MIE1628

Assignment 2: APACHE SPARK

Due Date: Jun 22, 2025

- *No submission is accepted via email. We have no exception.*
- *Students are responsible for submitting the correct files on time.*
- *Assignments submitted up to 48 hours late will incur a 20% penalty.*
- *Your grade will be zero if you submit your answer after 48 hours.*

Contact your TA for any questions related to this assignment or post clarification questions on the Piazza platform.

Submission Requirements:

- This assignment must be completed using PySpark or Spark SQL.
- Submit your code scripts (Python files or Jupyter notebooks).
    - A PDF document that details your methodologies, results, and theoretical insights and result screenshots as needed.

PART A: Advanced Data Analysis with PySpark or SQL Spark (35 points)

1. **Count Odd and Even Numbers** [Marks: 5]
    - Using the provided ***integer.txt*** file, develop a spark script to count the number of odd and even integers.
2. **Salary Aggregation with Statistical Analysis** [Marks: 10]
    - Analyze the **salary.txt** file to compute total salaries per department. Expand your analysis to investigate trends or discrepancies in salary distributions.
    - **Requirements**:
        - Utilize statistical measures (e.g. mean, median, standard deviation) and visualizations (e.g. box plots, histograms) to convey your findings.
3. **Implement an Optimized MapReduce** [Marks: 10]
    - Utilize the **shakespeare.txt** file to implement an optimized MapReduce operation that counts specific terms, allowing for case-insensitivity and punctuation removal.
    - **Requirements**:
        - Show how many times these particular words appear in the document: Shakespeare, What, The, Lord, Library, GUTENBERG, WILLIAM, COLLEGE and WORLD. (Count exact words only)
4. **Word Frequency and Distribution Analysis** [Marks: 10]

- From **shakespeare.txt**, calculate top 10 and bottom 10 words. Show 10 words with most count and 10 words with least count.

PART B: Advanced Recommender System with Apache Spark (65 points)

The objective of this part is to develop a sophisticated distributed recommender system using Apache Spark's capabilities, emphasizing accuracy, efficiency, and comparative evaluations.

Data Input: Utilize the provided **movies.csv** dataset, available for download from Quercus.

Implementation Steps: Load the dataset and import the necessary libraries. Address the below questions, applying advanced techniques and theoretical principles.

1. **Data Description and Insights Analysis** [Marks: 10]
   - **Description**: Provide a detailed description of the dataset's structure and contents. Analyze the distribution of ratings across each movie and identify the top 10 movies with the highest average ratings.
   - **Requirements**:
     - Identify the top 10 users who have contributed the most ratings (not just high ratings) and discuss their influence on the dataset.
     - Conduct exploratory data analysis, such as visualizing the distribution of ratings, to uncover patterns in user behaviors (e.g., how many ratings each user provides) and preferences.
     - Discuss any potential implications for marketing strategies based on user engagement and rating tendencies.
2. **Split Dataset and Performance Assessment** [Marks: 10]
   - **Experiment**: Split the dataset into training and testing subsets using 2 different ratios (for e.g. 60/40, 70/30, 75/25, and 80/20). Implement stratified sampling to ensure users are represented proportionally across the splits.
   - **Requirements**:
     - Report on how different splits influence the performance of your collaborative filtering model (you can use one of the evaluation metrics to show this).
     - Represent the performance variation of the model based on each split ratio, identifying the most effective configuration based on empirical findings.
3. **In-Depth Evaluation of Error Metrics** [Marks: 10]
   - **Metrics**: Define and explain key metrics for evaluation: MSE, RMSE and MAE. Introduce advanced metrics like Precision, Recall, and F1 Score specifically tailored for recommendations.
   - **Requirements**:
     - Provide a detailed evaluation of each model's performance using these metrics, discussing the strengths and weaknesses of each in the context of a recommendation system focused solely on user ratings.
     - Make observations about the trade-offs involved when selecting different metrics, particularly in scenarios of sparse data or imbalanced ratings.

4. **Hyperparameter Tuning Using Cross-Validation Techniques** [Marks: 20]
    - **Tuning**: Conduct systematic hyperparameter tuning for at least two parameters of the collaborative filtering algorithm, such as rank, regularization, or iterations etc. utilizing methods like grid search or randomized search combined with cross-validation.
    - **Requirements**:
        - Visualize the impact of different hyperparameter configurations on model performance (e.g., varying RMSE scores) and provide a rationale for your tuning choices based on your findings.
        - Discuss how each parameter affects model performance and overall training time, including insights on how to balance complexity against performance.
5. **Personalized Recommendations and Analysis for Selected Users** [Marks: 15]
    - **Recommendations**: Generate personalized movie recommendations for user IDs 11 and 21 based on their rating preferences.
    - **Requirements**:
        - Discuss how the collaborative filtering approach utilizes user ratings to generate these recommendations and the effectiveness of this technique for a dataset with limited features.
        - Compare performance outcomes between the refined recommendations for these users and any baseline recommendations generated earlier in your analysis. Discuss potential enhancements or features that could be added for improved personalization in future iterations.

Additional Notes:

- Ensure clear documentation of your code and methodologies throughout the assignment.
- Include visualizations (e.g., graphs, charts) where appropriate to reinforce your findings and enhance result clarity.
- The PDF report should detail your methodologies, findings, and insights, citing any references for external sources utilized during your research and code development.