Department of Computer Science

Summative Coursework Set Front Page

Module Title	Programming in Python for Data Science
Module Code	CS2PP22
Lecturer responsible	Dr Todd Jones
Type of Assignment (e.g., technical report, set exercise, in-class test)	Report and Presentation
Individual or Group Assignment	Group
Weighting of the Assignment	60%
Word count/page limit	Maximum 3,000 words as body text in Markdown cells (counting script provided)
Expected hrs spent for the assignment (set by lecturer)	12 hours per group member
Items to be submitted	3 separate files: 1. HTML of fully executed Notebook report 2. .zip archive of all project code 3. Video presentation file
Work to be submitted on-line via Blackboard Learn by	**2024 March 12th (Tuesday) 12:00 (noon)**
Work will be marked and returned by	**2024 April 5th (Friday)**

Note

By submitting this work, you are certifying that you have read the assessment guidelines, which are displayed in the folder of Assessment on the Blackboard course for this module, and that you have conformed to and understand the associated policies and practices, 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 students with relevant needs, attaching with a green sticker

1. Assessment Classifications

This coursework assesses your ability to:

- demonstrate an understanding of the use of functional and object-oriented programming paradigms in Python;
- read and manipulate data in several formats to extract specific features;
- assemble, implement, and select appropriate data science methodologies in Python;
- employ third-party Python libraries appropriately to design and create well-structured programs for practical applications.

In general, you will gain credit for:

- preparing and submitting required files as requested;
- successful implementation of the specified tasks;
- writing efficient, functional code;
- providing thoughtful, clear, well-structured written analysis that conveys complex information understandably.

Your assignment will be marked according to the marking scheme provided below. The scheme is designed so that the collectively weighted assignment mark will correspond to the following qualitative degree classification descriptions. The table below shows what is typically expected of specific aspects of the work to obtain a given mark.

Classification Range	Typically, the work should meet these specifications:
First Class (>= 70%)	This work demonstrates coding proficiency with high efficiency and based on advanced techniques. Evidence of independent research into the methods used and a thorough justification of applications of these methods is presented clearly. Work at this level demonstrates exceptional understanding, creativity, and application in all aspects of the project. Data collection and analysis are comprehensive and sophisticated, with expert use of Python tools. Visualisations and data manipulations are insightful, innovative, and technically advanced. The written report and video presentation are outstanding, with excellent structure, clarity, and depth, offering insightful interpretations and discussions. Individual contributions are impressive and integral to the project, and the group collaboration is exemplary, showing a high degree of coordination, cooperation, and collective proficiency. The code is exemplary in quality, showcasing exceptional efficiency, organisation, and innovative techniques; documentation is thorough, enhancing clarity and understanding.
Upper Second (60-69%)	Good work with few mistakes. Some minor tasks have not been carried out or are not completely correct. Work at this level exhibits strong comprehension and application of the project's objectives. Data collection and analysis are thorough, demonstrating proficient use of Python tools. Visualisations and data manipulations are insightful and well-executed. The written report and video presentation are clear, detailed, and well-organised, displaying a high level of understanding. Individual contributions are substantial, and group collaboration is highly effective, demonstrating a coordinated effort. The code demonstrates strong proficiency, with efficient and well-organised structure; it is well-documented, facilitating easy readability and comprehension.

