DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Module 1: Introduction to Prototyping Gen AI Applications

Course Overview

✅ How GenAI tools can speed up your app development process

✅ Build real apps that respond to user input

✅ Generate Python code on the fly

✅ Use design principles like MVP and MAP to stay focused on what matters

Module 1:

Build a basic GenAI interface

↓

💡 Prompt for Python code

↓

🚀 Run in Streamlit

↓

🎉 Share working prototype

Build Fast with the Tools You Have

- Large language models  Claude Gemini
- Lightweight app frameworks Streamlit gradio 
- Prompt engineering techniques + basic RAG

*Basic knowledge of SQL is helpful but not required

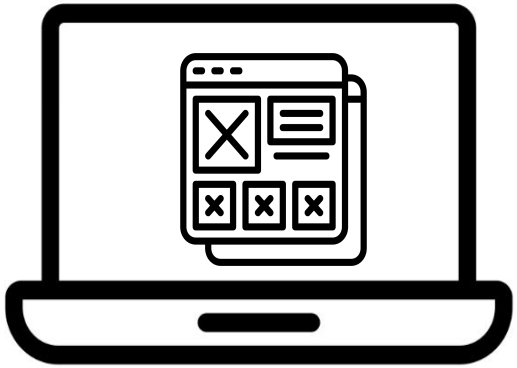
DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

The Benefits of Prototyping

Defining Prototyping for Gen AI

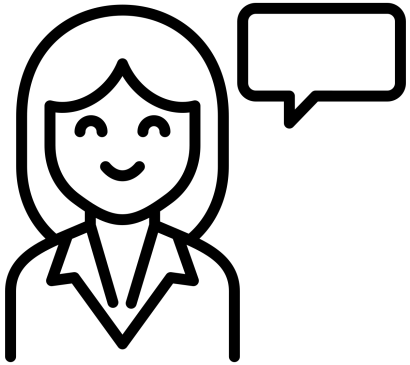
A **prototype** is a quick, lightweight and tangible version of an idea



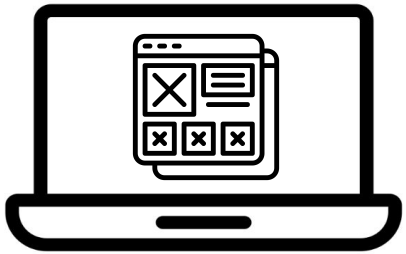
- Built fast
- Not polished or complete
- Meant for testing, not for launch
- Often discarded or heavily modified after evaluation

When to Prototype

Tell



Show



Showing your vision with a **prototype** beats telling people about your idea every time

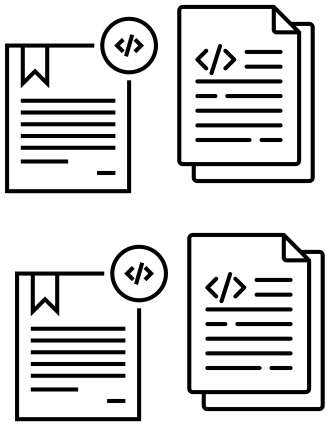
When to Prototype

Get feedback early

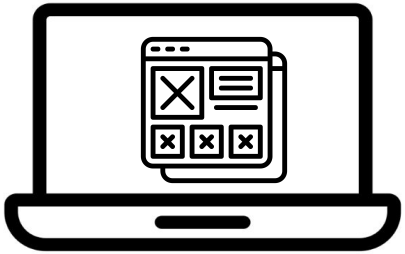
Test risky ideas without building a full product

Validate your assumptions using real data and users

When to Prototype



misinterpreted!

