Inferential Statistics and Applied Probability

Project Assignment

(DEADLINE: 25/12/2024)

REG#: ___________________ NAME: _________________________________

COURSE CODE: DS211 INSTRUCTOR: DR FAHAD BIN MUSLIM

TOTAL MARKS: 10

Instructions

- This project needs to be completed in groups. The group details have already been uploaded on Teams.
- You are free to consult other groups/peoples for verbal help. However, **copying or sharing the soft/hard copy with each other will not only result in the cancellation of the current assignment, but it may also impact your grade in the final exam as well**.
- List your collaborators on the last page of your assignment. Collaborators are any people you discussed this assignment with.
- You must attach this assignment at the top of your solution.

Problem Statement:

The grading system at universities often relies on either **absolute** or **relative** grading schemes to determine final grades. In an absolute grading system, grades are assigned based on fixed thresholds (e.g., >=90% = A, >=80% = A-, etc.), while in a relative grading system, individual grades are adjusted to fit a specified grade distribution (e.g., a normal distribution or percentile-based distribution). The goal of this project is to design and implement a Relative Grading System that takes student scores, analyzes their distribution, and adjusts the grades to conform to a predefined grade distribution while maintaining fairness and statistical validity. The system should also allow instructors to specify grade thresholds and provide analytics on the adjustments made.

This project also requires you to follow the **HEC guidelines** (from pages 15-20) provided in the attached document for the grade distributions and criteria. These guidelines will help shape the grade boundaries and statistical methods used in both grading schemes. Ensure that the design of the grading algorithms aligns with the official recommendations of the HEC for fairness and consistency.

You will work in groups of up to three students (already shared) to design, implement, test, and document this system. The system should include statistical calculations and visualizations to justify the adjustments.

Key Features to Implement:

1. Input Module:
   a. Allow the instructor to input student grades (e.g., via CSV or excel file).
   b. Provide an option for the instructor to select between **absolute grading** (e.g., fixed grade thresholds) or **relative grading** (e.g., adjusting grades to match a predefined distribution like a normal curve).
   c. If the instructor selects **relative grading**, optionally, let the instructor specify the desired grade distribution (e.g., percentages of A, B, C, D, F). If **absolute grading** is selected, allow the instructor to define the grade thresholds (e.g., >=90% = A).
2. Statistical Analysis:
   a. For **relative grading**, calculate descriptive statistics (mean, variance, skewness) for the input grades.
   b. Plot histograms or density plots of the grade distribution before and after adjustments.

   c. For **absolute grading**, the system should still provide summary statistics (e.g., the percentage of students falling into each grade category)

3. Grade Adjustment:
   a. **Relative grading**: Apply relative grading algorithms (e.g., curve fitting to normal distribution, z-score scaling, or percentile-based scaling).
   b. **Absolute Grading**: Apply absolute thresholds based on percentages and grade boundaries as defined by the instructor or the HEC guidelines.
   c. Ensure that the final distribution (for relative grading) matches the instructor's specified requirements, and the absolute grading applies the pre-defined thresholds.
4. Reporting and Visualization:
   a. Provide a detailed report showing the original and adjusted grades.
   b. Display visualizations (e.g., bar charts, histograms, boxplots) comparing the original and adjusted distributions.
   c. Provide summary statistics and insights (e.g., how many students moved between grade categories).

5. Error Handling: ensure the system handles missing or invalid inputs gracefully.
6. User Interface (Optional): implement a simple GUI or web interface for user interaction.

7. Deliverables:
   a. Code: well-documented Python code implementing the grading system.
   b. Report:
      i. A 4–5 page report detailing the methodology, algorithms used, and results.
      ii. Include the original and adjusted grade distributions with visualizations.
   c. Presentation: a 5-10 minute presentation demonstrating the project, including system workflow, results, and challenges faced.
   d. Demo: a live or recorded demonstration of the system in action.

8. Appendix:
   a. Include the code, sample input files, and outputs in the appendix of the report or as a zip file submission.

9. Grading Rubric:
   a. Functionality (40%): Does the system meet all the requirements?
   b. Statistical Accuracy (20%): Are the adjustments statistically valid and meaningful?
   c. Visualization (15%): Quality and clarity of visualizations.
   d. Code Quality (15%): Readability, documentation, and error handling.
   e. Report and Presentation (10%): Clarity and thoroughness.

Submission:
Submit all deliverables and reports (as a single document, PDF Only) on *Teams*.