Data Engineering W24 - Milestone 2 Description

DEADLINE: WEDNESDAY 27th NOV 2024 @ 11:59 pm

1 Introduction

The objective of this milestone is to package your milestone 1 code in a docker image that could be run anywhere. In addition, you will load your cleaned and prepared dataset as well as your lookup table into a PostgreSQL database which would act as your data warehouse.

2 Requirements

For this milestone your dependencies so far would be python (and the necessary packages) and PostgreSQL. Each of these dependencies should be a separate image that would run under the same network(docker compose).

2.1 Building the python image:

- You must use the python:3.11 image as your base image.
- Convert your notebook into a python script. Why? because the code you have written to clean the dataset must be executed inside the container from the terminal. The format (ipynb/python notebook) is not an executable file and cannot be run from the terminal.
- Important notes regarding converting from ipynb to py:
  - If you implemented your cleaning code as functions, you should have at least two python scripts. One for listing the functions you created in M1 and the other would be your main script were you call these functions.
  - In your script you only need the functions/code that changes the dataset (I.e. cleaning, encoding, disc., adding features). You DO NOT need to include code you have written to analyze the dataset to reach your decisions (i.e. you do not need the basic EDA, 5 questions, any visualizations or debugging code).
  - In your main.py, you should first check if the cleaned dataset exists, if it does then you do not need to call the functions. Check the method from the OS package os.path.exists(path)
- Add the functions required to connect to the database and upload your cleaned dataframe AND the lookup table into the database (you should not re-upload the dataframe if the table already exists, check 'if_exists option' here)
- All in all your python script should check if the cleaned csv exist,
  - If it doesn't, read the csv file, prepare the dataset, save to new csv, connect to postgres and load the dataset
  - If it does, then you can load into PostgreSQL directly.

- For your python image you should separate your source code and datasets in different directories(check lab 5)
- Add your package dependencies(pandas,etc) in the image.
- DO NOT copy any files from host to container. Instead mount volumes.
  - Important: When mounting volumes make sure to have your relative host path and not the full path. i.e. the path relative to your docker file and to the absolute path (c:/users/...). The reason being that this way the path is agnostic to your host machine and anyone could run the image easily.
- When you run the image it should execute the main.py file.

2.2 PostgreSQL image:

- You must use postgres:13 as your image.
- Make sure to mount volumes for the database data and expose the appropriate ports. Your directory at the host where the database data will reside MUST be named postgres_data_mount
- Place your SQL queries in a directory called 'm2_queries' and mount this directory as well into the PostgreSQL container at /var/lib.

2.3 SQL queries:

Write SQL queries to answer the following questions. Place the sql queries in a file called m2_queries.sql under the m2_queries folder.

1. All info for the 20 highest loan amount.
2. What is the average income of a person per state.
3. On average, which state has highest interest rate.
4. On average, which state has lowest interest rate.
5. What is the most frequent grade for each state.
6. Which state is less likely to pay the loan.
7. Between 2015 and 2018 what is the average loan amount.

Take screenshots of the output (whether that be from the terminal or Pgadmin) and place them in another directory called queries_screenshots(under your root directory/folder (m2_docker)).

You are more than free to add more containers within your network such as Pgadmin to easily run your sql queries(check lab5 task).

3 Submission guidelines

DEADLINE: WEDNESDAY 27th NOV 2024 @ 11:59 pm

You should simply create a folder containing the required files/folders to build both images and run them(similar to lab 5).

- At the root of the folder there should be a docker-compose.yaml file such that by running docker-compose up the pipeline would be run (cleaning and preparing and then loading to PostgreSQL database).

- The root folder that you will create MUST be named m2_name_major_id.
- Upload your folder as a zip folder in your Milestone 2 drive folder.
- Directory names and path name MUST be as instructed
- EXTREMELY IMPORTANT: In your uploaded zip folder DO NOT INCLUDE the following:
  - Database data info that is mounted on to the container (the postgres_data_mount folder)
  - The cleaned CSV file

  The reason being is so that the pipeline could be run as an initial run (before creating the database and cleaning).