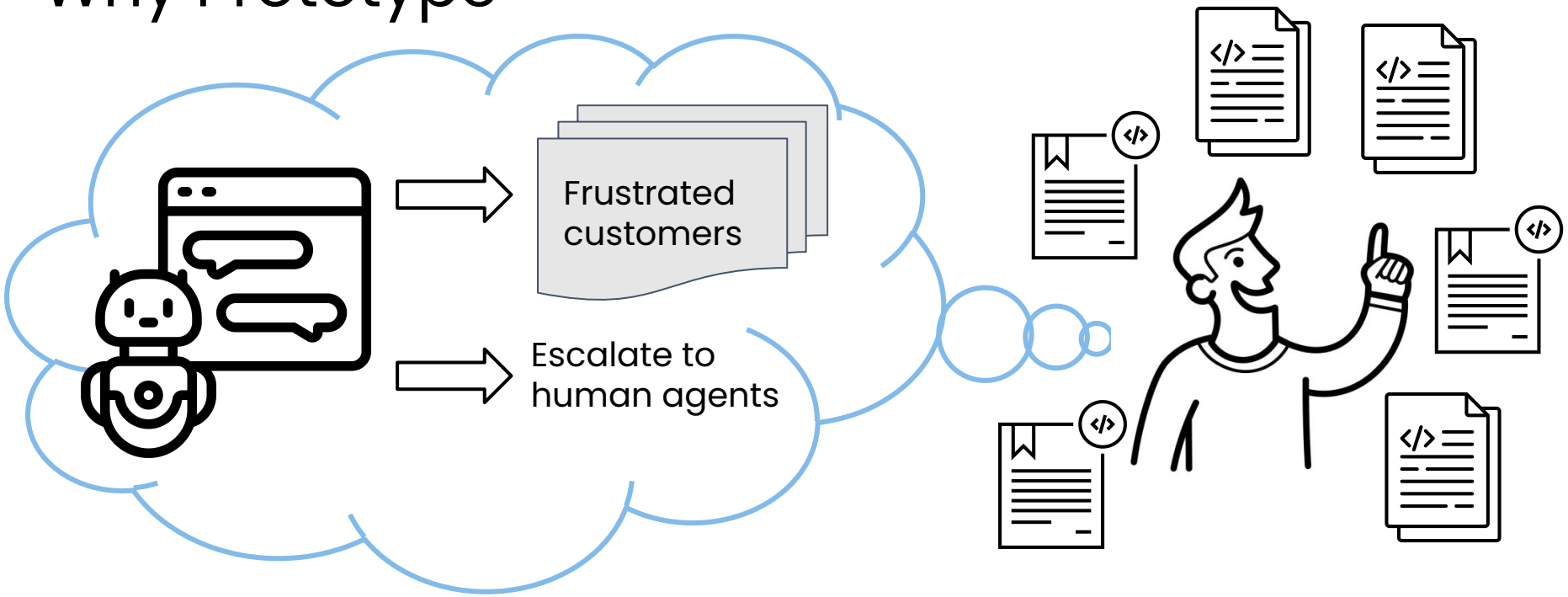


how your tool works!

Why Prototype

Forbes lists **"failure to prototype"** as one of the top 10 reasons why software projects fail.

Why Prototype



Why Prototype

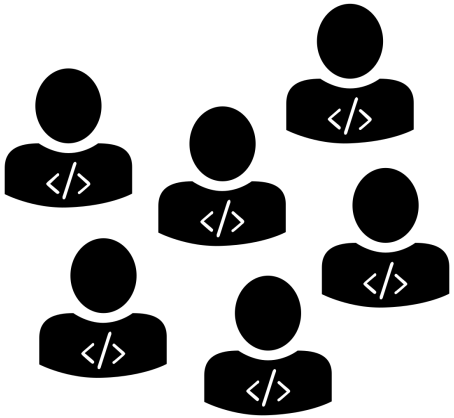
Alex

"conversational interface”

