Assignment 1
Data Processing Pipelines

CS611 - Machine Learning Engineering
Version: April 2025

Objectives

The objectives of this exercise are as follows:

1. Understand and build ETL pipelines compliant to Medallion Architecture
2. Assemble machine learning compatible feature and label store with data pipelines
3. Work with Docker containers to manage environments and dependencies
4. Practice git and writing code to repositories
5. Practice deck building, documentation and presentations

Context

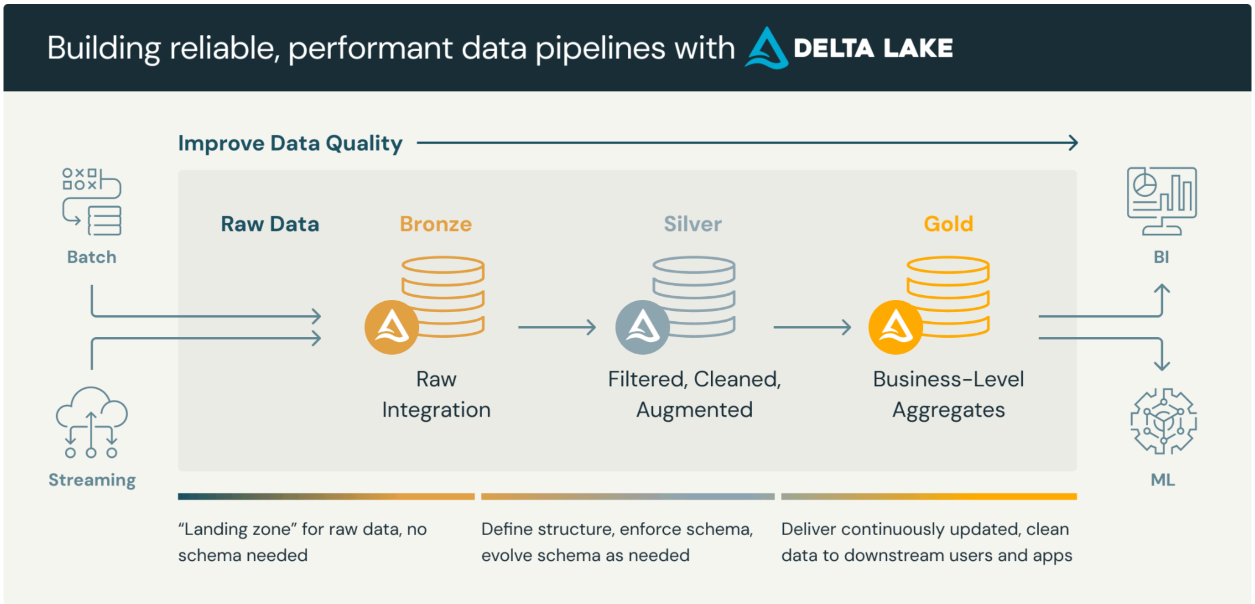
You are a data scientist working at a financial institute (e.g. bank). Your company lends money to users in the form of cash loans. You are tasked to eventually build a machine learning model that can predict whether a user will default on their loan at the point of loan application. For this assignment, you will be preparing the data through production data pipelines in preparation for machine learning model training (in the next assignment). You have 2 tasks: prepare the feature and label data stores compliant to the Medallion Architecture, and prepare a presentation deck (max 10 slides) to present to your manager, other engineers and business users of what tables you will create under which parts of the Medallion Architecture.

Task 1: Building bronze, silver, gold tables feature stores with data pipelines (10 marks)

Explore the data provided and design feature store table(s) that is compatible with a classification machine learning model. You are only required to build the data pipeline for the feature store table(s) from raw data up to gold table(s). The machine learning model pipeline will be in the next assignment.

To help you design your feature store, the label store has already been built in Lab 2. You can reuse the codes you developed in Lab 2.

Your eventual feature store and label store will be gold level table or tables (depending on your design choice). Ensure that all pipelines and jobs are compliant with Medallion Architecture that you learnt in class. This assignment is most similar to Lab 2.


