CSE232 - Project 1

January 2025

Project Overview

This project will help you apply the knowledge you have learned so far in a practical scenario by comparing two datasets of movies from different genres. Make sure to read the entire document before starting the assignment. You will complete the project based on the specifications below and submit your work for grading via the D2L system.

The project is worth 40 points and must be completed no later than **11:59 PM on Thursday, 02/13/25**. You can submit the project up to 2 days after this deadline with 25% penalty per day.

This project is both important and practical as it applies key programming concepts to a real-world scenario that is analyzing and comparing movie reviews. By working with datasets, implementing statistical analysis, and performing sentiment evaluation using a "Bag of Words" approach, you gain hands-on experience in handling structured data and extracting meaningful insights. The project reinforces essential programming skills such as file handling, vector operations, and algorithm design while also introducing basic sentiment analysis, which has applications in various fields like social media monitoring, product reviews, and customer feedback analysis. Additionally, by considering edge cases and handling exceptions, you develop robust coding practices that are crucial for real-world software development.

Assignment Deliverable

The deliverables for this assignment are the following file(s):

proj01.cpp – your program code

If you choose to use generative AI, you should also submit your chat history with the AI:

history.txt – A text file containing either your chat history with the AI or a link to it

Assignment Specifications

- The purpose of the program is to compare two genres of movies based on the reviews and scores received from a social media group.
- The program will expect two dataset files and a dictionary file to be present in the same working directory as the program executable: 'set1.csv', 'set2.csv', and 'dictionary.txt'. Upon execution, the program will read the data contained in the two files to perform statistical analysis and compare the two data groups. There will be no need to provide additional inputs through command-line

arguments or during runtime. For example, here's a sample call to the program where 'proj01' is the name of the compiled executable:

```
./proj01
```

- After execution, the program will output the following criteria to compare the data groups:

  1. Data *count* (Total count of data including duplicate titles)
  2. Review scores *mean*
  3. Review scores *standard deviation*
  4. *Min* review scores
  5. *Max* review scores
  6. Number of *positive* reviews
  7. Number of *negative* reviews
  8. Number of *inconclusive* reviews
  9. The movie with the *highest* rating overall
  10. The movie with the *lowest* rating overall

- The input datasets will always have a correct format and will consist of 0 or more lines. Each line will have the following information in a single line with no line breaks:

  ```
  Title, Year, Rating, Review
  ```

  For example:

  ```
  Raiders of the Lost Ark, 1981, 8.4, There are puzzles to be solved and riddles
  broken, the dialogue's a joy, beautifully spoken, action packed from start to
  end, returns a massive dividend, engaging all the way and thought provoking.
  ```

- Movie ratings can range between 0 to 10.

- Functions for retrieving data from files and storing them in C++ vectors are provided to you in the base code (see additional information).

- Consider the 10 criteria used for comparing the two datasets. Criteria 1 - 5 can be calculated using basic statistical techniques. For criteria 6 - 8, you'll need to perform a sentiment analysis on the reviews to determine whether a review is positive or negative (see below). There are different approaches to calculate 9 - 10. For example, searching in a vector.

- An important programming skill is to think about edge cases. Much like real-life situations, for the rest of the projects, we won't tell you what edge cases will be used to test your code. For this project, however, to give you a sense of what you should consider when handling edge cases, the edge cases are as follows:

  ```
  - The case in which one or both of the datasets are empty
  ```

  ```
  - Multiple movies tie on having the most positive/negative ratings
  ```

```
- Multiple reviews for the same movie

- Empty dictionary file
```

Your code should handle such cases gracefully. This means your code should acknowledge the issue, print an appropriate message (in case of errors), and either exit the application or continue execution. Whether to exit or not is a design decision you need to make depending on the situation, but outputting a proper message in case of errors is a must. Any detail not specified in the project specifications is up to you to decide.

- In case a movie has multiple reviews, the scores given to the movie, and the positivity/negativity of the reviews should be averaged same as if the entries were for different movies.
- An expected output of the program for comparing the datasets is as follows:

```
                Set 1           Set 2
-----------------------------------------------
Count:          16              10
Mean:           6.96            6.72
STDV:           0.85            0.80
Min:            5.80            5.60
Max:            8.40            8.20
Pos:            9               5
Neg:            5               4
Inc:            2               1
Overall Best Title: Raiders of the Lost Ark
Overall Worst Title: The Dark Tower
```

You should aim to produce an identical output with regards to the titles and the information showed in every line. Of course, the values associated with each criterion will differ for different input datasets.

- You should implement the mean and the standard deviation functions on your own and shouldn't rely on available functions included in C++ libraries.

Sentiment Analysis

- For this program, you should implement a simple sentiment analysis technique called "Bag of Words". A dictionary file will be provided to you along with the dataset files. The dictionary file will contain two lines. The first line will include positive words, and the second line will include negative words. Code for extracting negative/positive words and storing them in two vectors is shared with you in the base code (see additional notes).
- To perform sentiment analysis, you should count the number of positive/negative words in each movie review.

If the count of positive words is greater than the negative words, you can consider the review to be positive.

If the count of negative words is greater than the positive words, you can consider the review to be negative.

If the count of negative and positive words is equal, you can consider the review to be inconclusive.

Additional Notes

- When displaying floating-point numbers, round them to **two decimal places**. Here's a link to a formatting guide in C++: https://www.geeksforgeeks.org/iomanip-in-cpp/
- You should use the same best practices for commenting/formatting your code as your other coding assignments. That includes providing attributions to AI tools.
- It is up to you to come up with a proper code structure for this project. That means that you should define functions that perform specific tasks and avoid coding everything inside the main function.
- Two sample datasets are provided for you along with the assignment specifications for testing purposes. For the most part, your code output should match the output provided in this specification as an example.
- Make sure to watch all the videos related to the project before starting.
- A base code will be provided to you as the starting point for this project. The base code will contain materials needed for the project but not covered in the course yet. This will include functions for retrieving data from files and saving them into vectors. For convenience, it is recommended that you build upon the base code.
- If you notice any errors or warnings during compilation, resolve them and compile your code again.
- When you don't know something, try searching for it. Don't just rely on AI tools. A strong programmer should be skilled at searching!
- Do not give away any part of the solution on Piazza.
- It's up to you which IDE to use when coding.
- Your submissions won't be graded based on time efficiency, however, your program should output results within an acceptable time-frame (less than 2 seconds).
- You must work on this project individually without collaborating with your peers.

Using Generative AI

You are allowed to use AI tools in this project, but you must do so responsibly. All the rules regarding AI usage included in the syllabus also applies to the projects.

DOs: Use AI to debug errors, generate ideas, and deepen your understanding of C++ concepts. Cite AI-generated code when applicable and make sure you fully understand and can explain any AI-assisted solutions.

DON'Ts: Don't just copy chunks of AI-generated code even if you fully understand it. For example, do not copy an entire function generated by AI. Your submission should be your own work and you should be able to explain every part of it. You shouldn't use AI to bypass the learning process. Misusing AI, such as submitting unmodified AI-generated solutions or failing to properly attribute sources may result in penalties. Use AI as a learning aid, not a shortcut.