drop-down menus and forms

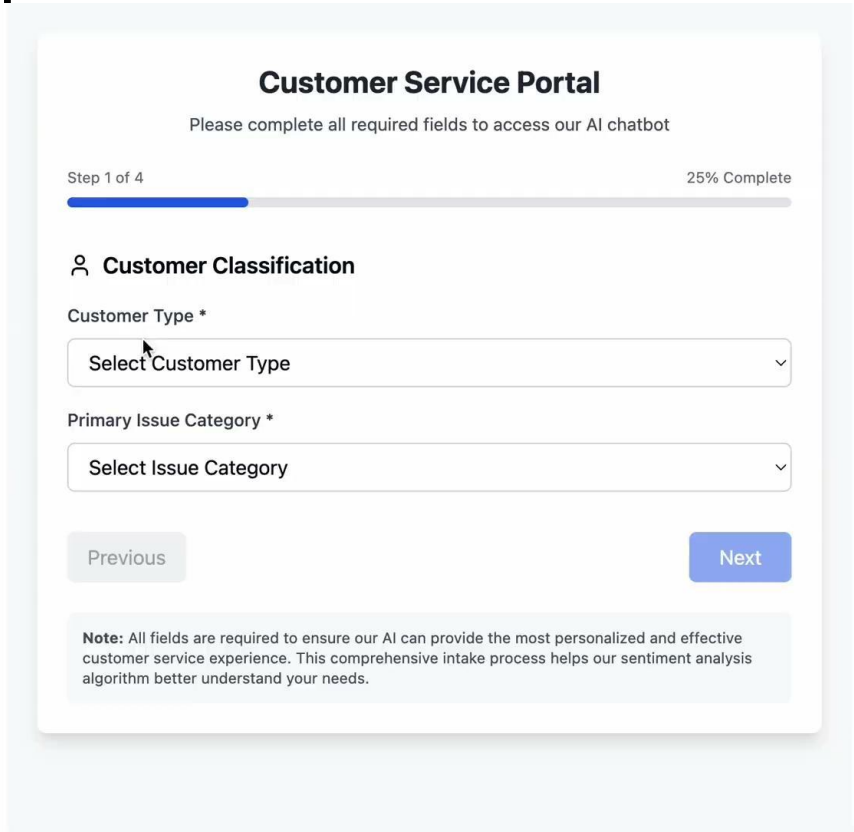
"sentiment analysis” (😊,😐,😠)

"I need help" → angry customer



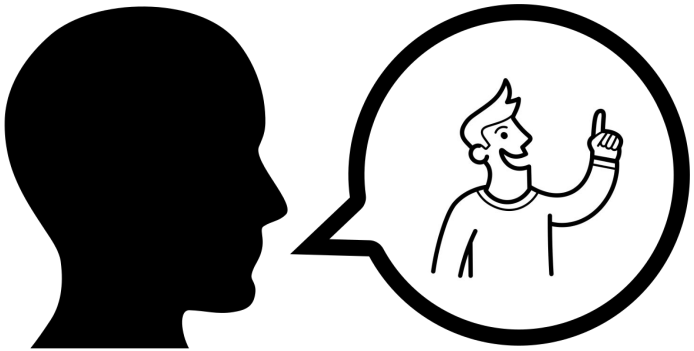
Developers

Why Prototype



Why Prototype

"This feels like a step backward from our current support system."

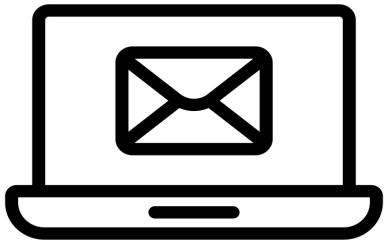


❗ This project was shelved after six weeks of development time.

Building a Prototype

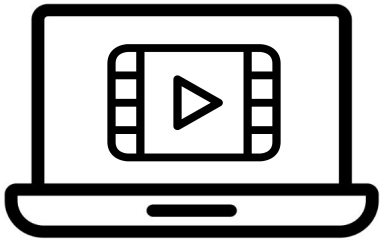
Building a prototype is a proven strategy!

