Assignment 4

Due: 17th Nov 23 03:59 pm EST

Submission:

1. Github Repo Link
2. A 10 video recording of the demo posted on an Online platform and link added to Readme

Additional notes:

1. Required attestation and contribution declaration on the GitHub page:
   WE ATTEST THAT WE HAVEN'T USED ANY OTHER STUDENTS' WORK IN OUR ASSIGNMENT AND ABIDE BY THE POLICIES LISTED IN THE STUDENT HANDBOOK
   member1: 33.3%
   member2: 33.3%
   member3: 33.3%
2. Make sure you do not push anything to your GitHub after submission date (Editing Readme.md is ok but no code pushing after deadline)
3. Create a Codelab document describing everything you did.In your GitHub you should have a readme.md files which would tell what all things are there in this GitHub repository.
4. Keep your repository private until the deadlines. Incase of any plagiarism cases both the teams which be equally held responsible.

Instructions:

Part A:

1. Reproduce the steps[1] outlined in the "*Data Engineering Pipelines with Snowpark Python*"[2] as demonstrated in the class.
2. This task is to be completed individually. Include the name and link of the forked repository in the Readme file for reference.

Part B:

3. Work as a team and each team member selects a dataset from the Snowflake marketplace[3], which offers approximately 700+ free datasets for selection.
4. Brainstorm and develop a thematic story around the chosen datasets, defining a use case for its implementation. This narrative should aim to address a specific problem statement, supported by a Proof of Concept (POC).

---

[1] https://quickstarts.snowflake.com/guide/data_engineering_pipelines_with_snowpark_python/index.html?index=..%2F..index#0

[2] https://github.com/Snowflake-Labs/sfguide-data-engineering-with-snowpark-python

[3] https://app.snowflake.com/marketplace/

5. Construct an architectural diagram detailing the steps to achieve the project's objectives using Diagrams
6. Each workflow should encompass a minimum of three SQL processes and three Procedure/User Defined Functions. **NOTE: The queries should leverage 2 or all three datasets.**
7. Utilize Git actions for deployment purposes (CI/CD) as shown in demo
8. Provide clear instructions for the teardown process after completion.
9. Ensure there are no duplications in the selected datasets.

Streamlit:

10. Establish the connection of Snowflake to facilitate the creation of analytics based on the processed data.
11. Develop a text-based SQL query feature capable of interpreting user natural language input and build out equivalent SQL queries. Execute these queries to display the corresponding actual records.
12. To aid in the creation of SQL queries using an OpenAI service, retrieve the table schema and append it to the prompt as applicable. Use Langchain for this. See this example for ideas.
13. You should present the generated raw sql code to the end user. The user should be able to update the code or provide feedback for the OpenAI api through a modified prompt.
14. The query only after vetting should be executed.

Testing:

15. Create at least 3 test cases for each of the 6 use cases. One test case should generate sql code without any issues. Second test case should by default generate an inaccurate query. But with query prompt modification should be able to generate accurate queries. The third should fail repeatedly despite prompting. Comment on each test cases and your analysis on why failures occurred.

Deployment:

16. Application to be deployed to a public cloud platform providing public access. GCP is a good place to start since it has $300 credits

References: