



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

Module 2: Rapid  
Prototyping with  
Streamlit on Snowflake



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

## Building Prototypes on Snowflake

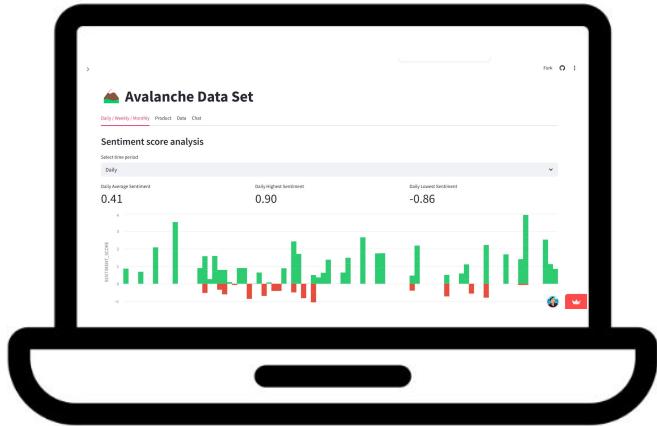
# MVP Build Plan

## Avalanche Dashboard App



1. Load a dataset with pandas
2. Analyze the reviews sentiments
3. Display the dataset
4. Visualize the sentiment analysis

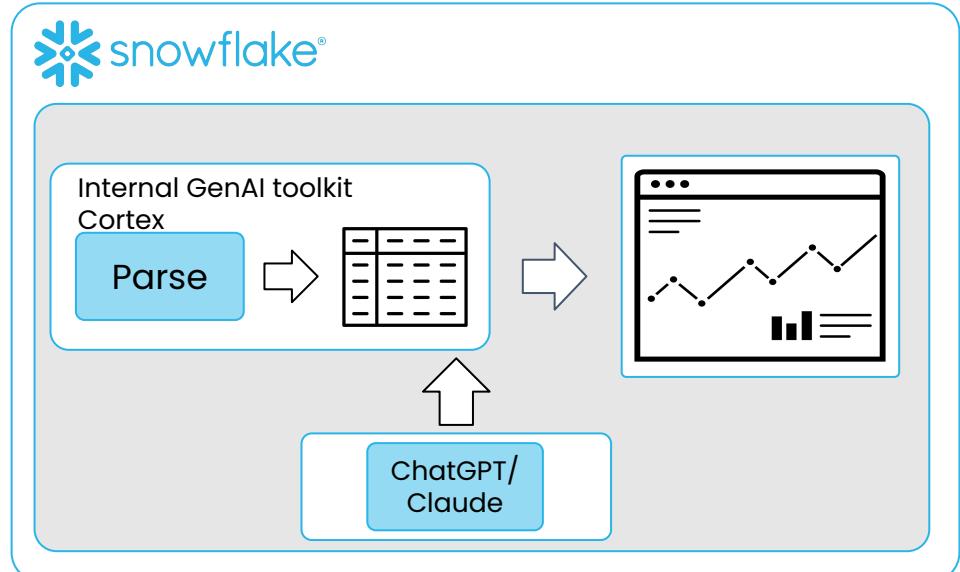
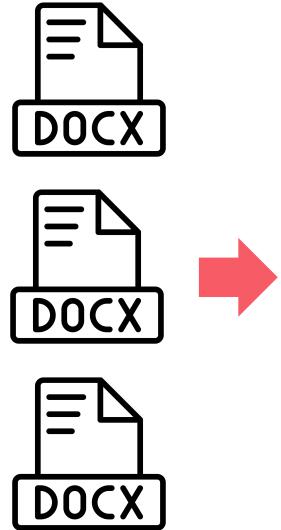
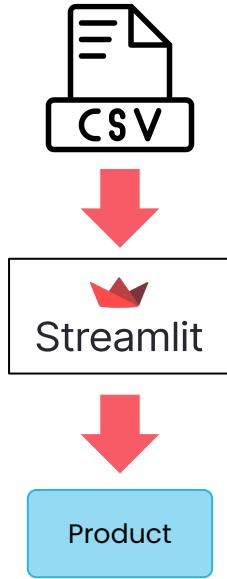
# What You've Built So Far



- ✓ Ingested a .csv file of customer reviews
- ✓ Performed cleanup and sentiment analysis on the data
- ✓ Visualized the results
- ✓ Used Streamlit to create a quick user interface accessible in your web browser

# Streamlit vs Snowflake Workflows

Avalanche  
customer\_reviews.csv





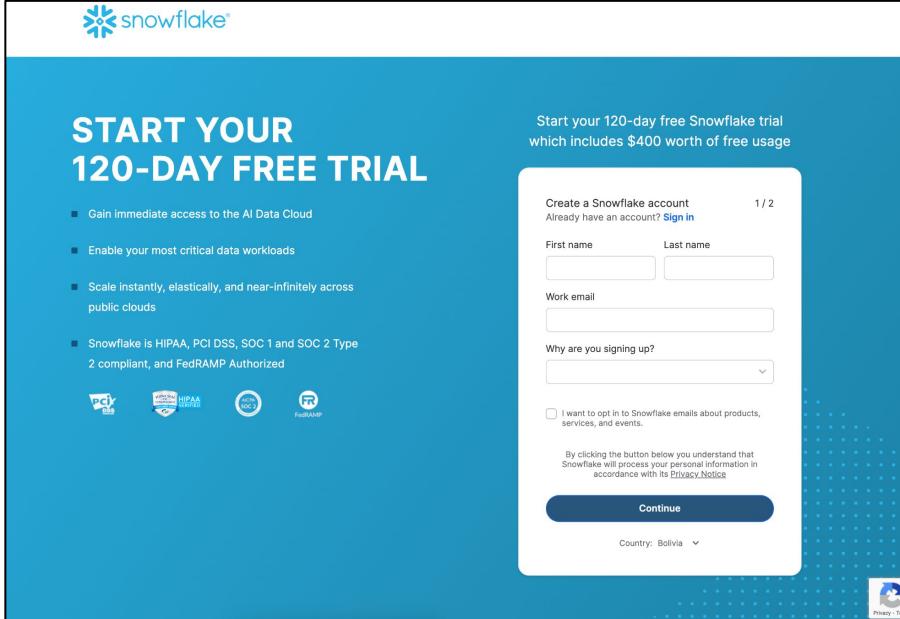
DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

