dataroots

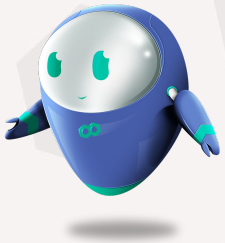
Snowflake + ML

Using Snowflake as a full end-to-end solution

15/08/2022

dataroots

Agenda for today



1. ML Engineering ⚙️
2. Snowflake recap ❄️
3. Snowpark and Python 🐍
4. Snowflake and ML 🤖
5. Demo 🤩
   a. App
6. Solution architecture 📐
7. Demo 🎬
   a. Streams
   b. Tasks
   c. UDFs
   d. Stages
8. CICD and stored procedures 📝
9. Model training in Snowflake 🦾
10. Thoughts and takeaways 💭

dataroots

About me

- ✉️ murilo@dataroots.io
- 🇧🇷 Brazilian
- 🤓 B.Sc. in Mechanical Engineering @PNW
- 👨‍🎓 M.Sc. in Artificial Intelligence @KUL
- ☁️ GCP - (Data &) ML Engineer
- ☁️ AWS - Machine Learning
- ☁️ Hashicorp - Terraform
- ☁️ Astronomer - DAG Authoring & Airflow
- ⚽ Co-captain & coach @datafoots
- 🙋‍♂️ Coach & tech lead @ AI Unit
- 🤖 MLE @dataroots

dataroots

ML Engineering

dataroots

Snowflake recap

dataroots

Snowflake in 🥜

- SQL warehouse
  - Compute
  - Storage
  - (Access management, ...)
- Highlights 🤩
  - UDFs
  - Stored procedures

From Snowflake Summit (June, 2022)

dataroots

Snowpark

dataroots

Snowpark + ML

snowflake WHY SNOWFLAKE THE DATA CLOUD SOLUTIONS RESOURCES COMPANY

← More Trending Articles

MACHINE LEARNING TOOLS

Machine learning, a subset of AI that enables computers to learn from data and improve through experience, is a booming segment. Spending around machine learning is expected to grow from about $1.58 billion in 2017 to $20.8 billion in 2023, according to a recent report.

A variety of sectors, including banking, government, health care, life sciences, retail and telecom, are increasing spending around ML enterprises.

Meanwhile, a shortage of qualified AI practitioners inhibits even greater progress for many businesses.

With the right machine learning tools, however, data scientists are empowered to optimize ML endeavors.

WHY SNOWFLAKE THE DATA CLOUD SOLUTIONS RESOURCES COMPANY CONTACT START FOR FREE

SNOWFLAKE FOR DATA SCIENCE & ML

Accelerate your ML workflows with fast data access and elastically scalable data processing for Python and SQL.

EXPLAINER VIDEO

RAPIDLY DELIVER EXPERIMENTS INTO PRODUCTION

Watch Now ›

PUBLIC PREVIEW ANNOUNCEMENT

RUN SQL AND PYTHON SIDE TO SIDE WITH SNOWPARK FOR PYTHON

Read Now ›

Community Resources Blog

Ask a question...

DOCS » CONNECTING TO SNOWFLAKE » SNOWFLAKE ECOSYSTEM » MACHINE LEARNING & DATA SCIENCE

PREVIOUS | NEXT

- Connecting to Snowflake
- Snowflake Ecosystem
  - All Partners & Technologies (Alphabetical)
  - Data Integration
  - Business Intelligence (BI)
  - Machine Learning & Data Science
  - Security, Governance & Observability
  - SQL Development & Management
  - Native Programmatic Interfaces
- Snowflake Partner Connect
- General Configuration (All Clients)
- SnowSQL (CLI Client)

Machine Learning & Data Science

Also referred to as advanced analytics, artificial intelligence (AI), and "Big Data", machine learning and data science cover a broad category of vendors, tools, and technologies that provide advanced capabilities for statistical and predictive modeling.