Lower Second (50-59%)	Demonstrates knowledge of core concepts but with some mistakes. Work at this level demonstrates a good grasp of the project requirements. Data collection and analysis are well-handled, with clear understanding and application of Python tools. Visualisations and data manipulations are effective and demonstrate good technical skill. The written report and video presentation are clear and well-structured, covering most aspects of the project effectively. Individual contributions are significant, and the group collaboration is evident and mostly successful. The code is well-structured and functional, with a good level of efficiency and readability; documentation is clear and aids understanding.
Third (40-49%)	Some parts of the assignment are missing and/or have partially correct results. Work at this level shows an adequate understanding of the project's scope with satisfactory data collection and analysis. Technical aspects, including Python tool usage, are competently executed but lack sophistication. Visualisations and data manipulations demonstrate a standard level of proficiency. The written report and video presentation are clear but lack comprehensive coverage and depth. Individual contributions are notable, and there is evidence of teamwork, though not fully effective. The code is functional but simplistic, with some issues in efficiency and structure; basic documentation is present but lacks detail.
Pass (35-39%)	Work at this level meets a subset of the basic requirements but lacks depth in these. There is a rudimentary understanding of the project's objectives, with elementary data collection and analysis. Technical execution is basic, with minimal use of Python tools. Visualisations and data manipulations are simplistic. The written report and video presentation cover most essential elements but lack detail and clarity. Individual contributions fulfil some basic criteria, and group collaboration is limited but present. The code meets basic functional requirements but is rudimentary, often inefficient, and lacks proper organisation or documentation.

Fail (0-34%)	Work at this level fails to meet the minimum requirements. There is a lack of understanding and application of the key concepts and methodologies related to the project. Data collection and analysis are either incorrect, superficial, or missing. Technical implementation is inadequate, with significant errors or misunderstandings. Written and video communication lack clarity and coherence, and individual contributions are minimal or ineffective. Group collaboration is poorly demonstrated or not evident. The code is either non-functional, severely flawed, or largely absent, showing a fundamental misunderstanding of basic programming concepts.

2. Assignment Description

This is a **group assessment with individually assessed components**. Each element will be used to assess your implementation of several aspects of Python via the Data Science Process for a problem of your own devising.

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

Group Formation

Groups will have either **4 or 5 members**.

Groups have been available for self-selection via Blackboard since Week 2. Those expressing an interest in being assigned to a group have already been assigned to a group.

If you have not selected a group by **Friday, 9 February at noon**, you will be assigned to a group and notified of your assignment.

If your group has fewer than 4 members on **Friday, 9 February at noon**, additional members will be assigned to your group.

Embracing Collaborative Learning and Practical Application

Computer science is a dynamic field where the ability to collaborate effectively and apply theoretical knowledge to real-world problems is invaluable. This data analysis group coursework is designed not only to assess your **technical proficiency in Python programming** but also to enhance your **collaborative skills**, which are highly valued in the professional world. Research indicates that employers and graduates alike place significant importance on teamwork and the ability to work effectively in diverse groups (Hughes & Jones, 2011; Salas et al., 2005). Through this project, you will demonstrate your understanding of both functional and object-oriented programming paradigms in Python, showcasing your ability to read and manipulate data sourced from one of a number of potential formats. By engaging in this group task, you will experience firsthand the challenges and rewards of collaborative problem-solving, a critical skill in today's interconnected and constantly changing work environments.

Project Scope and Real-World Relevance

The core of this project involves selecting a public dataset and employing third-party Python libraries—ones not covered in our module material—to extract meaningful insights. This approach enables you to explore and implement appropriate data science methodologies in a practical context. The choice to work with new, external Python libraries is deliberate, fostering your ability to adapt to new tools and think critically about their application in designing well-structured programs. This project mirrors the tasks and decision-making processes you will encounter in your professional life, ensuring that your learning experience is not only academically rigorous but also practically relevant. As part of a team, you will navigate the complexities of group dynamics, learning to negotiate, communicate, and contribute effectively—skills that are crucial in any professional setting.

Understanding Assessment Criteria and Support

To ensure a fair and transparent assessment, we will consider the final product, individual efforts, and the process of working as a group. Each team member's contribution will be evidenced and evaluated, acknowledging individual efforts within the collective endeavour. As group work can present unique challenges, additional support and arbitration will be provided upon request. Throughout the project, we encourage open communication about issues you encounter. Reflective elements are designed to reinforce the values underpinning our group assessment process, ensuring that you gain not only technical skills but also a deeper understanding of the nuances of teamwork and collaboration in a professional context. Post-assessment, written feedback will be provided via Blackboard to help you understand how marks were allocated.

