10 Academy Batch 4 - Weekly Challenge: Week 11

Data Engineering: Data warehouse tech stack with MySQL, DBT, Airflow, and Spark

Overview

Business Need

You and your colleagues have joined to create an AI startup that deploys sensors to businesses, collects data from all activities in a business - from people's interaction to the smart appliances installed in the company to reading environmental and other relevant information. Your startup is responsible to install all the required sensors, receive a stream of data from all sensors, and analyse the data to provide key insights to the business. The objective of your contract with the client is to reduce the cost of running the client facility as well as to increase the livability and productivity of workers.

In this challenge you are tasked to create a scalable data warehouse tech-stack that will help you provide the AI service to the client.

By the end of this project, you should produce a tool that can be used as a basis for the data warehouse needs of your startup.

Data

In Data (ucdavis.edu) you can find example parquet data, and or sensor data in CSV formats with size ~1.5Gb uncompressed each.

Expected Outcomes

Skills:

- Create and maintain Airflow DAGs
- Work with Apache Airflow, dbt, redash and a DWH
- Apply ELT techniques to DWH
- Build data pipelines and orchestration workflows

Knowledge:

- Enterprise-grade data engineering - using Apache and Databricks tools

Competency Mapping

The tasks you will carry out in this week's challenge will contribute differently to the 17 competencies 10 Academy identified as essential for job preparedness in the field of data science, and Machine Learning engineering. The mapping below shows the change (lift) one can obtain through delivering the highest performance in these tasks.

MC0: Marginal contribution - causes no significant change

MC1: Minor contribution - recognized for routine performance gain

MC2: Measurable contribution - will contribute a value towards portfolio and job readiness metric

MC3: Major contribution - the best performance of these types of tasks at least three times within our training leads one to attain a job-ready level along that competency dimension.

Competency	**Value**	**Potential contributions from this week**
Business Understanding	MC3	Understanding and reasoning the business context. Thinking about suitable analysis that matches the business need. Thinking about clients and their interests.
Data Engineering	MC3	Thinking about how to store data for easy analysis, and what format to use to build responsive dashboards.
Data Understanding	MC3	Understanding the data provided and extract insight. Exploring different techniques, algorithms, statistical distributions, sampling, and visualization techniques to gain insight.
Dashboard & Visualization	MC3	Building a dashboard to explore data as well as to communicate insight. Advanced use of modules such as plotly, seaborn, matplotlib etc. to build descriptive visualizations. Reading

		through the modules documentation to expand your skill set.
Mathematics and Statistics	MC0	Thinking about statistical distributions, sampling, bias, overfitting, correlations.
MLOps & Continuous Delivery	MC2	Using Github for code development, thinking about feature store, planning analysis pipeline, using MLOps tools for code, data, model, artifact versioning, setting up docker containers for automated microservice deployment.
Modeling and evaluation	MC0	Comparing multiple Deep learning techniques; training and validating DL models; choosing appropriate architecture, loss function, and regularisers; hyperparameter tuning; choosing suitable evaluation metrics.
Python programming	MC3	Advanced object-oriented python programming. Python package building.
SQL programming	MC3	Building feature stores using SQL or NoSQL databases.
Fluency in the Scientific Method	MC1	Thinking about evidence. Generating hypothesis, testing hypothesis.

		Thinking about different types of errors.
Ethics	MC1	data privacy, data security, ethical use of data. The 8 principles of responsible machine learning
Statistical & Critical Thinking	MC1	Thinking about the difference between causal vs chance correlation. Giving reasonable recommendations. Thinking about uncertainties.
Software Engineering & Dev Environment	MC3	Reading articles on software project planning. Unit testing.
Impact & Lifelong learning	MC3	Learning new concepts, ideas, and skills fast, and applying them to the problem at hand.
Professional Culture & Communication	MC2	Writing a well-formatted presentation with no mistakes, formatted nicely.
Social Intelligence & Mentorship	MC2	Asking for help early, providing help for those who need it, avoiding being stuck.
Career Thinking	MC1	Working within groups in a successful way

Team

Instructors: Yabebal, Abubakar, Mahlet, Kevin

Key Dates

- **Discussion on the case** - 11:30 UTC time on Monday 20 September 2021. Use #all-weeks11 to ask questions.
- **Interim Submission** - 8:00 PM UTC time on Wednesday 22 September 2021.
- **Final Submission** - 8:00 PM UTC time on Saturday 25 September 2021

Leaderboard for the week

There are 100 points available for the week.

Badges

Each week, one user will be awarded one of the badges below for the best performance in the category below.

In addition to being the badge holder for that badge, each badge winner will get +20 points to the overall score.

Visualization - the quality of visualizations, understandability, skimmability, choice of visualization

Quality of code - reliability, maintainability, efficiency, commenting - in the future this will be CICD

An innovative approach to analysis -using latest algorithms, adding in research paper content and other innovative approaches

Writing and presentation - clarity of written outputs, clarity of slides, overall production value

Most supportive in the community - helping others, adding links, tutoring those struggling

The goal of this approach is to support and reward expertise in different parts of the Machine learning engineering toolbox.

Late Submission Policy

Our goal is to prepare successful learners for the work and submitting late when given enough notice, shouldn't be necessary.

