle journey.

The **columns** correspond to:

trip_id	Locally unique integer that identifies the trip.
duration	Length of trip in minutes.
start_time	The date/time when the trip began, presented in ISO 8601 format in local time.
end_time	The date/time when the trip ended, presented in ISO 8601 format in local time.
start_station	The station ID where the trip originated (for station name and more information on each station see the Station Table).
start_lat	The latitude of the station where the trip originated.
start_lon	The longitude of the station where the trip originated.
end_station	The station ID where the trip terminated (for station name and more information on each station see the Station Table).
end_lat	The latitude of the station where the trip terminated.
end_lon	The longitude of the station where the trip terminated.
bike_id	Locally unique integer that identifies the bike.
plan_duration	The number of days that the plan the passholder is using entitles them to ride; 0 is used for a single ride plan (Walk-up).
trip_route_category	"Round Trip" for trips starting and ending at the same station or "One Way" for all other trips.
passholder_type	The name of the passholder's plan.
bike_type	The kind of bike used on the trip, including standard pedal-powered bikes, electric assist bikes, or smart bikes.

Seed Shape Data: `seeds.csv`

This dataset contains measurements of seeds from several different plant species.

Each **row** corresponds to a single seed's measurement.

The **columns** correspond to:

`area`	**A**, the area of the seed.
`perimiter`	**P**, the length of the perimeter of the seed.
`compactness`	A measure of the area of the seed relative to the perimeter, $(4\pi A/P^2)$
`length`	The length of the seed.
`width`	The width of the seed.
`asymmetry`	A measure of the asymmetry of the seed.
`groove_length`	The length of the groove in the seed.

Data are modified from their source from UCI machine learning repository:

https://archive.ics.uci.edu/ml/datasets/seeds

Originally by: M. Charytanowicz, J. Niewczas, P. Kulczycki, P.A. Kowalski, S. Lukasik, S. Zak in 'Complete Gradient Clustering Algorithm for Features Analysis of X-ray Images', in: Information Technologies in Biomedicine, Ewa Pietka, Jacek Kawa (eds.), Springer-Verlag, Berlin-Heidelberg, 2010, pp. 15-24.

Task 1 – Bicycle Journeys: Exploratory Data Analysis

Using the `metro.csv` file, demonstrate a thorough implementation of exploratory data analysis and the data science process. This might include:

- Exploring the data format(s).
- Data pre-processing.
- Data cleansing.
- Identification, explanation, and handling of missing data, outliers, and other irregularities.
- Reduction and/or transformation (e.g., on the time fields).
- Explanatory data visualisations.
- Summary and more granular statistics.
- Interpretation, insights, and knowledge extraction.

Provide a **written explanation** of your chosen manipulation and analysis methods, including the justifications behind them, lessons learned from the data and its analysis, and other notable findings. This should be provided in a set of well-formatted Markdown cells at appropriate points in your Jupyter Notebook.

You will certainly want to display and discuss various descriptive statistics to characterise the data (e.g., distributions, means, or totals of various fields or conditioned fields). You might **explore** how the `duration` variable changes between each journey's starting hour and day of the week. You might perform **statistical tests** to determine whether mean `duration` differs between `passholder_type`.

There is significant room for freedom and creativity in this task. At the same time, it is essential to demonstrate proper usage of the techniques you have learned in the module for effective information discovery.

Task 2 – Seed Shape Analysis

Begin with the `seeds.csv` file, which contains various physical measurements of many seeds, as described above. Complete the following activities:

- **Implement two** of the methods described in the module lectures and practical sessions to model any natural groupings that might exist within the dataset. These are the **primary models** for this task. A more thorough analysis will perform more than two such tests.
  - Provide a **written explanation** of your chosen analysis methods, including their properties and justifications for choices you make in their implementation.
- **Evaluate** the quality of your chosen analysis methods. Do different analysis choices significantly affect the results and conclusions?
  - **Visualise** the model results and evaluation concisely, **discussing** the reasons why one might prefer the use of one of your tested methods over another.
- **Locate a copy** of the source dataset.
  - Determine and report how the source dataset differs from the file provided for this task and how this can be important.
  - Consider a potential plan for additional analysis based on what you have found in the source dataset. For instance,
    - How can your evaluation of the above models be enhanced?
    - Which new type of data analysis can now be performed?
- **Implement** an example of re-evaluating your previous results based on the newly discovered information.
  - Provide a **written explanation** of the technique and its utility.

- **Implement one** of the methods described in the module lectures and practical sessions that aligns with this new analysis. This is the **secondary model** for this task. A more thorough analysis will perform more than one such test.
  - Provide a **written explanation** of your chosen analysis method, including its properties and justifications for choices you make in their implementation.
- **Evaluate** the quality of your new chosen analysis method. Do different analysis choices significantly affect the results and conclusions?
  - **Visualise** the model results and evaluation concisely, **discussing** the reasons why one might prefer the use of one of your tested methods over another.