These tools and technologies often share some overlapping features and functionality with BI tools; however, they focus less on analyzing/reporting past data. Instead, they focus on examining large data sets to discover patterns and uncover useful business information that can be used to predict future trends.

The following machine learning and data science platforms and technologies are known to provide native connectivity to Snowflake:

Solution	Version / Installation Requirements	Notes
alteryx	**Alteryx:** Analytics 11.5 (or higher) **Snowflake:** ODBC Driver — download from the Snowflake Client Repository	• Available for trial via Snowflake Partner Connect. • Validated by the Snowflake Ready Technology Validation Program. • Additional resources: ◦ Snowflake In-Database Functionality Now Available (Alteryx Community Blog) ◦ Supported Data Sources — Snowflake (Alteryx Documentation)

dataroots

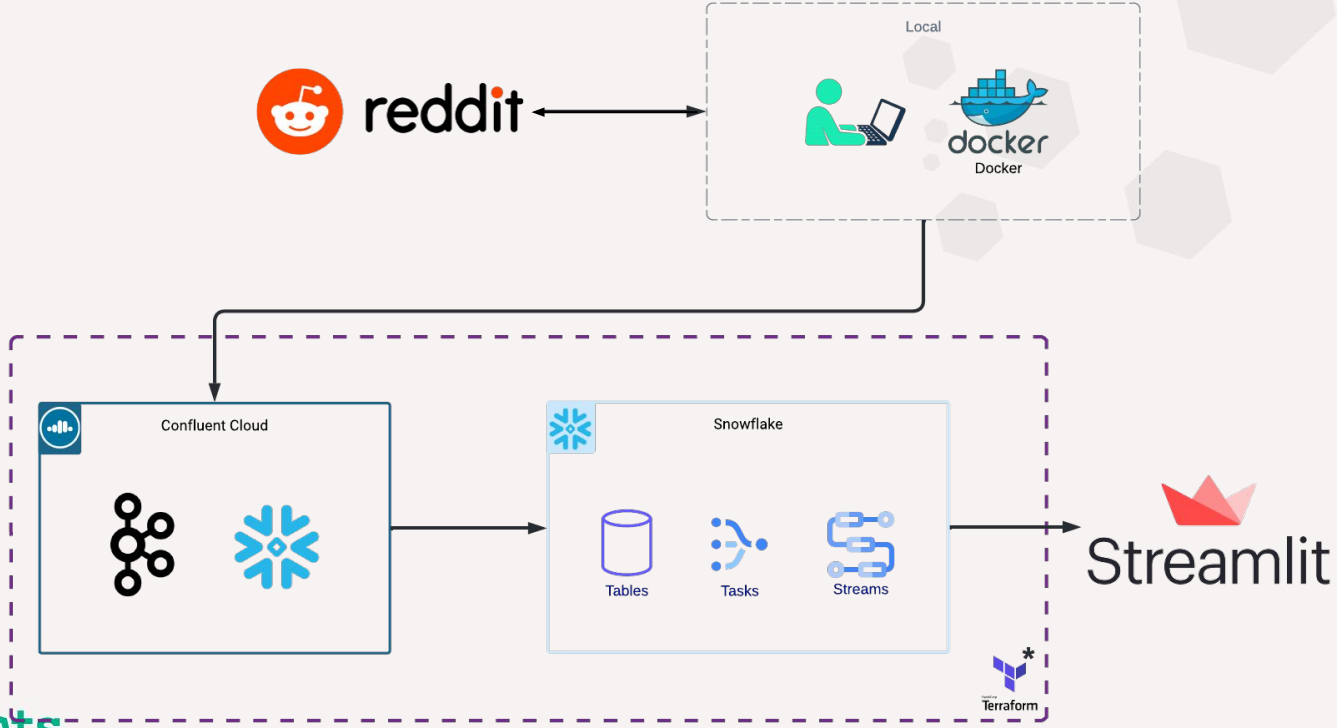
Snowpark + ML

dataroots

Demo - App

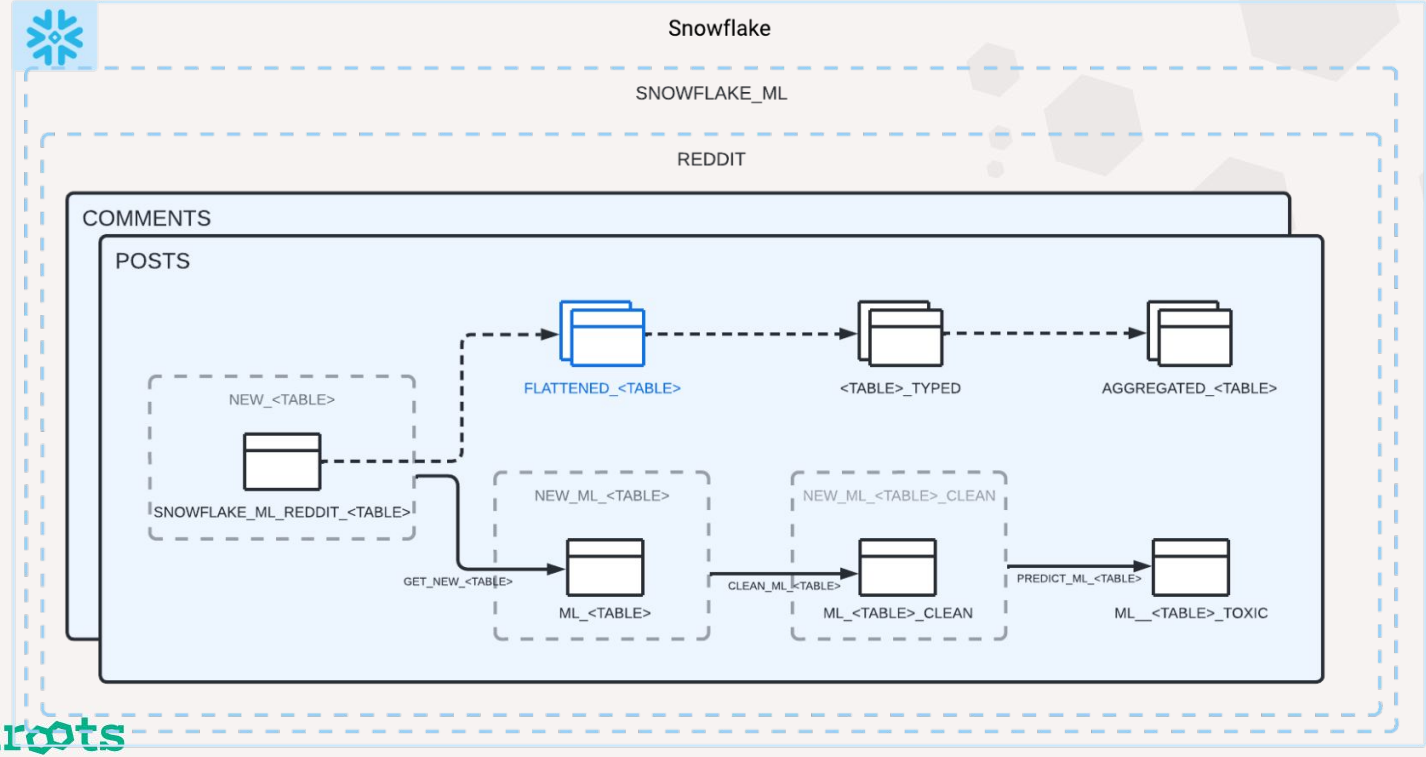
Architecture

ML solution architecture



dataroots

ML solution architecture: Snowflake



dataroots

Demo: streams, tasks, UDFs, stages

Demo: streams, tasks, UDFs, stages

```
-- SETUP
USE DATABASE SNOWFLAKE_ML;
USE SCHEMA REDDIT;
USE ROLE ACCOUNTADMIN;
USE WAREHOUSE "reddit_xs";
-- TABLE
CREATE TABLE FOO IF NOT EXISTS (something VARCHAR);
INSERT INTO FOO VALUES('BAR');
SELECT * FROM FOO;
-- STREAM
CREATE STREAM IF NOT EXISTS NEW_FOO ON TABLE FOO;
INSERT INTO FOO VALUES('BAZ');
SELECT * FROM FOO;
SELECT * FROM NEW_FOO;