Building reliable, performant data pipelines with DELTA LAKE
Improve Data Quality
Batch
Streaming
Raw Data
Bronze
Silver
Gold
Raw Integration
Filtered, Cleaned, Augmented
Business-Level Aggregates
BI
ML
"Landing zone" for raw data, no schema needed
Define structure, enforce schema, evolve schema as needed
Deliver continuously updated, clean data to downstream users and apps

Source: https://www.databricks.com/glossary/medallion-architecture

You are to treat the data provided as raw data and build your own bronze, silver and gold tables from that raw data, similar to what was done in Lab 2 but for your feature store table(s).

You are to use python, pyspark, docker and github repositories to practice these tools. There is no right or wrong answer to this task, but there are better design and code modularity practices and not so good ones, so do your best to keep things organized! Read the **Submissions** and **Assessment** sections on how to package up your code for submission.

To know that you are on the right track, you can fit a simple machine learning binary classification model using your feature store as input data and label store as classification labels. Your simple model should be able to train and run. You do not need to submit this model for this assignment, but this is a good sanity check to understand if your feature store is machine learning compatible and if you have any data leakage (refer to Data Provided section for more about data leakage).

Task 2: Build a presentation deck that documents your data pipeline (5 marks)

Build a presentation deck on your design choices and implementation from Task 1. Assume that this deck will be circulated to technical and non-technical colleagues and that you do not have a chance to present it to them; build the deck as a slideument.

A **slideument** is a hybrid between a **slide presentation** and a **document** — a presentation slide that is designed to function as both a visual aid during a presentation and a handout document. (very common in the business world!)

This is a powerful exercise as it allows you to sharpen your deck building, visualisation and storytelling skills which are vital at the workplace for visibility of your work. You can also get to use this in future tech interviews if ever they ask you what experience you have on machine learning engineering! The real benefactor of this task is really you! :)

Data Provided

You are provided with the following data about users:

1. feature_clickstream.csv
2. feature_attributes.csv
3. feature_financials.csv

You are provided with the following data about loans:

1. lms_loan_daily.csv

You are encouraged to perform Exploratory Data Analytics to discover what features you want to build in your pipeline.

BE CAREFUL OF DATA LEAKAGE! Data leakage (also called leakage or target leakage) is a common issue in machine learning where information from outside the training dataset — specifically from the future or from the target variable — is accidentally used to create the model. This leads to overly optimistic performance during training or validation, but poor generalization to unseen data. Types of Data Leakage:

- Target Leakage: Happens when the model has access to data that would not be available at prediction time. Example: Including a column like "Loan Paid Off" when trying to predict loan default — that’s the answer!
- Train-Test Contamination: Occurs when the test data somehow influences the training process. Example: Normalizing your full dataset before splitting into train/test, so info from test leaks into training.
- Temporal Leakage: Using future data to predict past or present. Example: Using stock prices from a week ahead to predict today’s price.

Submissions

Due date: **21 May 2025**

You are expected to upload the following to eLearn:

1. A zip file of your code artefacts (excluding datamart folder that your main.py script will create when run) from task 1. The zip folder should unzip to contain the following files and subfolders (similar to Lab 2):
    a. A python script called **main.py** that runs the whole pipeline.
    b. Dockerfile
    c. docker-compose.yaml
    d. requirements.txt
    e. utils (folder, if you want to put any code in here)
    f. data
    g. Readme.txt (with just 1 line to your github repo link)
2. PDF presentation deck (max 10 slides) from task 2.

Assessments

We will assess as follows **(total 15 marks)**:

1. We will unzip your zip.file into a folder and run the command in the terminal "**docker-compose build**" and "**docker-compose up**". Your docker-compose up should provide a link to JupyterLab (similar to Lab 2). If we can enter JupyterLab from docker-compose up, you get **5 marks**!
2. We will run your **main.py** by `python main.py`. Your script should create a datamart folder with bronze, silver and gold subfolders (similar to Lab 2) to represent the bronze tables, silver tables and gold tables in the datamart. If your main.py can run and create this datamart, you get **5 marks**!
3. We will read through your presentation deck and grade it as follows:
    a. Data pipeline technical design decisions and explanations: **3 marks**
    b. Is the deck pretty? Is it worthy of corporate / business standards? **2 marks**