CSE 587 – Fall 2024
Assignment 2
Due by: November 12, 2024, 11:59 PM

.

Topics: Distributed Processing and Analysis using Spark

Overview

This assignment builds on core PySpark skills by implementing word count, co-occurrence tasks, and analyzing data flow. You will deepen your understanding of distributed data processing in Spark, emphasizing RDD transformations, data streaming with DStreams, and handling biases in datasets. The programming tasks are structured to develop your Python and PySpark programming skills and consolidate the discussions around distributed processing of big data.

General Requirements

- **Work Environment:** Use PySpark in Jupyter Notebook, Jupyter Lab, or Google Colab.
- **Submission Format:** Submit a zip file containing your `src/` directory with all code files and a comprehensive PDF report with answers and screenshots of results.

Part 1: Word Count and Data Analysis with PySpark [50 Points]

1. **Basic Word Count Implementation** [20 Points]
   Write a basic word count program in PySpark that:
   - Reads a text file (e.g., any book from Project Gutenberg).
   - Outputs a list of words with their occurrence counts as key-value pairs in descending order.
   - You will explain why you choose this particular book for analysis and will also explain and comment about the result of analysis.
2. **Extended Word Count** [20 Points]
   Extend your word count program with the following additional features:
   - Make it case-insensitive, remove punctuation, and ignore stop words.
   - Output the top 15 most common words with their counts.
3. **Data Flow Analysis** [10 Points]
   Answer the following:
   - Describe how many stages your word count program execution is broken into. Include a screenshot of the DAG.
   - Draw a simple lineage graph (DAG) of your RDD transformations.

Part 2: Word Co-Occurrence with PySpark [30 Points]

1. **Word Co-Occurrence Implementation** [20 Points]
   Implement a word co-occurrence program using PySpark that:
   - Reads the same text file.
   - Processes the text by normalizing to lowercase, removing punctuation, and filtering stop words.
   - Generates and counts bigrams (two consecutive words) in the text file.
   - Outputs the 10 most common bigrams.
2. **Analysis of Co-Occurrence** [10 Points]
   Describe the role of bigrams in analyzing word relationships. Answer the following:
   - Why might analyzing bigrams be valuable in text processing?
   - Compare the bigram analysis to the results from the word count. Include insights on the most common words and bigrams and any patterns observed.

Part 3: Data Bias and Review [20 Points]

1. **Bias Identification** [10 Points]
   For each scenario, identify the type of bias and propose a potential solution:
   - A fitness app tracks user activity but excludes data from users without internet access.
   - A recruitment algorithm favors candidates from top universities without considering applicants' experiences or skills.
2. **Review of Streaming Data Paper** [10 Points]
   Write a 300-word review on the paper *"Discretized Streams: Fault-Tolerant Streaming Computation at Scale"* covering its main concepts and its significance to Spark Streaming.

Submission Instructions

- Include a single `src/` folder with your code and a PDF report covering all responses.
- Submit on Brightspace by (Deadline date) at 11:59 PM. No late submissions accepted.
- Your explanation will help us understand and distinguish your work from others.

Additional References for Learning Spark

To further explore Spark and find tutorials, you can use the following resources:

- Zaharia, M., Das, T., Li, H., Hunter, T., Shenker, S., & Stoica, I. (2013). *Discretized Streams: Fault-Tolerant Streaming Computation at Scale.* Available at:
  `https://dl.acm.org/doi/pdf/10.1145/2517349.2522737`
- **Apache Spark Documentation Archive** – Choose the version of Spark you are using at the following link to access related documents and tutorials:
  `https://archive.apache.org/dist/spark/docs/`