Written responses should be provided in a set of well-formatted Markdown cells at appropriate points in your Jupyter Notebook.

Assignment Submission Requirements

"Front page" of the Submission

The following are **compulsory**. Please add these items to at the **start of your Jupyter Notebook** in a Markdown cell. To be extra helpful, please repeat this information in the **Add Comments** section of the submission page.

Module Code:

Assignment Report Title:

Date (when work was completed):

Actual hours spent on assignment:

We will use information about how long you spent on the assignment when we review and balance coursework between modules for later years. An exact answer is not necessary, but please try to give a reasonable approximation.

Assignment Content

You must create a Python (**version 3.10** or above) Jupyter Notebook (**version 6.3.0** or above). Where possible, use the packages included in the associated Anaconda3 distribution used in this module (**2023.03**).

If you find good reason to employ **additional Python packages**, please provide an excruciatingly detailed description of the package installation procedure that includes specification of your Anaconda3, Python, and Jupyter Notebook versions, as well as the version information for your additional Python packages.

As outlined above, your submission should take the form of a single archive file containing **one Jupyter Notebook** file with responses to each of the two tasks and **one HTML version** of the Jupyter Notebook.

You will find the submission point on the module's Blackboard page under **Assessment**. The name of the archive should be formatted with your 8-digit student ID number, the module code, and the tag "Coursework" (e.g., **`12345678_CSMAD21_Coursework.tar.gz`**).

Within the single archive, please name each file with your student ID (e.g., **`12345678_CSMAD21_Coursework.ipynb`**). Inclusion of your own modules are permitted with appropriate documentation.

Code Plagiarism

This coursework is expected to be the result of your own individual effort. Do not work closely with others on this coursework. Do not employ pair programming techniques. Copying whole tutorials, scripts or images from external sources is not permitted. Any material you borrow from other sources **to build upon or to support your arguments** should be clearly referenced (use comments to reference element within Python scripts and code cells and supply formal references in a Markdown section at the end); otherwise, it will be treated as plagiarism, which may lead to investigation and subsequent action. Work **inspired by** module materials is permitted, but such material should not be used without significant modification. We understand that similar lines of code are inevitable, however, very similar lines of analysis and reporting spanning significant sections of the coursework will be investigated as potential academic misconduct (e.g., it is highly unlikely that you will independently choose the same model parameters and variable names).

Marking Scheme

Task	Marks Available
Task 1 – Bicycle Journeys: Exploratory Data Analysis Implements EDA, data cleansing, and data visualisation techniques in Python as requested in the task description. • [10 marks] Basic EDA • [10 marks] Data cleansing • [20 marks] Informational insights in the form of explanatory visualisations and associated written interpretations • [10 marks] Quality of visualisations (e.g., correct chart types, annotation, etc.), code and report quality	**50**
Task 2 – Seed Shape Analysis Demonstrates understanding and appropriate application of statistical and machine learning approaches in Python by implementing the necessary code and providing critical written analysis of the results, describing all figures, and explaining the chosen approach. • [14 marks]* Primary model implementations • [14 marks]* Primary model evaluations • [2 marks] Data discovery and reporting • [4 marks]* Re-evaluation of primary models • [8 marks]* Secondary model implementation • [8 marks]* Secondary model evaluation For each starred (*) element above, half of the marks correspond to the quality of the associated visualisations and written responses, considering the specific elements requested in the task description and the collective points below.	**50**
Total	**100**

In all aspects of the above, consider these points:

- Application of best coding practices is expected. This includes clear documentation (comments and docstrings), logical importation of libraries, implementation of your own functions, classes, and modules, exception handling, clear, concise, and efficient data processing code, and appropriate use of DataFrames, lists, dictionaries, or any other data structures and flow control elements defined during the analysis.
- Appropriate visualisation and statistical methods should be chosen to answer questions about the data. Visualisations should be comprehensively annotated.
- Clear report structure and reasoning are expected through the notebook (e.g., organisation into sections, text formatting, completeness, clarity, and logical explanation).
- Consider the ways in which Jupyter Notebooks might display data. Retain and describe elements which are noteworthy, informative, and clear, and avoid including unhelpful displays (e.g., long lists of numbers).

- Clear written communication, conforming to accepted standards of academic writing is expected (e.g., complete explanatory sentences that provide explanation and describe the values and figures produced). This should be included in Markdown cells adjacent to the content being discussed.
- Citations, which are a preferred method of supporting your arguments with ideas based outside the module content, should be provided in a format conforming to a single formal citation method of your choice (e.g., Harvard, APA, AMS, ISO 690, etc.).