Key Requirements

Your group will implement all stages of the **Data Science Process** as described in the module, but in a limited manner, for a problem of your own devising.

The following 5 elements are essential to successful completion of the coursework:

1. Dataset Selection

   - Choose a readily available, publicly accessible dataset for analysis.
   - The dataset MUST be documented at its source. That is, the provider must present sufficient descriptions of data features and methods of acquisition so that the data are quickly understandable.
   - Data may be acquired via existing files or selective extraction (e.g., APIs).
   - Take care to assure that the data characteristics are amenable to use with your selected data analysis technique (e.g., sufficient quality, appropriate number of records, interpretable format, etc.). These will be evidenced by comprehensive **Exploratory Data Analysis**.
   - Avoid any data that requires financial payment in exchange for acquisition. Free registrations are acceptable, if you are comfortable with this.
   - You might choose data that is commonly used to demonstrate data analysis techniques.

- Give due consideration to potential social, legal, or ethical concerns about the use of the dataset.

2. Incorporation of a New Python Package

    - The project must include the use of a **Python package that has not been used in the module's lectures or practical sessions**.
    - Consider packages that contribute to the project aims (e.g., data acquisition, statistical analyses, data visualisation, model evaluation, result presentation) or new/old areas of interest/experience (e.g., GUIs, website development, parallel or multi-threaded processing, image manipulation)
    - The package **must** be installable via `pip` or `conda`, not directly from source code.
    - You must provide written documentation of your installation method (command and version) into your Anaconda environment within your submitted Notebook.
    - You can also use libraries already installed with your distribution of Anaconda that we have not closely explored.
    - Often packages have myriad applications; you may note this in your descriptions, but you are only required to implement and explain **one element** of the package/library. **Try to keep it simple!**

3. Implementation of an In-Depth Data Analysis Technique

    - The project will apply one of the data analysis techniques covered in the module (e.g., regression, clustering, classification, or graph/network analysis) befitting your acquired and **pre-processed** data.
    - **Complete analysis** will consist of framework development, implementation, and evaluation, with appropriate supportive **visualisations** in each component.
    - Some performance metrics should be included, as applicable to the selected method, but these will **not dictate your assessment performance** (unless they are calculated incorrectly).

4. Collaborative Coding with Individual Contributions

    - The project must consist of **collaborative coding efforts, accompanied by implementation of code produced individually.**
    - Each member is responsible for the creation of their own Python **module file** containing **one function or class** that will be **imported** into the main project notebook and used to aid at **any point** in the larger analysis to **build upon** the core Data Science Process elements.

- These functions or classes must include **formally structured docstrings** containing your **name** and **student ID number**, and their names must include your initials as a suffix (e.g., `myPythonClass_TRJ`).
- Here, the focus is mainly on the technical implementation of a function or class from a separate module, not on grand complexity of the code.
- As examples, your code might demonstrate additional functionality from the new Python package, perform an additional EDA procedure or test an alternative implementation of a machine learning model.
- Discuss your contribution plans together to ensure that individual efforts are roughly equivalent in intended scope.

5. Individual and Group Reflective Accounts

- **As a group**, provide a Markdown cell at the end of your Notebook containing an account of how you worked as a group. In what manner and with what frequency did you meet? What communication strategies did you employ? Who assumed which roles? How were tasks planned? What was the intended scope of the individual code contributions?

    i. Were contributions of effort distributed unequally? If so, express the contributions of effort as a percentage (i.e., 100% is full effort, assumed by default). **See note following Marking Scheme table.**

    ii. If a group member did not contribute their own module code or their contribution performs improperly, please note how this affected the implementation of the Data Science Process.

- **Individually**, reflect upon your development of your individual module file code. Describe the process of designing your code and the way in which it is instrumental to the functionality of the collective project. What issues did you face in its development, and how might you approach this differently if you were to revise this element? Each group member should do this in a separate Markdown cell of their own, annotated with their name and student ID number.

