Assignment 1: Advanced Intelligent Data Ingestion, Transformation, and Exploratory Data Analysis (EDA)

Timeline: 13th February 2025 – 21st February 2025

Instructions

1. **Plagiarism Policy:**
    - Any form of plagiarism will result in disqualification.
    - Ensure all work is original.
    - Cite any external sources used in your submission.
2. **Dataset:**
    - Use the dataset from Assignment 0: Amazon Review Data.
    - Dataset Link: Amazon Review Data
    - Ensure you use the same loaded dataset as in Assignment 0.
3. **Git Repository:**
    - Maintain a private Git repository for this assignment.
    - Push your work regularly to GitHub.
    - Ensure proper folder structure (eda/, visualizations/, logs/).
4. **Naming Conventions:**
    - Use clear and meaningful names for scripts, functions, and variables.
    - Examples: eda_summary.py, review_distribution.png, sentiment_analysis.ipynb.
5. **Documentation:**
    - A README.md must be included with:
        - An overview of the assignment.
        - Steps to run your EDA code.
        - Key insights and findings.
    - A report that will be in the zip folder you upload on the GCR.

Objective

This assignment focuses on developing an advanced real-time data ingestion and transformation pipeline while conducting an in-depth, interactive Exploratory Data Analysis (EDA) to extract meaningful insights. The assignment consists of two primary components:

1. Real-time Data Ingestion & Transformation

- Utilize **Apache Spark Structured Streaming** to create a scalable data pipeline.
- Implement **dynamic** data cleaning, preprocessing, and **real-time anomaly detection** during ingestion.

2. Advanced Interactive Exploratory Data Analysis (EDA)

- Perform an in-depth, **query-driven analysis** and visualization of the processed data.
- Identify trends, patterns, and correlations through **multidimensional EDA**.

Assignment Breakdown

1. Data Ingestion & Transformation (Apache Spark Streaming)

Students are required to build a **real-time data pipeline** that ingests data from a source (e.g., **Kafka, socket, or local file stream**). The pipeline should:

- **Stream and preprocess** data dynamically.
- Perform **data cleaning**, including handling missing values, removing duplicates, and normalizing data.
- **Detect outliers and anomalies** using **Spark SQL or MLlib**.
- Convert **raw data into a structured format** suitable for analysis.

Expected Deliverables

- A **Python script or Jupyter Notebook** implementing the pipeline using **Apache Spark**.
- A **GitHub repository** containing well-structured and modular scripts.
- **Error-handling mechanisms** to ensure smooth data ingestion.

2. Advanced Exploratory Data Analysis (EDA) on Transformed Data

After ingestion and cleaning, students must conduct an **in-depth, query-driven EDA**, including:

Step 1: Data Understanding & Summary Statistics

- Evaluate dataset shape, column types, missing values, and **basic statistics**.
- Summarize dataset using **mean, median, mode, min/max, standard deviation**.

Step 2: Meaningful Query-Based EDA

Perform **at least 8 meaningful queries** from the dataset. Examples include:

1. **What are the top 5 most reviewed products?**
2. **How do average review ratings differ across product categories?**
3. **What is the correlation between review length and review rating?**
4. **How do reviews trend over time?** (time-series analysis)
5. **What percentage of reviews mention words like 'refund', 'return', or 'defective'?**
6. **Which brands/products have the most polarized reviews (most 1-star and 5-star reviews)?**
7. **How do verified and non-verified purchases compare in ratings?**
8. **Which product categories tend to have the most fake-looking reviews (e.g., excessive repetition of words)?**

Step 3: Text Analysis (Basic NLP)

- **Most Frequent Words**: Generate a **word cloud** after removing stopwords.
- **Sentiment Analysis**: Classify reviews as **positive, neutral, or negative** based on ratings.
- **Topic Modeling (Bonus)**: Implement **LDA** to find common themes in reviews.

Step 4: Correlation & Business Insights

- Identify **highly reviewed products and top-rated categories**.
- Detect **which words correlate with positive/negative reviews**.
- Determine which features impact a product's success **using correlation heatmaps**.

Step 5: Interactive & Visual EDA

- **Generate at least eight insightful, interactive visualizations** using **Plotly, Dash, or Altair**.
- Store all visuals in the **visualizations/** folder.

Bonus Dataset

In addition to the main dataset, students will be provided with a **bonus JSON dataset**. **Performing EDA on this dataset will result in extra marks.**

- The bonus dataset will contain **user behavioral data** (clicks, views, add-to-cart actions).
- Students can **combine insights** from both datasets for a **richer analysis**.
- The **bonus dataset file will be included in the assignment package**.

Submission Requirements

1. GitHub Repository

- Push all work regularly to a **private GitHub repository**.
- Ensure a proper **folder structure** (eda/, scripts/, README.md).
- Name files meaningfully (e.g., eda_summary.py, streaming_pipeline.py).

2. Report Submission

- A **README.md** explaining:
    - Steps to **run the ingestion pipeline**.
    - **Summary of EDA findings**.
- A **PDF report** containing:
    - Visualizations
    - Key insights from query-based analysis
    - Discussion on trends, patterns, and business implications.

3. Logging & Error Handling

- Maintain logs in a **logs/** folder to track **errors and processing time**.

4. Code Documentation

- Ensure the code is **modular, well-commented, and includes function descriptions**.

5. Final Upload to GCR

- Submit a **zip folder** containing:
    - **All scripts** (.py / .ipynb)
    - **Final PDF report**
    - **Bonus dataset analysis (if done for extra marks)**
- **Zip folder name:** [RollNumber1]_[RollNumber2].zip

Evaluation Criteria

Criteria	Weightage
Real-time Data Ingestion & Cleaning	30%

Criteria	Weightage
Query-Driven EDA & Visualizations	30%
Code Quality & GitHub Usage	20%
Documentation & Report Clarity	20%

Bonus Points

- **Use Spark MLlib** to detect anomalies in review ratings.
- **Develop an interactive dashboard** using **Dash, Streamlit, or Voila**.
- **Optimize Spark jobs** for better performance (**caching, partitioning**).
- **Incorporate Bonus Dataset** for additional insights.

Deadline

21st February 2025 (End of day)

Students are expected to maintain **regular commits on GitHub** and ensure their work is **well-documented and reproducible**. Each assignment should have an individual folder i.e. Assignment 1, Assignment 2 and so on...