## Introducing Snowflake

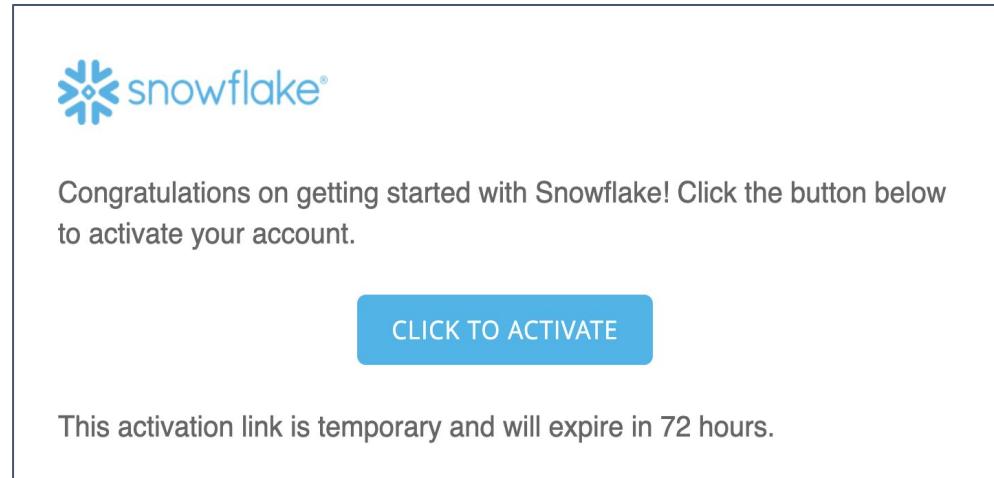
# Setting Up Your Snowflake Trial Account



bit.ly/snowflake-dlai-trial

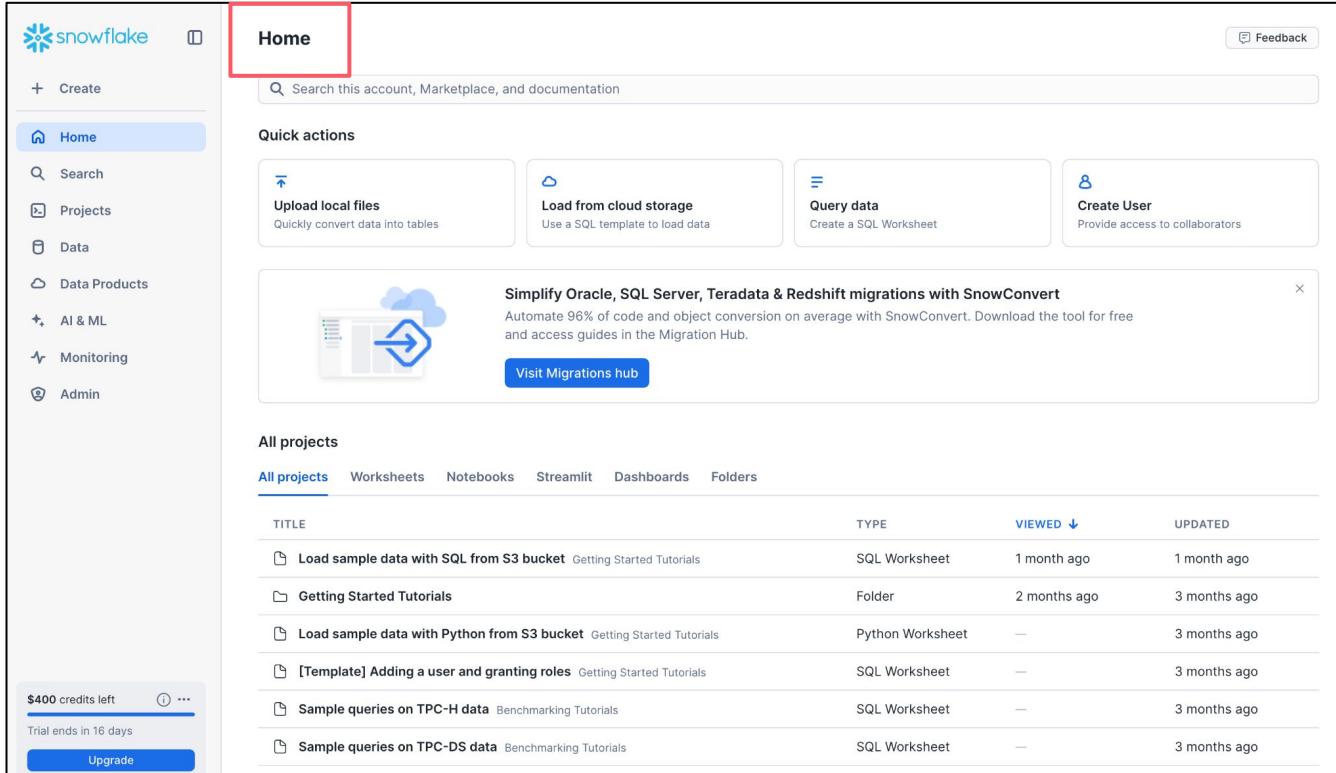
# Final Step

-  Check your inbox for a verification email
-  Click the verification link
-  You'll be able to log in and access the platform