3. Assignment submission requirements

"Front page" of the Submission

The following are **compulsory**. Please add these items to at the **top of your Jupyter notebook** in a Markdown cell.

Module Code:

Assignment report Title:

Date (when the work completed):

Actual hrs spent for the assignment:

Format of the Required Work

You must use Python (**version 3.10** or above) Jupyter Notebooks (**version 6.3.0** or above). Where possible, use the packages consistent with the Anaconda3 distribution used in this module (**2023.03**).

Three files are expected in your Blackboard submission:

1. **HTML** copy of your group's fully and sequentially executed notebook-embedded report and analysis
2. **.zip** archive, containing all project module (one per group member) and notebook files
3. **Video** presentation file, any common format.

You will find the submission point on the module's Blackboard page under **Assessment**.

The names of the 3 submitted files should be formatted with your group number, the module code, and the tag "CW2". For example:

```
Group08_CS2PP22_CW2.html
Group08_CS2PP22_CW2.zip
Group08_CS2PP22_CW2.mov
```

Content of the Required Work

For a problem of your own devising, you must demonstrate **all** elements of the Data Science Process:

1. Frame the Problem
2. Collect Data
3. Exploratory Data Analysis
4. Data Pre-Processing
5. In-Depth Analysis
6. Communicate Results

The Notebook

Each of these **6 components** will be presented in the above order through execution of a **single Jupyter Notebook**, incorporating the **5 key requirements** from the above **Assignment Description**.

Each of the 6 components will contain Python code cells written to complete the related tasks and Markdown cells containing academic-style writing describing the intent of the code and discussing the results of its execution. In particular, **your notebook contents will specifically include**:

- Sequentially executed code cells
- Dataset source references (e.g., link, paper reference, etc.) and a statement regarding the extent of any social, legal, or ethical concerns about your dataset
- Installation notes for your new Python package
- Precise descriptions of important variables/features
- Noteworthy discoveries from EDA
- Justifications for pre-processing methods
- Justifications for in-depth analysis framework and parameter selection
- Fully annotated visualisations
- Concluding, summative remarks

You will additionally include the **Group and Individual Reflective Accounts** in Markdown cells at the end of the notebook.

See **`CS2PP22_CW2 Example Notebook Template.pdf`** on Blackboard for an example of how you might incorporate these elements into your single notebook.

Rember to convert this to HTML for submission.

The Module Files

Individually produced module files with functions or classes will provide functionality to enhance the above outputs. **Their use should be explicitly highlighted in the Notebook.**

The Video

Finally, you will create a **5-minute video presentation** to demonstrate your final product. The intended audience consists of your peers, as these videos will be presented to the class during the final two lecture sessions. The video presentation may be created in any way you see fit (e.g., PowerPoint with voiceover, screen recordings, talking heads, etc.) and in any common format.

In any case, **your video must contain the following 4 features** (in any order that makes narrative sense):

- Introduction to your selected dataset and framing of your problem
- Introduction to your new Python package, rationale for its selection, and demonstration of its use (i.e., we want to watch execution of a cell)

- Explanation of how your selected in-depth analysis method is applied and why it is suitable to address your problem
- Key findings of interest alongside supportive visualisations (e.g., from EDA or model performance)

You may add other elements to your video that aid in communicating your results, time permitting.

Code Plagiarism

This coursework is expected to be the result of your own group effort, **not that of other people or systems**. That said, you should actually work closely with others on this coursework. You may employ pair programming techniques, for instance. 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 elements within Python scripts and code cells and supply formal literature references in a Markdown section at the end); otherwise, use of such material 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.

