Assignment: Building an ETL Pipeline with Apache Airflow

Due Date: 27th June 2024, 2 PM CEST

Objective

The objective of this assignment is to introduce students to Apache Airflow and its application in building ETL (Extract, Transform, Load) pipelines. The students will gain hands-on experience in orchestrating data workflows, a common task for ML engineers. This assignment will also introduce the use of a NoSQL database for storing processed data.

Assignment Overview

You are tasked with building an ETL pipeline using Apache Airflow that processes and prepares data for a machine learning model. The dataset to be used is the UCI Machine Learning Repository's "Online Retail" dataset, which contains transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.

Dataset

- Online Retail Data Set: UCI Machine Learning Repository - Online Retail

Steps to Complete the Assignment

1. **Setup Apache Airflow:**
   - Install Apache Airflow on your local machine or use a cloud-based service (like Google Cloud Composer, Amazon MWAA, or Astronomer).
   - Set up a new Airflow project and create the necessary directories and files.
2. **Download the Dataset:**
   - Write an Airflow DAG that downloads the Online Retail dataset from the UCI Machine Learning Repository.
   - Ensure the dataset is stored in a designated data directory within your project.
3. **Data Cleaning:**
   - Create a task in your DAG to clean the dataset. This should include:
     - Handling missing values.
     - Removing any duplicates.
     - Converting data types as necessary (e.g., dates).
4. **Data Transformation:**
   - Add a task to your DAG that performs the following transformations:
     - Add a new column for total price (quantity * unit price).
5. **Loading Data to NoSQL Database:**
   - Load the transformed data into a MongoDB.
   - Ensure that the NoSQL database is accessible from your Airflow environment.
6. **Triggering the ETL Pipeline:**
   - Schedule the DAG to run daily, simulating a real-world scenario where new data is processed daily.
7. **Documentation:**
   - Document your code and the steps taken in a PDF. Include explanations for each Airflow task, the DAG structure, and how each part of the pipeline contributes to preparing the data for machine learning.
8. **Optional - Monitoring: (For extra credits)**
   - Set up Airflow task monitoring and alerts for any failures.

Deliverables

1. **Airflow DAG Code:**
   - Python files defining the Airflow DAG and tasks.
2. **Documentation:**
   - A pdf file explaning your ETL pipeline.
3. **NoSQL Database:**

    - A snapshot of the NoSQL database after loading the transformed data.
4. **Execution Logs:**
    - Logs from Airflow showing successful execution of the DAG.

Grading Criteria

- **Correctness:** (40%)
    - Accurate implementation of the ETL pipeline.
    - Correct handling of data cleaning and transformation.
- **Code Quality:** (20%)
    - Clean, readable, and well-documented code.
    - Adherence to best practices in Python and Airflow. Such as the use of variables or connections to store database connection details instead of hard coding in code.
- **Documentation:** (10%)
    - Comprehensive documentation explaining the pipeline and each task.
- **Execution:** (20%)
    - Successful execution of the Airflow DAG.
    - Evidence of data being correctly loaded into the NoSQL database.
- **Monitoring:** (10%)
    - Alerts for failures
    - Extra task to send a summary email after completion of execution.

Submission Instructions

- Submit a zip file containing all the required deliverables.
- Include a README file with instructions on how to set up and run your Airflow project.

Additional Resources

- Apache Airflow Documentation
- UCI Machine Learning Repository - Online Retail Data Set
- MongoDB Documentation
- Pandas Documentation

If you have any questions, feel free to reach out to me. Good luck!