*Don't see the email?  
Check your **Spam** or **Promotions** folder!*

# Snowflake Dev Environment Overview

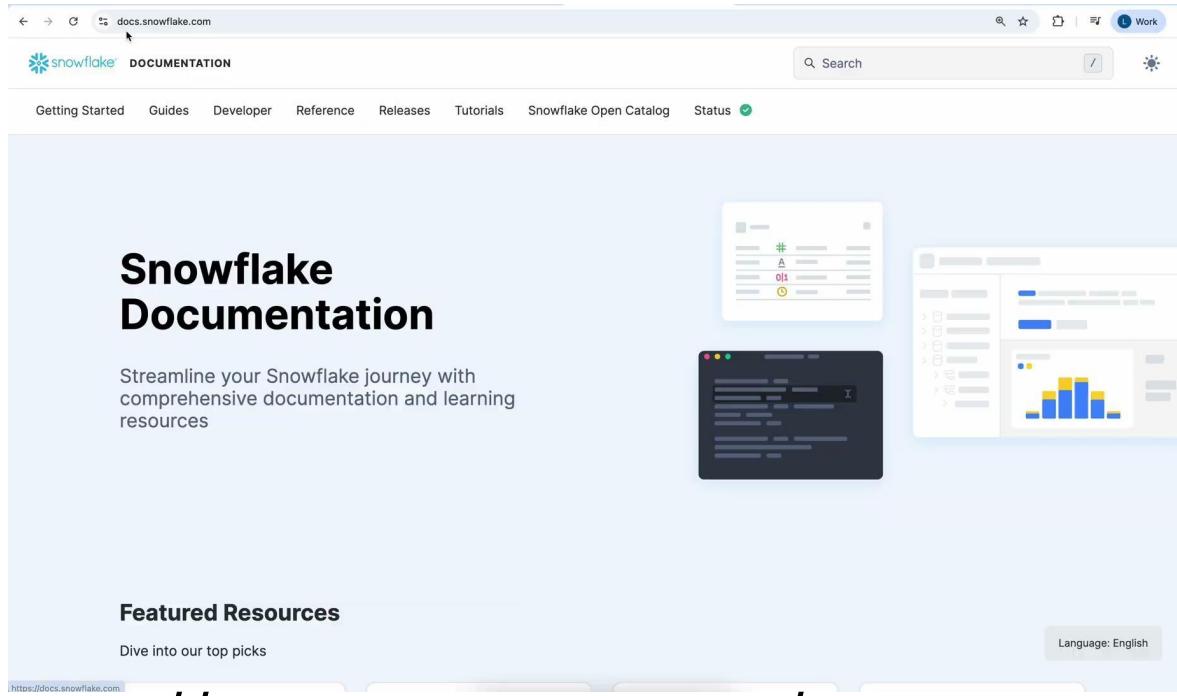


The screenshot shows the Snowflake Home page. On the left is a sidebar with navigation links: + Create, Home (which is selected and highlighted in blue), Search, Projects, Data, Data Products, AI & ML, Monitoring, and Admin. A message at the bottom of the sidebar says '\$400 credits left' and 'Trial ends in 16 days'. A blue 'Upgrade' button is also present. The main content area has a 'Home' button with a red border. Below it is a search bar with placeholder text 'Search this account, Marketplace, and documentation'. Under 'Quick actions', there are four cards: 'Upload local files', 'Load from cloud storage', 'Query data', and 'Create User'. A promotional banner for 'SnowConvert' is displayed, showing an icon of a database and clouds, with text about simplifying migrations from Oracle, SQL Server, Teradata, and Redshift. Below the banner is a section titled 'All projects' with tabs for All projects (selected), Worksheets, Notebooks, Streamlit, Dashboards, and Folders. A table lists several project entries, including 'Load sample data with SQL from S3 bucket', 'Getting Started Tutorials', 'Load sample data with Python from S3 bucket', '[Template] Adding a user and granting roles', 'Sample queries on TPC-H data', and 'Sample queries on TPC-DS data'. The table has columns for TITLE, TYPE, VIEWED (sorted by descending date), and UPDATED.

| TITLE   | TYPE             | VIEWED ↓     | UPDATED      |
|---|------------------|--------------|--------------|
| Load sample data with SQL from S3 bucket Getting Started Tutorials    | SQL Worksheet    | 1 month ago  | 1 month ago  |
| Getting Started Tutorials   | Folder           | 2 months ago | 3 months ago |
| Load sample data with Python from S3 bucket Getting Started Tutorials | Python Worksheet | —            | 3 months ago |
| [Template] Adding a user and granting roles Getting Started Tutorials | SQL Worksheet    | —            | 3 months ago |
| Sample queries on TPC-H data Benchmarking Tutorials                   | SQL Worksheet    | —            | 3 months ago |
| Sample queries on TPC-DS data Benchmarking Tutorials                  | SQL Worksheet    | —            | 3 months ago |

← Snowsight

# Snowflake Dev Environment Overview



The screenshot shows the homepage of the Snowflake Documentation website at <https://docs.snowflake.com>. The page features a large title "Snowflake Documentation" with a subtitle "Streamline your Snowflake journey with comprehensive documentation and learning resources". Below this, there's a section titled "Featured Resources" with a "Dive into our top picks" button. The interface includes a search bar, a navigation menu with links like "Getting Started", "Guides", "Developer", etc., and a status indicator. Three icons representing different tools or features are displayed on the right: a data grid, a terminal window with code, and a chart.

<https://docs.snowflake.com/>

# Snowflake Dev Environment Overview

Welcome to Snowflake! How do you want to start?



Explore a sample data set

Load and query sample data from **TastyBytes**, a fictional global food truck business.

[Start](#)

OR



Load data into Snowflake

You can load data from any of the following:

- Local files
- External cloud providers
- 3rd party connectors (20+ providers)

[Start](#)

[Skip for now](#)

# Snowflake Dev Environment Overview

**Snowsight:**  
The Snowflake  
web interface

The screenshot shows the Snowflake web interface. On the left is a sidebar with a blue header containing the Snowflake logo and a 'Create' button. Below the header are links for Home, Search, Projects, Data, Data Products, AI & ML, Monitoring, and Admin. At the bottom of the sidebar is a promotional banner for 'SnowConvert' with a 'Visit Migrations hub' button. The main content area has a 'Home' header and a search bar. It features a 'Quick actions' section with four buttons: 'Upload local files', 'Load from cloud storage', 'Query data', and 'Create User'. Below this is a banner for 'Simplify Oracle, SQL Server, Teradata & Redshift migrations with SnowConvert'. The main part of the page is titled 'All projects' and lists several items under 'All projects': 'Load sample data with SQL from S3 bucket', 'Getting Started Tutorials', 'Getting Started Tutorials', 'Load sample data with Python from S3 bucket', '[Template] Adding a user and granting roles', 'Sample queries on TPC-H data', and 'Sample queries on TPC-DS data'. Each item includes a title, type (e.g., SQL Worksheet, Python Worksheet), and last viewed/updated times.

| TITLE   | TYPE             | VIEWED ↓     | UPDATED      |
|---|------------------|--------------|--------------|
| Load sample data with SQL from S3 bucket Getting Started Tutorials    | SQL Worksheet    | 1 month ago  | 1 month ago  |
| Getting Started Tutorials   | Folder           | 2 months ago | 3 months ago |
| Load sample data with Python from S3 bucket Getting Started Tutorials | Python Worksheet | —            | 3 months ago |
| [Template] Adding a user and granting roles Getting Started Tutorials | SQL Worksheet    | —            | 3 months ago |
| Sample queries on TPC-H data Benchmarking Tutorials                   | SQL Worksheet    | —            | 3 months ago |
| Sample queries on TPC-DS data Benchmarking Tutorials                  | SQL Worksheet    | —            | 3 months ago |

# Snowflake Dev Environment Overview

Command  
Line  
Interface  
(CLI)

**Downloads**

| CLI Client (snowsql) |
|----------------------|
| JDBC Driver          |
| ODBC Driver          |
| Python Components    |
| Node.js Driver       |
| Spark Connector      |

**CLI Client (snowsql)**

Download the latest version of SnowSQL for your platform.