4. Marking Scheme

Element	Marks Available
Group Elements*	
Organisation: Preparation and submission of all required files in the correct format and with the correct naming conventions; notes on new Python package; project imports	**5**
Frame the Problem: Clarity and relevance of problem statement, understanding of project scope; *approximately 1 paragraph*	**5**
Collect Data: Appropriateness of chosen dataset, adherence to criteria (e.g., public accessibility, documentation), technical implementation of importing the data for analysis, well-commented code; *a few sentences and associated code*	**5**
Exploratory Data Analysis: Description of data, depth and thoroughness of EDA, insightful findings, use of visualisations, technical implementation of Python tools for analysis, well-commented code; *produce at least 3 visualisations and associated code*	**10**
Data Pre-Processing: Appropriateness and effectiveness of data cleaning and preparation techniques stemming from EDA results or to conform to the requirements of the analysis method, technical implementation of Python tools for analysis, well-commented code; *manipulate the data in at least 2 ways*	**5**
In-Depth Analysis: Implementation of the selected data analysis technique and evaluation methods, use of visualisations, technical accuracy, and sophistication of the approach, well-commented code; *produce at least 3 supporting visualisations and associated code*	**15**
Communicate Results 1, Written: Quality of the written report including structure, appearance, clarity, correctness in interpretations and grammar, and comprehensive coverage of the project components; *encompasses explanations and justifications of your implementations and discussion of results within the EDA, Pre-Processing and Analysis components above, as well as overall conclusions*	**15**
Communicate Results 2, Video: Effectiveness of the video presentation in conveying the 4 outlined key aspects of the video, clarity, and quality of delivery; *approximately 5 minutes*	**15**
Group Reflection: Quality of group reflection on teamwork, role distribution, and collaborative process; *approximately 2-3 paragraphs*	**10**

Continued on next page…

Individual Elements	
Individual Reflection: Depth of individual reflection on your contribution, challenges faced, and learning outcomes; *approximately 1 paragraph*	**5**
Individual Module Code: Quality and functionality of individual code contributions, adherence to naming conventions and intended scope, documentation as evidenced in the module files, integration into main analysis	**10**
Coursework 2 Total	**100**

*These elements are group marks. By default, marks earned in these categories will be applied equally to all members of the group. However, these elements will be ***proportionally weighted*** in the event contributions of effort were distributed unequally and noted as percentages in the group reflection section of your Notebooks.

5. References

Available as PDFs on the Blackboard CW2 Assessment Item

Hughes, R. L., & Jones, S. K. (2011). Developing and assessing college student teamwork skills. *New Directions for Institutional Research*, *2011*(149), 53–64. https://doi.org/https://doi.org/10.1002/ir.380

Salas, E., Sims, D. E., & Burke, C. S. (2005). Is there a “Big Five” in Teamwork? *Small Group Research*, *36*(5), 555–599. https://doi.org/10.1177/1046496405277134

6. Suggested Resources

Datasets

Bearing in mind the criteria for dataset selection, you might quickly find suitable datasets from:

https://archive.ics.uci.edu/ (**strongly encouraged; search by analysis method, well-documented**)

https://lib.stat.cmu.edu/datasets (often nicely documented)

https://snap.stanford.edu/data/ (for networks)

https://www.kaggle.com/datasets

https://datasetsearch.research.google.com/

https://registry.opendata.aws/

New Python Packages

Please consider the following resources. Some are much more complex than others. Aim for something simple. Avoid losing yourself in something complicated and challenging to understand.

Data Acquisition: Requests, Beautiful Soup, Scrapy, PyQuery, MechanicalSoup

Statistical Analysis: SciPy, Statsmodels, Pingouin, Researchpy, SymPy

Data Visualisation: Plotly, Bokeh, Altair, HoloViews

Model Evaluation: Yellowbrick, ELI5

Result Presentation: RISE, Dash by Plotly, Streamlit

GUI Development: Tkinter, PyQt/PySide, Kivy, wxPython

Website Development: Django, Flask, Bottle, CherryPy

Performance Enhancements, Parallel or Multi-threaded Processing: Multiprocessing, Concurrent.futures, Joblib, Dask, Numba, RAPIDS

Image Manipulation: Pillow, OpenCV, SimpleCV, Mahotas

Common Libraries: os, sys, json, collections, re, itertools, functools, subprocess, threading