Prototype → Launch → Grow



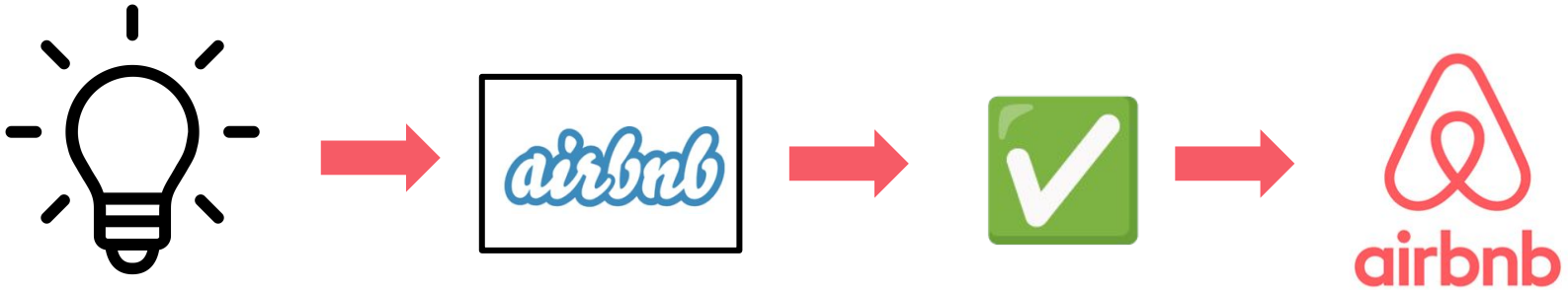
Landing page

Basic app



Video

AirBnB

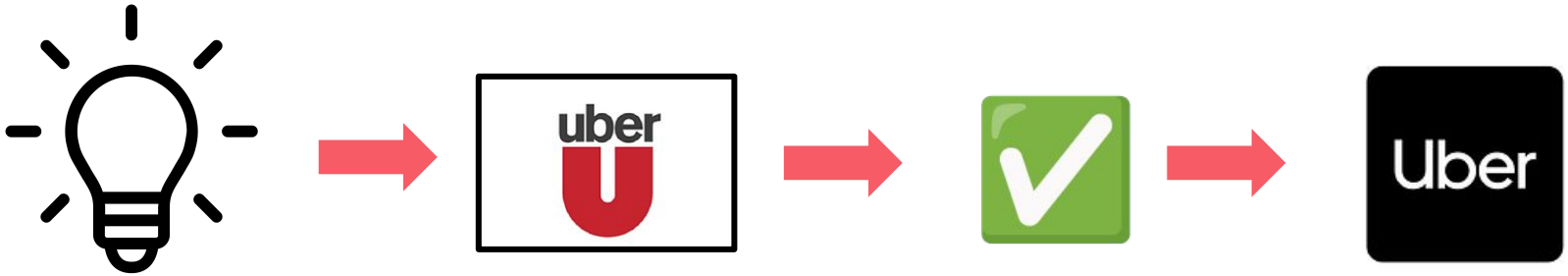


Short-term rental

Prototype

Gather early feedback

Uber



Connect riders with car drivers

Prototype

Gather early feedback

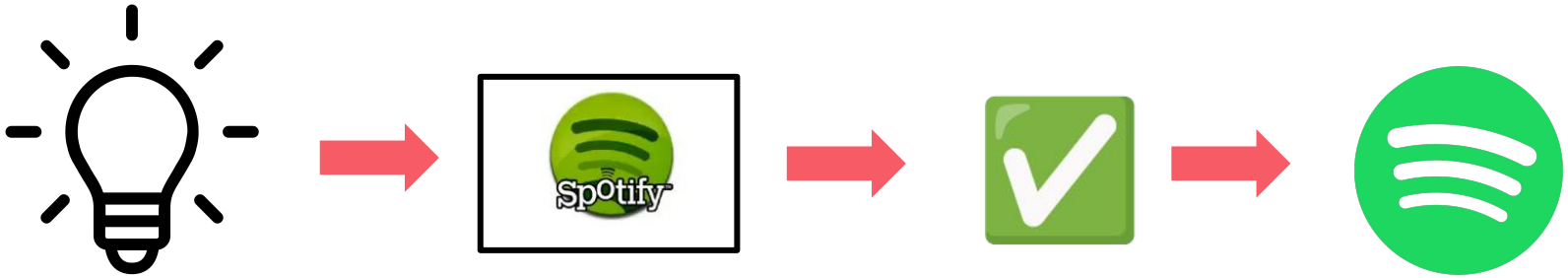
Instagram

Photo sharing with basic filters

Prototype

Gather early feedback

Spotify



Music streaming experience

Prototype

Gather early feedback

GPTZero

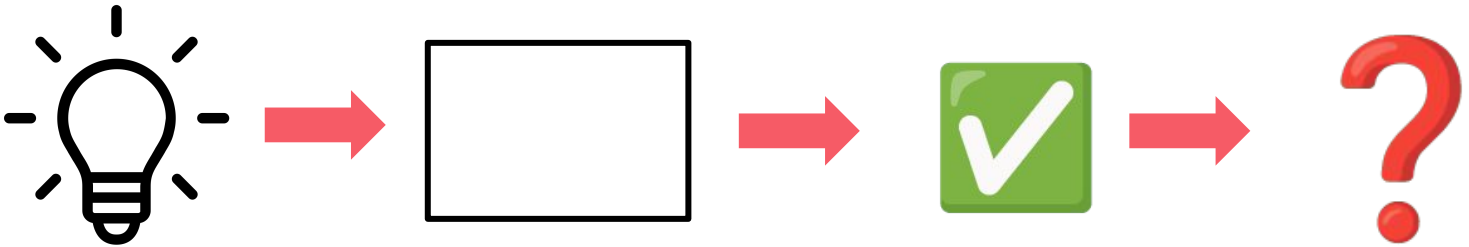
Detect AI generated text

Prototype

Powered by

Gather early feedback

Each Example Follows the Same Pattern



DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

How GenAI Revolutionized Prototyping

Prototyping Before GenAI

App that analyzes product reviews

Without GenAI:

✅ Design the interface ⇨ text box, button, display

✅ Write code ⇨
- import libraries for sentiment analysis
- clean and load your data

✅ Connect input to backend logic
✅ Display the results

✅ Test, fix errors, and refactor

Prototyping with GenAI

With GenAI:

“Create a Streamlit app that lets users paste in reviews, run a sentiment analysis and show the results as a pie chart”

AI writes the code

You click **RUN** →
and you’re already **testing** your idea 🎉

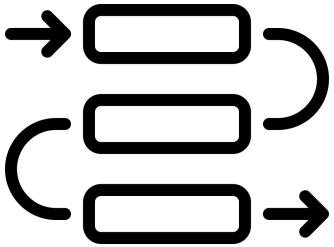
App that analyzes product reviews

⇨

- import libraries
- set up the interface
- run the analysis
- create the chart

It’s Not About Code, It’s About Mindset

Traditional software development

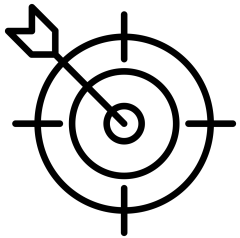


planning first

building second

“How do I build this?”