For installation and configuration details, see the [Snowflake Documentation](#)

 [CLI Client \(snowsql\) 1.1.62 Linux](#)

 [CLI Client \(snowsql\) 1.1.62 64-bit Windows](#)

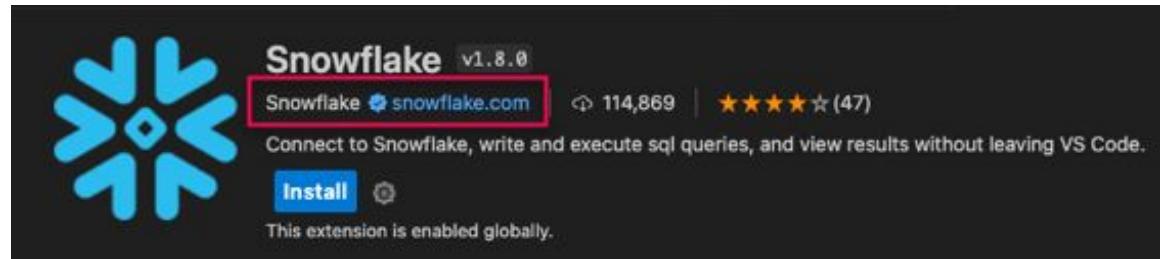
 [CLI Client \(snowsql\) 1.1.62 64-bit Mac OSX](#)

 [Snowflake GPG Public Key For Linux](#)

**Done**

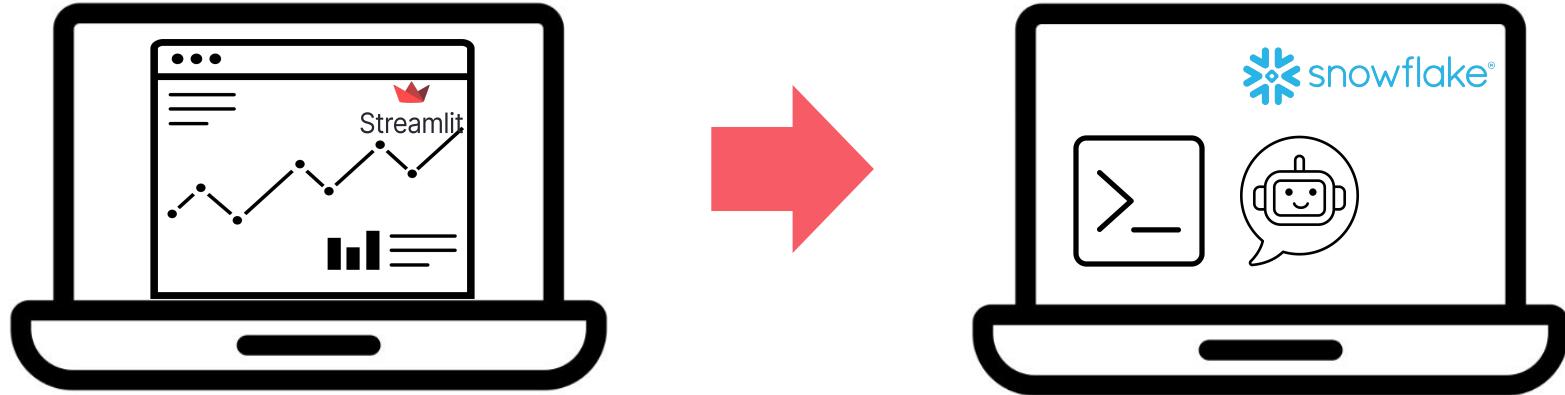
# Snowflake Dev Environment Overview

VS Code  
Extension

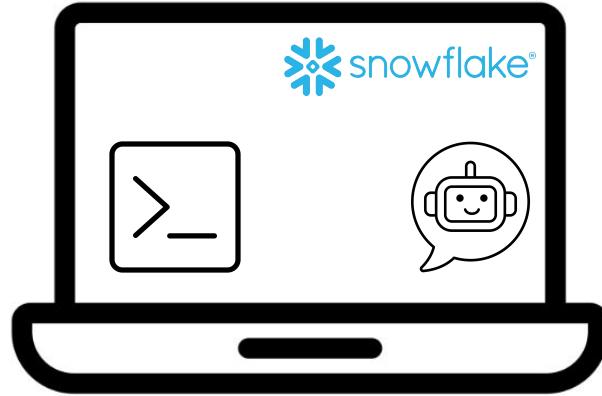


[docs.snowflake.com/en/user-guide/vscode-ext](https://docs.snowflake.com/en/user-guide/vscode-ext)

# Using Streamlit in Snowflake



# Using Streamlit in Snowflake

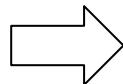


- ✓ Create dashboards directly from Snowflake tables
- ✓ Explore and visualize your data in real time
- ✓ Keep your entire workflow in one place

# Using Streamlit in Snowflake

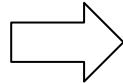


Snowflake Notebook



- Great for rapid prototyping
- Working with teammates

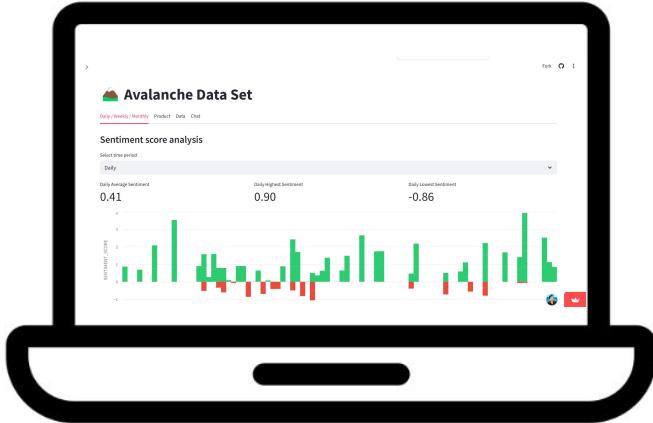
Streamlit in Snowflake



- When you're ready to share your app more broadly

# What You've Already Built

## Module 1



## Module 2

- Connecting your app to **Snowflake** tables
- Adding filters, interactive widgets, and visualizations



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

Snowsight  
Development  
Environment

# What is Snowflake?



Snowflake is your all-in-one data command center in the cloud.