For interim submissions, those submitted 1-6 hours late will receive a maximum of 50% of the total possible grade. Those submitted >6 hours late may receive feedback, but will not receive a grade.

For final submissions, those submitted 1-24 hours late, will receive a maximum of 50% of the total possible grade. Those submitted >24 hours late may receive feedback, but will not receive a grade.

Instructions

The fundamental tasks in this week's challenge are the following - building data warehouse techstack

- Consisting of
    - A "data warehouse" (mysql, sqlite)
    - An orchestration service (Airflow)
    - An ELT tool (dbt)
    - A reporting environment (redash)
- Set it up locally using
    - fully dockerized

Complete the following tasks:

1. Create a DAG in Airflow that uses the bash/python operator to load the data files into your database. Think about a useful separation of Prod, Dev and Staging
2. Connect dbt with your DWH and write transformations codes for the data you can execute via the Bash or Python operator in Airflow. Write proper documentation for your data models and access the dbt docs UI for presentation.
3. Check additional modules of dbt that can support you with data quality monitoring (e.g. great_expectations, dbt_expectations or re-data).
4. Connect the reporting environment and create a dashboard out of this data
5. Write a short article about your approach and what were the most important decisions along the way

Consider the following elements when doing the above tasks

- AIRFLOW:
    - If you want to use templates in Airflow, what is a good way to manage metadata and variables within your DAGS? (read about context)
    - Automated Alerting - what happens if the DAG is failing, e.g. a slack or email alert
    - Built hard circuit breaker pipelines with dbt (e.g. if a test fails, do not update the production tables)

- dbt
    - Automate the generation of dbt docs and make it available via web frontend
    - Explore macros and write your own to create dynamic documentation and functions
    - Automate the dbt to Airflow connection by automatically creating DAGS out of dbt metadata (see here: https://www.astronomer.io/blog/airflow-dbt-2)
- Redash
    - Built a version control script system by hitting the API, download the queries and built an automated git storage process

Submission

Interim: Due Wednesday 22 Sep 20:00 UTC

1. Link to your code in GitHub
2. Submit a two pages max document that shows the tech-stack flow diagram, and explanation of the different elements
3. Screenshot of the data lineage from dbt

Final Due Saturday 25 Sep 20:00 UTC

1. Link to your code in GitHub
2. Link to your deployed dbt data warehouse documentation
3. Screenshot of the data view you built
4. A blog (or report of not more than 5 pages) explaining the process you followed to build the tech stack. What are the challenges? What can be improved with more time?

Feedback

You will receive comments/feedback in addition to a grade.

References

dbt:

General Information:

1. Installing dbt https://docs.getdbt.com/dbt-cli/installation/#pip
2. Introduction Videos on dbt https://www.youtube.com/playlist?list=PLy4OcwImJzBLJzLYxpxaPUmCWp8j1esvT
3. Redshift config; https://docs.getdbt.com/reference/resource-configs/redshift-configs/
4. Docs from Gitlab: https://about.gitlab.com/handbook/business-ops/data-team/platform/dbt-guide/
5. CLI command reference: https://docs.getdbt.com/reference/dbt-commands/
6. Basic Introduction to dbt https://www.kdnuggets.com/2021/07/dbt-data-transformation-tutorial.html

Articles:

1. https://medium.com/the-telegraph-engineering/dbt-a-new-way-to-handle-data-transformation-at-the-telegraph-868ce3964eb4
2. https://medium.com/hashmapinc/dont-do-analytics-engineering-in-snowflake-until-you-read-this-hint-dbt-bdd527fa1795

Repo Examples:

1. https://github.com/mattermost/mattermost-data-warehouse
2. https://gitlab.com/gitlab-data/analytics/-/tree/master/

How to structure repo's

1. https://discourse.getdbt.com/t/how-we-structure-our-dbt-projects/355
2. https://discourse.getdbt.com/t/should-i-have-an-organisation-wide-project-a-monorepo-or-should-each-work-flow-have-their-own/666/2
3. https://discourse.getdbt.com/t/how-to-configure-your-dbt-repository-one-or-many/2121

4. https://medium.com/photobox-technology-product-and-design/practical-tips-to-get-the-best-out-of-data-building-tool-dbt-part-1-8cfa21ef97c5

Airflow:

1.https://livebook.manning.com/book/data-pipelines-with-apache-airflow/chapter-1/v-6

2. https://www.linkedin.com/in/marclamberti/ :)

docker:

1. https://www.youtube.com/watch?v=fqMOX6JJhGo

2. https://docker-curriculum.com/#docker-images

Redash:

1. https://github.com/dwyl/learn-redash
2. https://fitdevops.in/how-to-setup-redash-dashboard-on-ubuntu

Superset:

1. https://www.startdataengineering.com/post/apache-superset-tutorial/
2. https://superset.apache.org/docs/creating-charts-dashboards/first-dashboard
3. https://superset.apache.org/docs/installation/installing-superset-from-scratch

Virtual environments:

1. https://www.ianmaddaus.com/post/manage-multiple-versions-python-mac/
2. https://www.codeblocq.com/2016/01/Search-through-history-in-OSX-terminal/
3. https://janakiev.com/blog/jupyter-virtual-envs/
4. https://medium.com/@blessedmarcel1/how-to-install-jupyter-notebook-on-mac-using-homebrew-528c39fd530f