Bonaobra, Aaron Stefano F.
BSIT – 3A

Big Data Laboratory 4 : Documentation

1. Introduction

This project focuses on analyzing a dataset of car prices using Apache Spark (PySpark). The goal is to perform data preprocessing, filtering, and SQL-based querying to extract meaningful insights. The results are then stored in CSV, JSON, and TXT formats for further analysis.

2. Project Setup & Environment

Tools & Technologies Used:

- Python 3.x
- Apache Spark (PySpark)
- Pandas (for easier handling of final data output)
- OS & Shutil (for file management)
- Hadoop Dependencies (for Spark file operations on Windows)

Folder Structure:

C:/For School/Big Data/Lab4/

-- car_price_dataset.csv # Input dataset

-- filtered_car_prices.csv # Processed data output

-- top_brands.json # JSON output

-- summary.txt # TXT summary

3. Data Preprocessing

- Loading the Dataset: The raw car dataset is read into a PySpark DataFrame.
- Data Cleaning: Missing values are dropped, and column names are standardized.
- Type Casting: Ensuring numerical fields like year are properly cast as integers.

4. Data Processing & Filtering

- Cars priced below $5000 are removed to maintain relevant listings.
- A temporary SQL table is created to enable complex querying.

SQL Queries Executed:

1. Top 5 Most Frequent Car Brands (based on listing count)
2. Average Price Per Brand (sorted in descending order)

5. File Output & Storage

- CSV Output: Final processed dataset is converted into a single CSV file.
- JSON Output: The extracted top brands are stored in a structured JSON file.
- TXT Output: A human-readable summary of the top car brands is generated.

6. Challenges & Solutions

Issue: PySpark writes CSV as a folder instead of a single file.

- Solution: Convert Spark DataFrame to Pandas and save it using to_csv().

Issue: JSON file throws Permission Denied (Errno 13) on Windows.

- Solution: Ensure that the file is not being accessed by another process before writing.

Issue: TXT file encoding error (charmap codec can't encode character).

- Solution: Set encoding='utf-8' while writing to avoid unsupported character errors.

7. Conclusion

This project successfully demonstrates big data processing with PySpark, showcasing:
- Efficient data cleaning & transformation
- Complex SQL-based queries for insight extraction
- Proper file output handling in different formats

This analysis can be expanded further by integrating machine learning for price prediction or deeper trend analysis!

8. Future Enhancements

- Automate dataset updates by integrating with live car listings.

- Improve filtering by adding more dynamic criteria (e.g., fuel type, brand).

- Use ML models to predict car prices based on historical trends.

By leveraging PySpark, we can process massive datasets efficiently, making it a powerful tool for real-world analytics.

Thank you, sir!