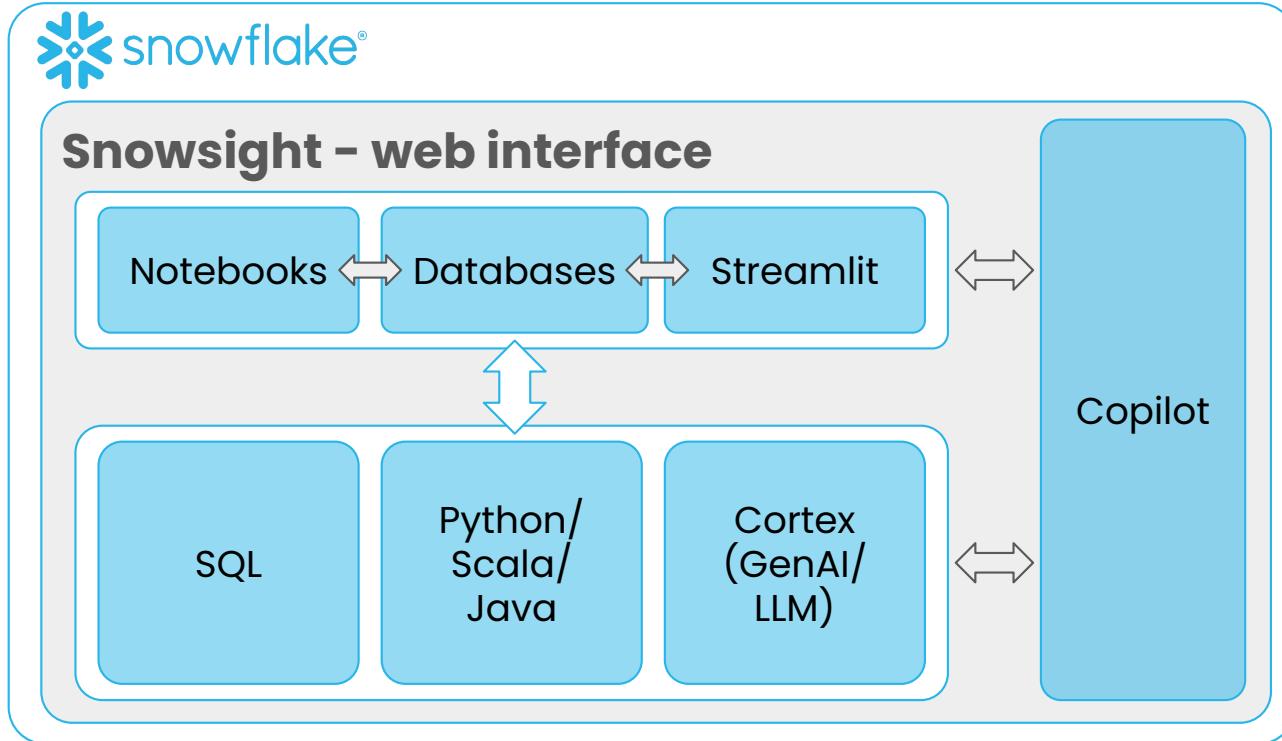
Virtual  
Warehouses



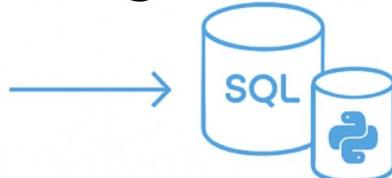
Storage Layer



# What is Snowflake?



# How It All Works Together



1 Explore data  
in Snowsight

2 Analyze with  
SQL or Snowpark

3 Build app  
with Streamlit

sentiment analysis  
and text generation



Snowflake  
Cortex

Claude or ChatGPT  
to work with python



Snowflake  
Copilot



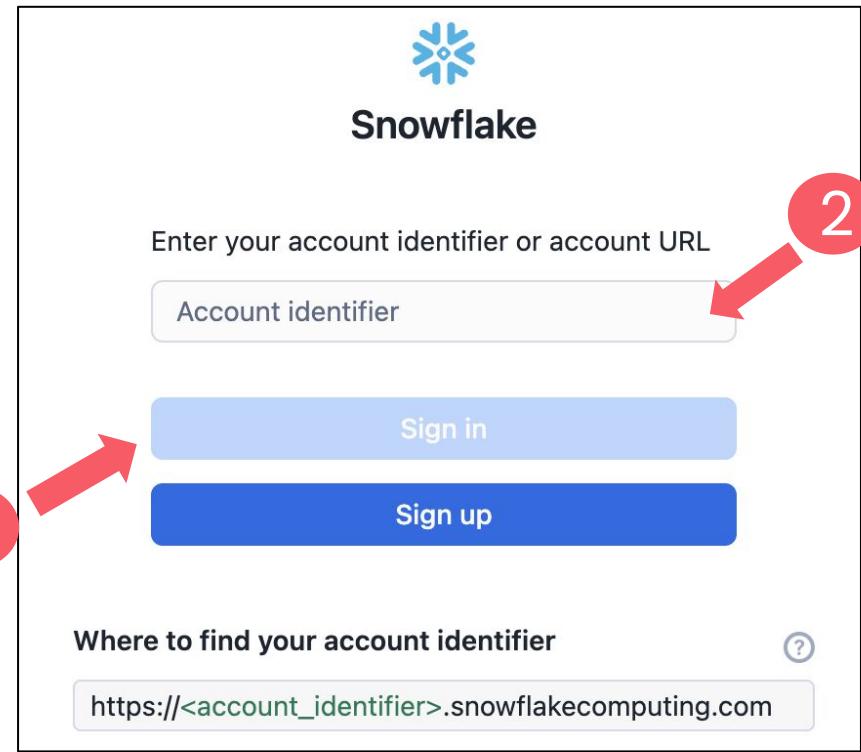
4 Deploy in Snowflake  
or on Streamlit  
Community Cloud

# Logging In to Snowsight

1

<https://app.snowflake.com>

2



# How Snowflake Organizes Your Data



## **Database**

└ Schema

(subfolders)

└ Table

(where the data actually lives)

└ View

(virtual table)

└ Stage

(file upload location)

# Wrap Up

- Logged into Snowsight 
- Explored the Avalanche database 
- Navigated schemas and tables 
- Peeked inside the customer\_reviews table 
- Checked out key tools like Worksheets, Notebooks, and Apps  




DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

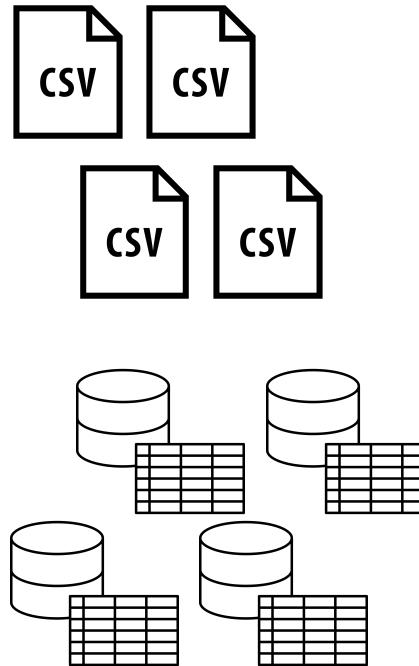
From CSV to Cloud – Using  
Notebooks to Ingest  
Avalanche Data

# MVP Build Plan



1. **Getting data into Snowflake**
2. Parsing and Structuring
3. Cleanup Data
4. Analyze Data
5. Visualize Results
6. Create interface with a tab called "Data" that displays the head of the dataset
7. Deploy to cloud

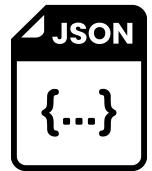
# What Kind of Files does Snowflake Support?



