Live processing with AWS (6 Weeks)

Steps to Implement the Workflow: [2 Week]

- **Common Steps:**
  - **Data Ingestion**:
    - Read data from the csv which reside inside bucket or local system.
    - Python script will write data into the kafka topic using producer.
  - **Data Processing with Spark**:
    - Then you've to subscribe the topic and consume data using the spark structure streaming.
    - Save the processed data into an output bucket in Parquet format with a filename pattern **dd_mm_yyyy_filename**.
  - **Loading Data into SQL Database**:
    - Use AWS Glue (for AWS) to load the data from the output bucket to the RDS database.
    - You've to use medallion architecture for this aws glue pipeline.

Detailed Steps for AWS:

1. **Data Ingestion with Kafka**:
   - Create a Kafka producer in Python to read CSV files and send the data to a Kafka topic.

   Data Processing with Spark Structured Streaming:
   - Create a Spark Structured Streaming job to subscribe to the Kafka topic.
   - Process the data and write it to the Bronze layer in Amazon S3 in Parquet format (**dd_mm_yyyy_filename.parquet**).
   - Create subsequent Spark jobs to transform data from Bronze to Silver, and from Silver to Gold layers.
2. **Loading Data with AWS Glue**:
   - Create Glue jobs to load data from the Gold layer in Amazon S3 to Amazon RDS/Aurora.
   - Schedule these jobs to run after the Spark jobs complete.

Data Exploration and Machine Learning Modelling: [2 Week]

Parts below outline the steps for data exploration, implementation of machine learning (ML) modeling methods, and development of a user interface (UI) for querying data using natural language. The goal is to achieve the best fit for the model, to showcase findings using charts and dashboards along with a basic UI to facilitate in question–answer on the data.

2. Data Exploration

Utilising the data from the pipeline created above, include below parts in the data handling.

2.1 Data Preprocessing

- Handle missing values.
- Normalize and standardize data if necessary.
- Encode categorical variables.
- Split the data into training and testing sets using a 70-30 split ratio (0.3 parameter for train-test split).

2.2 Exploratory Data Analysis (EDA)

- Perform statistical analysis to understand data distribution.
- Visualize data using histograms, scatter plots, box plots, and correlation matrices.
- Identify any patterns, trends, and outliers in the data.

3. Machine Learning Modelling

3.1 Model Selection

- Experiment with various Machine learning and deep learning models appropriate according to data.
- Compare model performance using appropriate metrics based on data.

3.2 Model Training and Validation

- Train models on the training set.
- Validate model performance on the test set.
- Perform hyperparameter tuning to optimize model performance.

3.3 Model Evaluation

- Evaluate the models using appropriate metrics.

- Select the model with the best performance for deployment.

4. User Interface Development

4.1 LLM (large language model) Integration

- Implement a UI that allows users to ask questions about the data using natural language.
- Utilize LLM to understand and process user queries.
- Provide accurate and relevant answers based on the data.

4.2 UI Design

- Develop a user-friendly interface for data querying.
- Ensure the UI is intuitive and easy to navigate.

4.3 Features

- Allow users to input questions in natural language.
- Display answers in a clear and concise manner.
- Include options to visualize the queried data.

5. Findings and Visualization [1 Weeks]

5.1 Findings

- Summarize key findings from data exploration and model evaluation.
- Highlight important insights and patterns discovered during EDA.

5.2 Visualization

- Create charts and dashboards (powerBI or Tableau) to visualize key findings.
- May use tools such as Matplotlib, Seaborn, and Plotly for creating visualizations along with other methods.
- Include visualizations such as(not limited to):
    - Bar charts
    - Line graphs
    - Heatmaps

- Pie charts
- Interactive dashboards

Diagram for AWS Setup:

- **Amazon S3**: Storage for raw and processed data.
- **Apache Spark**: Read, process, and write data.
- **AWS Glue**: Load data from Amazon S3 to SQL Database.
- **Amazon RDS/Aurora**: Destination SQL database.
- Technical document

6. Conclusion

This document outlines the steps to setup data pipeline, implement ML models, and develop a natural language-based querying UI. The objective is to find the best-fitting model and effectively communicate findings through visualizations and interactive dashboards.