

BTPR3203 Python for Data Science

Project (20%)

2025B

This is a group project, to be attempted in a group of **FIVE (5)** students at maximum. It is worth 20% of the overall marks of this course.

By the end of this project, you will be able to:

|  |  |
| --- | --- |
| CLO2 | Demonstrate the concepts and techniques of programming with Python by using modularisation, different structured types, and file and exception handling. (C3, PLO2) |
| CLO3 | Apply the knowledge of programming with Python and Python libraries to develop a solution. (C4, PLO8) |

**Project Title**

Social Media Sentiment Analysis for Social Impact

**Project Description**

In the age of digital communication, social media platforms serve as vast repositories of public opinion and sentiment. This project aims to harness this data to gain insights into public perception regarding a chosen social issue or domain (e.g., environmental awareness, community development, educational initiatives, social justice movements, consumer feedback on ethical products, or public response to policy changes).

Students are tasked to develop a Python-based application that processes an anonymised dataset of social media posts (e.g., tweets, comments). The application will analyse the sentiment expressed in these posts, identify key themes, and visualise trends. The ultimate goal is to demonstrate how data analysis can inform strategies, identify areas of concern, or combat misinformation within their chosen domain, thereby contributing positively to societal well-being.

**Core Requirements and Components**

* **Modularisation**:
  + The source code must be organised into logical, well-defined functions and classes.
  + Consider creating separate modules (e.g., data\_loader.py, sentiment\_analyzer.py, visualizer.py, main.py) to enhance organization and reusability.
* **Data Structures**: Utilise lists, dictionaries, NumPy Arrays and Pandas DataFrames for efficient data storage and manipulation.
* **File Handling**:
  + **Reading Data**: The application must be able to read social media data from a specified input file (e.g., a CSV file named social\_media\_data.csv or a JSON file).
  + **Writing Data** (Optional but Recommended): Ability to save processed data, sentiment scores, or summary reports to a new file (e.g., sentiment\_report.csv).
* **Exception Handling**: Gracefully handle potential errors that may occur during data processing.
* **Visualisation**:
  + Create at least **TWO (2)** distinct and informative visualisations of the analysed data. Examples include:
    - Sentiment trend over time (e.g., daily or weekly average sentiment).
    - Distribution of sentiment categories (positive, negative, neutral) using a bar chart or a pie chart.
    - Frequency of key terms or topics related to the chosen social issue.
    - Comparison of sentiment across different groups.
  + Ensure plots have appropriate titles, labels, and legends.

**Suggested Dataset**

Use an anonymised public dataset of social media posts relevant to their chosen social impact domain. The dataset should ideally contain:

* post\_id (unique identifier)
* timestamp (date and time of post)
* text (the content of the social media post)

**Deliverables**

1. **Source Code**: The GitHub repository link to your Python script.
2. **Sample Data File**: The input CSV/JSON file used for analysis.
3. **Project Report**: A report describing the following:
4. Project objectives and chosen social impact domain.
5. Explanation of data cleaning, sentiment analysis, and visualisation techniques.
6. Screenshots of the generated visualisations.
7. Discussion of findings and potential implications for their chosen social impact domain.
8. Challenges encountered and how they were overcome.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLO2:** Demonstrate the concepts and techniques of programming with Python by using modularisation, different structured types, and file and exception handling. (C3, PLO2) | | | | |
| **Criteria** | **Excellent** | **Good** | **Satisfactory** | **Needs Improvement** |
| Modularisation and Code Organisation (15%) | Code is exceptionally well-structured into logical functions, classes, and modules (.py files). | Code is well-structured with clear functions and classes. Some minor areas for improved modularity. | Code shows an attempt at modularization (functions, some classes) but may have inconsistencies, redundancy, or less logical grouping. | Code is monolithic or poorly organized, lacking clear functions/classes, making it difficult to understand or reuse. |
| Data Structures (15%) | Demonstrates expert and efficient use of lists, dictionaries, and NumPy arrays for appropriate data storage and manipulation. Optimal choice of data structure for each task. | Effectively uses lists, dictionaries, and NumPy arrays. Choices are generally appropriate, though minor optimizations could be made. | Uses lists and dictionaries adequately; NumPy array usage might be basic or less integrated, or choices of data structures are sometimes suboptimal. | Limited or incorrect use of appropriate data structures. Data handling is inefficient or prone to errors due to poor structure choices. |
| File Handling (10%) | Implements robust and efficient file reading/writing operations. Handles various file formats correctly and efficiently. | Correctly implements file reading/writing. Handles common file formats without significant issues. | Basic file reading/writing is functional but may lack efficiency or full handling of different file formats. | File handling is incomplete, incorrect, or causes errors. |
| Exception Handling (15%) | Comprehensive and graceful handling of all foreseeable errors using try-except blocks. Provides clear, informative error messages and ensures application stability. | Implements try-except blocks for most common errors. Error messages are generally helpful, and the application is mostly stable. | Basic try-except blocks are present for some errors, but significant potential errors are not handled, or error messages are vague. | Little to no exception handling. Application crashes on common errors without informative messages. |
| **CLO3:** Apply the knowledge of programming with Python and Python libraries to develop a solution. (C4, PLO8) | | | | |
| Correctness and Functionality (15%) | The application runs flawlessly, performs all required data loading, cleaning, sentiment analysis, and visualization accurately, and produces consistently correct outputs. | The application runs correctly with minor issues or occasional inaccuracies in output. Core functionalities are present and mostly work as expected. | The application is partially functional, with noticeable bugs or incorrect outputs in key areas. Some core functionalities may be missing or incomplete. | The application is largely non-functional, produces incorrect results, or fails to implement core requirements. |
| Effective Use of Libraries (15%) | Demonstrates sophisticated and highly effective use of libraries for data cleaning and manipulation, for compelling and insightful visualisations, and for efficient numerical operations. | Libraries are integrated appropriately, and visualisations are clear. | Usage might be basic, inefficient, or not fully leveraged for the task. Visualisations are present but may lack clarity or depth. | Limited or incorrect use of required libraries. Libraries are not integrated or used effectively to solve the problem. |
| Problem Solving and Solution Development (10%) | Demonstrates excellent problem-solving skills in developing a comprehensive and efficient solution. The approach is innovative and well-justified. | Demonstrates good problem-solving skills, leading to a functional solution. The approach is logical and generally effective. | Attempts to solve the problem, but the approach may be inefficient, overly complex, or incomplete. Requires significant guidance. | Unable to effectively solve the problem or develop a coherent solution. |
| Report and Presentation (5%) | The report is clear, concise, and comprehensive, detailing objectives, methodology, findings, and a strong discussion of the social impact within the chosen domain. Presentation is engaging and insightful. | The report is generally clear and covers most aspects, with a good discussion of social impact. Presentation is informative. | The report is somewhat disorganized or incomplete, with a basic discussion of social impact. Presentation may lack clarity or depth. | The report is poorly written or missing key information. Little to no discussion of social impact. Presentation is unclear or unprepared. |