# What Kind of Files does Snowflake Support?



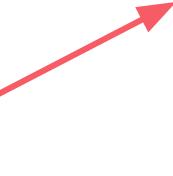
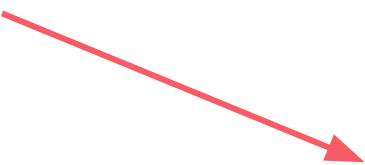
Structured



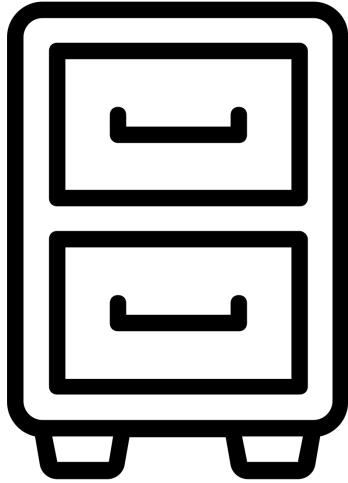
Semi-structured



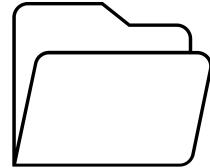
Unstructured



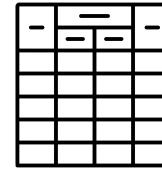
# How Snowflake Organizes Your Data



Databases



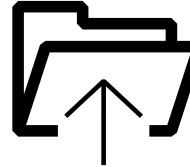
Schemas



Tables

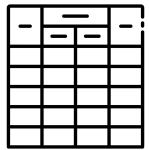


Views

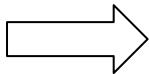


Stages

# How Snowflake Organizes Your Data



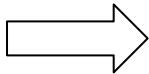
Tables



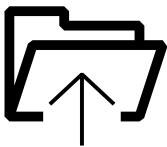
rows and columns of structured data



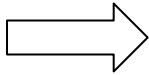
Views



saved queries that look like tables



Stages



secure zones to store raw files

# How Snowflake Organizes Your Data



Database

└ Schema

└ Table

└ View

└ Stage

(subfolders)

(holds your structured data)

(queries that look like tables)

(file upload location)

Avalanche 'customer\_reviews.csv'

# Load Your Data into DataFrames

Database

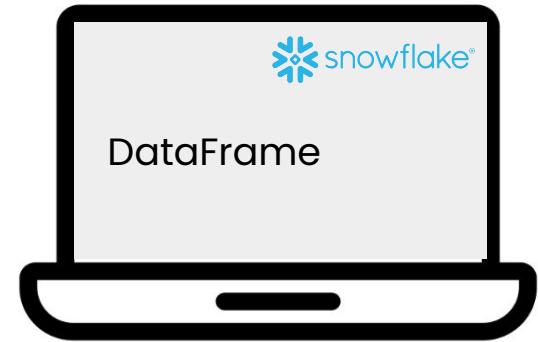
  └ Schema

    └ Table

    └ View

    └ Stage

      ➡ 'customer\_reviews.csv' ➡



- Pandas DataFrames
- Snowpark DataFrames

# Snowflake vs Pandas Dataframes

Pandas DataFrames



✓ small datasets

✗ slow down or crash with larger data

Snowpark DataFrames



✓ when you're ready



Snowflake's cloud infrastructure

# Snowflake vs Pandas Dataframes

| Feature          | Pandas Dataframes                                   | Snowpark Dataframes   |
|------------------|---|---|
| <b>Execution</b> | Eager<br>(runs immediately in local memory)         | Lazy<br>(SQL runs in Snowflake when needed)   |
| <b>Location</b>  | Resides in your local Python process                | Resides in<br>Snowflake's cloud infrastructure  |
| <b>Scale</b>     | Limited by your machine's memory                    | Handles massive datasets with ease  |
| <b>Syntax</b>    | Familiar pandas-style indexing and operations       | Similar to pandas, but uses <code>.filter()</code> ,<br><code>.select()</code> instead of direct indexing |
| <b>Use Case</b>  | Best for small to medium data on your local machine | Best for working with large, cloud-scale datasets securely  |



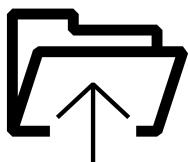
DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

Uploading a  
Batch of Files

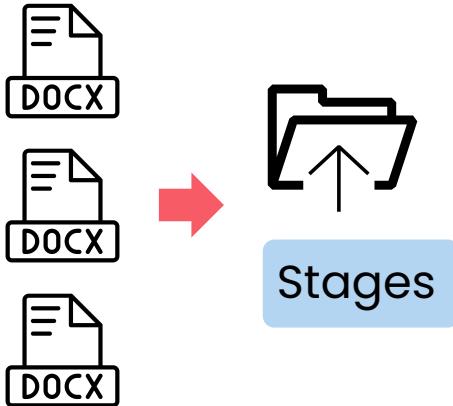
# Why Use Stages



Stages

Stages are like  
a secure  
holding zone  
for your files.

- ✓ Help you preview and validate files before ingestion
- ✓ Keep your uploads organized
- ✓ Let you reuse the same files across multiple workflows



Database            AVALANCHE\_DB  
└ Schema            AVALANCHE\_SCHEMA  
    └ **Stage**        AVALANCHE\_STAGE

/Data/customer\_reviews\_docx.zip



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

From Stage to Table  
with Cortex

# MVP Build Plan

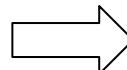


1. Getting data into Snowflake ✓
2. Parsing and Structuring
3. Cleanup Data
4. Analyze Data
5. Visualize Results
6. Create interface with a tab called "Data" that displays the head of the dataset
7. Deploy to cloud

# MVP Build Plan

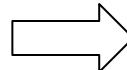
## 2. Parsing and Structuring

PARSE\_DOCUMENT



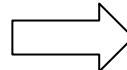
extract content from each of  
your staged Word Docs

Create a table



doc's filename and the  
extracted\_text

Preview the results in  
your Snowflake Notebook



make sure everything worked as  
planned



|          |                  |
|----------|------------------|
| Database | AVALANCHE_DB     |
| └ Schema | AVALANCHE_SCHEMA |
| └ Stage  | AVALANCHE_STAGE  |

# Using Cortex

Snowflake  
Cortex

- to process and analyze data → simple SQL
- PARSE\_DOCUMENT → extract content

Cortex handles it all for you – fast, secure, and right inside Snowflake!



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

## Extracting Information from the Content

# MVP Build Plan



1. Getting data into Snowflake ✓
2. Parsing and Structuring
3. Cleanup Data
4. Analyze Data
5. Visualize Results
6. Create interface with a tab called "Data" that displays the head of the dataset
7. Deploy to cloud



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

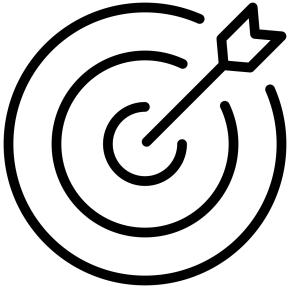
---

Lab 1 - Avalanche  
Shipping Analytics

# Your Mission



shipping-logs.csv



- Upload the file to the Avalanche stage
- Clean and validate the content
- Load it into a Snowflake table
- Prep the cleaned data for combining with the customer reviews in the next lab

# Your Files

GitHub repo:

M2

  └ Lesson\_01

    └ Lab1

      └ shipping-logs.csv

      └ M2Lab1\_solution.ipynb

Database      AVALANCHE\_DB  
          └ Schema    AVALANCHE\_SCHEMA  
                └ Stage    AVALANCHE\_STAGE