-- TASKS
-- (Show defs)
-- UDFS
SHOW USER FUNCTIONS;
SELECT SOMETHING, ML_PREDICT_DEV(SOMETHING)  FROM FOO;
SELECT SOMETHING, ML_PREDICT_DEV(SOMETHING)  FROM NEW_FOO;
SELECT SOMETHING, ML_PREDICT_PROD(SOMETHING)  FROM NEW_FOO;
DESCRIBE FUNCTION ML_PREDICT_DEV(VARCHAR);
DESCRIBE FUNCTION ML_PREDICT_PROD(VARCHAR);
-- STAGES
SHOW STAGES;
LIST @PY_UDFS;
LIST @PY_UDFS/0.3;
LIST @PY_UDFS/0.4;
-- CLEANUP
DROP TABLE FOO;
DROP STREAM NEW_FOO;
```

dataroots

CICD and Stored procedures

CI/CD workflow

Changes requested (CI) 🙏

1. Pull request
   a. Build model package release candidate
   b. Create/update development *UDF*
   c. Create a *temporary stored procedure* for model evaluation
   d. Call stored procedure
   e. Store artifacts in Snowflake
   f. Publish evaluation results in PR
   g. Tag release candidate
2. 👍 or 👎

Changes approved (CD) ✅

1. Build model package with new minor version
2. Create update production UDF
3. Tasks will use latest UDF version
4. Create a new git tag

dataroots

Model training in Snowflake

Development workflow

1. Code changes
   a. Code
   b. ML
2. Create (temporary) stored procedure
   a. Saves artifacts in a stage
3. Call stored procedure on all the data
4. Change reference on inference function
   a. Pull artifacts from stage
   b. Load model
   c. Call model for predictions

dataroots

Thought and takeaways

Beefs

- Terraform/Snowpark is still buggy
  - Roles
  - Materialized views
  - State doesn't converge
  - Hard setup
  - Not-so-helpful error messages
  - Lowercase vs uppercase