# Lesson 2

Data Analysis with Snowflake and GenAI



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

One Table to Rule  
Them All

# MVP Build Plan

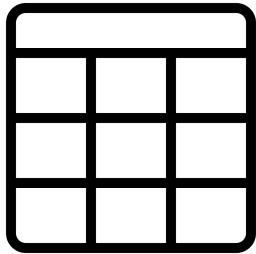


1. Getting data into Snowflake ✓
2. Parsing and Structuring ✓
3. Clean up Data
4. Analyze Data
5. Visualize Results
6. Create interface with a tab called "Data" that displays the head of the dataset
7. Deploy to cloud

# Set Up Your Environment

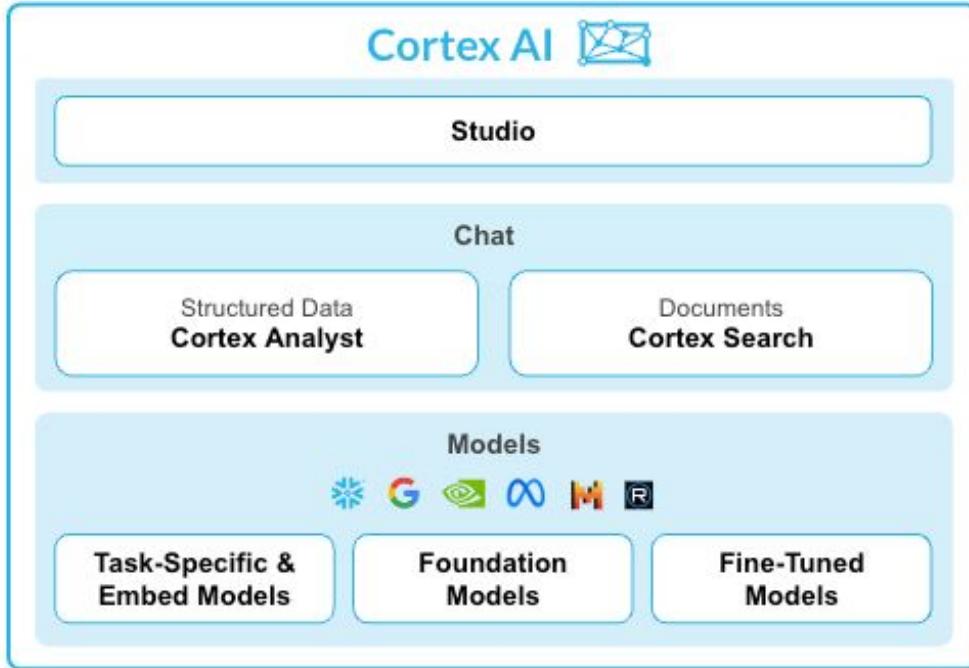
```
Database           AVALANCHE_DB
└ Schema          AVALANCHE_SCHEMA
    └ Stage        AVALANCHE_STAGE
```

# Set Up Your Environment



- customer\_reviews
- shipping\_logs

# Quick Recap: What is Snowflake Cortex?



- **Cortex Analyst:** ask questions about structured data without writing SQL  
*"What were our total sales in Q2 by region?"*
- **Cortex Search:** explore unstructured text documents  
*"What customer complaints mention late deliveries?"*

[https://quickstarts.snowflake.com/guide/getting\\_started\\_with\\_cortex\\_analyst](https://quickstarts.snowflake.com/guide/getting_started_with_cortex_analyst)

# Quick Recap: What is Snowflake Cortex?

Cortex – AI Tools



Easily generate and run SQL queries

Quickly parse documents and extract text

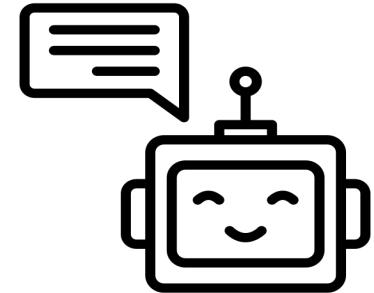
Summarize and classify content

Analyze data using LLM

# Use GenAI As Your Coding Partner

Write SQL → Snowflake Notebook

Ask your GenAI app to refresh your memory on what those tables contain!



# Use GenAI As Your Coding Partner

① Open a Snowflake Notebook

② Write Snowflake SQL



Ask your GenAI app to help



```
DESC TABLE CUSTOMER_REVIEWS;  
DESC TABLE SHIPPING_LOGS;
```

>\_

*Write Snowflake SQL  
to show me a  
preview of two tables  
named  
customer\_reviews  
and shipping\_logs.*

# Use GenAI As Your Coding Partner

>\_

*"Write Snowflake SQL to join two tables—customer\_reviews and shipping\_logs—on order\_id. Include review text, shipping date, carrier, and status."*



```
CREATE OR REPLACE TABLE
merged_reviews AS
SELECT
    r.order_id,
    r.review_text,
    s.carrier,
    s.shipping_date,
    s.status
FROM
    customer_reviews r
JOIN
    shipping_logs s
ON
    r.order_id = s.order_id;
```

# Use GenAI As Your Coding Partner

- Tables are merged

```
SELECT * FROM merged_reviews LIMIT 10;
```

# Clean Up And Summarize The Customer Reviews



Clean Data →  Better Sentiment Analysis

Spend time cleaning up the customer reviews



AI model for sentiment analysis



Quality of your results → Quality of your input data

# Clean Up And Summarize The Customer Reviews



Clean Data →  Better Sentiment Analysis

## ⚠ Issue

Empty rows



## Why it matters

Adds noise, no value

---

Inconsistent formatting

Affects pattern recognition

---



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

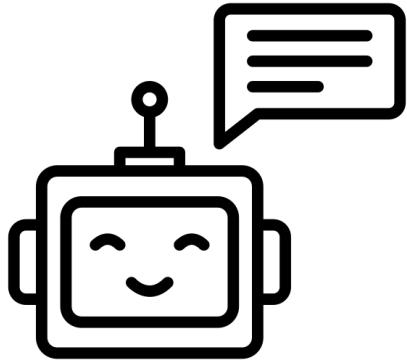
## Sentiment Analysis with Cortex

# MVP Build Plan



1. Getting data into Snowflake ✓
2. Parsing and Structuring ✓
3. Clean up Data ✓
4. Analyze Data
5. Visualize Results
6. Create interface with a tab called "Data" that displays the head of the dataset
7. Deploy to cloud

# How Else Can GenAI Support You?



- Ask what functions are available in Cortex
- Write and debug Python and Snowpark code
- Parse and extract data from complex fields
- Visualize the output in matplotlib or Streamlit
- Suggest error handling or model alternatives



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

## Data Visualization in Snowflake

# MVP Build Plan



1. Getting data into Snowflake ✓
2. Parsing and Structuring ✓
3. Clean up Data ✓
4. Analyze Data ✓
5. Visualize Results
6. Create interface with a tab called "Data" that displays the head of the dataset
7. Deploy to cloud



DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

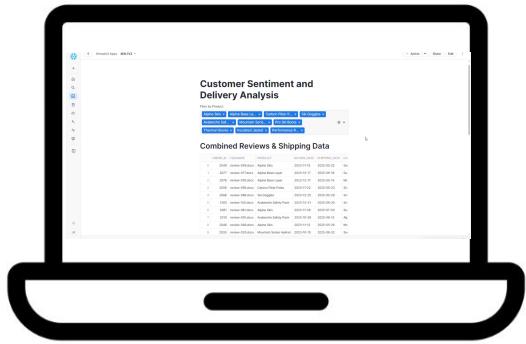
Building Your  
AI-Powered Streamlit  
App Inside Snowflake

# MVP Build Plan



1. Getting data into Snowflake ✓
2. Parsing and Structuring ✓
3. Clean up Data ✓
4. Analyze Data ✓
5. Visualize Results ✓
6. Create interface with a tab called "Data" that displays the head of the dataset
7. Deploy to cloud

# Starting a Snowflake Notebook



## 1 Set Up

Create a new notebook

"Avalanche\_App"

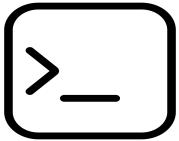
*\*Connected to  
Avalanche database and schema*

You'll write your entire Streamlit app as code in a notebook cell

Point it at the combined table

# Create the App Shell

**Create  
② just a basic  
Streamlit shell**



"Write a Streamlit app that loads a table on Snowflake for sentiment analysis, shows some basic stats, and adds a title and sidebar."



Python code (copy/paste the code)



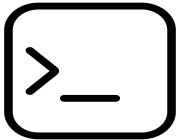
Run the cell

# Create the App Shell

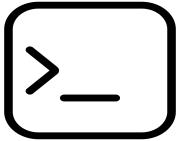
**Create**

② **just a basic**

**Streamlit shell**



"Help me fix this error:....."



"Can you rewrite this function to  
avoid this error?"

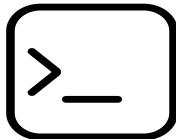
# Add Visualizations

Use charts (created in earlier lessons)

- A line chart of shipments per day
- A bar chart of sentiment distribution
- Product-level stats or top-N product breakdowns

③ Add  
Visualizations

Ask GenAI:



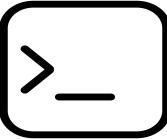
"Make a Streamlit line chart to show shipment."

```
st.line_chart(  
df.set_index("shipping_date")["shipment_count"])
```

# Add Filters

## ④ Use Streamlit widgets

Ask GenAI:



"Add a date and sentiment filter to my Streamlit app."

```
selected_date = st.date_input("Select a  
shipping date")  
  
filtered_df = df[df["shipping_date"] ==  
selected_date]  
  
st.dataframe(filtered_df)
```

# Adding a Chatbot Data Assistant

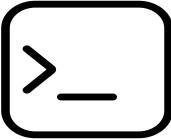
Use

⑤

Streamlit

Cortex

Ask GenAI:



"Create a chatbot in Streamlit that uses Snowflake Cortex to answer questions about the clean\_reviews table."

```
from snowflake.cortex import complete  
  
response = complete(prompt=f"Answer this  
question using the dataset:  
{user_question}", model="gpt-3.5")  
  
st.write(response['choices'][0]['text'])
```

# Adding a Chatbot Data Assistant

install the snowflake-ml-python package  
from the dropdown menu

**Use**

⑤ **Streamlit**

**Cortex**

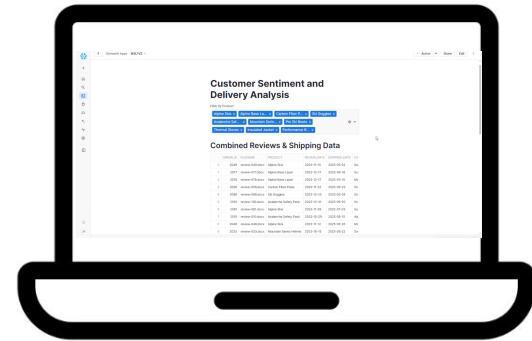
Ask Questions:

- "What's the average sentiment for Product A?"
- "Why were there fewer shipments on this date?"

# Your Final Prototype

⑥ **Run the  
Notebook**

M2/Lesson\_02/M2L2V4.ipynb





DeepLearning.AI

# Fast Prototyping of GenAI Apps with Streamlit

---

## Lab 2 Overview- Using Copilot for Sentiment Analysis

# Your Mission



Build a Product:  
Intelligence Dashboard for Avalanche's product team



- Explore customer sentiment across time and regions
- Visualize trends in shipping delays and delivery performance
- Ask natural language questions

# Your Mission



Build a Product:

Intelligence Dashboard for Avalanche's product team



- Create new visualizations by combining sentiment and shipping data
- Build a filtered table
- Add a chatbot assistant
- Use GenAI heavily throughout the process

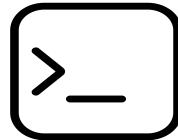
# What You'll Do

1

## Connect to Snowflake and Load Data

Use your existing combined table

Ask GenAI:

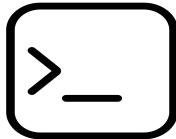


"Write Python code to connect to Snowflake and load the reviews\_with\_sentiment table into a Pandas DataFrame."

# What You'll Do

## ② Create a New Streamlit App

Start a new notebook and ask GenAI:



“Create a basic Streamlit app that loads my dataset and includes a sidebar with product filters.”

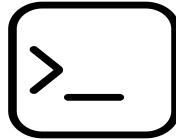
Add a title, sidebar filters, and a data preview

# What You'll Do

**Visualize**

③ **Sentiment  
by Region**

Ask GenAI:



"Plot average sentiment score by region using matplotlib and Streamlit."

Your plot  
should help  
answer:

*Which regions have the most negative feedback?*

# What You'll Do

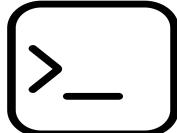
**Highlight  
Delivery  
Issues**

④

Create a table that shows:

- product\_id
- region
- delivery\_status
- sentiment

Ask GenAI:

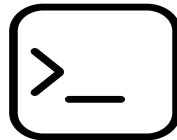


“Filter for reviews with negative sentiment and delivery issues. Show a table grouped by region and product\_id.”

# What You'll Do

## ⑤ Add a Custom LLM-Powered Chatbot

Ask GenAI:

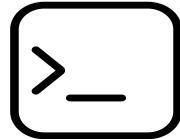


"Add a chatbot to my Streamlit app using Cortex complete"

# What You'll Do

⑥ **Deploy or Share**

Ask GenAI:



"How do I deploy this Streamlit app in Snowflake?"

"Follow GenAI's steps to share the app"

# Tips for Success



Use GenAI for every step



Keep your app simple and focused



Add comments to your code

# Where to Start



In the course Github repo:

- M2/Lesson\_02/Lab2/