- UDFs can only be single function
  - Need to hack things up to get local imports
- Python imports from conda only
  - Need to vendorize otherwise
- Not-so-great documentation
  - Hard to find info
- Stage files kinda tricky?
  - Path vs file names
- Tasks are still in early stages
  - No retries
  - No errors when tasks fails
  - No easy view on tasks statuses
- Unsure about training models efficiency
  - Stored procedures - only on one machine?
  - How is it different from running locally?
  - No accelerated (GPU) compute (yet)
- Still hard to use Snowpark
  - Loading models from hub not straightforward
  - Runs out of memory
  - `transformers` is available however
  - Hard to debug

dataroots

Delights

- Really powerful
  - 10x developer
  - SQL finishes
  - Rich ecosystem
- Tons of flexibility
  - Python, Java, Javascript
- Main functionalities available
  - Compute
  - Inference
  - Workflows (tasks and streams)
- IaC available
  - Have my beefs, but gotta start somewhere right?
- These were all preview features
  - “Use at your own risk”
  - Tendency to be more robust with time
- Structured and unstructured
  - Simple to use
- Streams are super easy to use

**I probably suffer from negativity bias 😅

dataroots

Bottom line

- Fun project
- Many bugs/lacking docs
- Not ready (yet) for critical apps
- Only simple applications

- Opens tons of new possibilities
- Early stages
  - Maybe in 1-2 years it'll be ready for full applications
- Contribute!
  - Terraform
  - MLFlow
  - DVC

We miss entirely:

- Experiment tracking

dataroots

dataroots