With GenAI

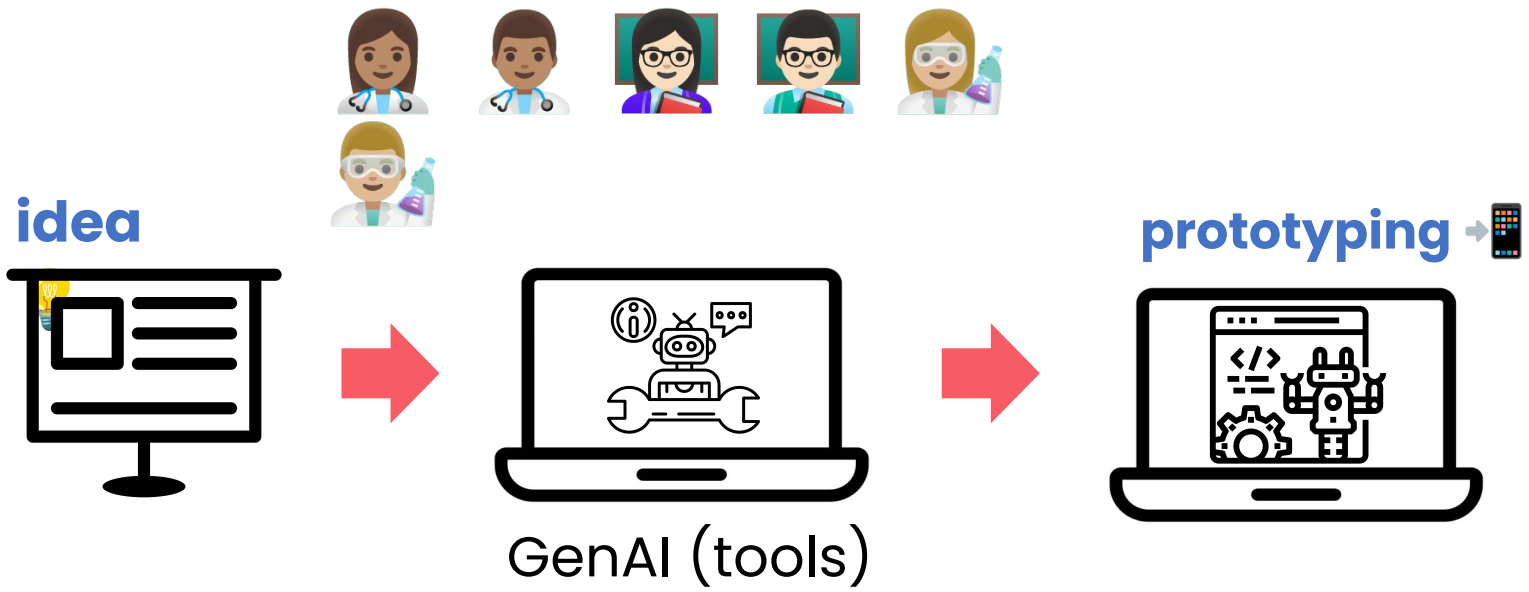


You start with intent

“What’s the fastest way to test if this idea even makes sense?”

Shifting you from **solution-first** to **exploration-first**

Democratizing App Development with Generative AI



GenAI changes the game!

Removing the bottlenecks

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

The Prototyping Development Cycle for GenAI

Prototyping with GenAI

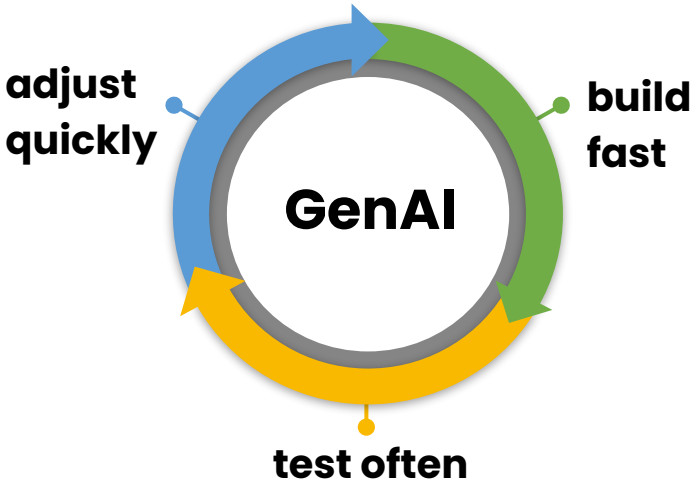
GenAI development = **exploration** + **iteration**

🔮 You don't always know what the output will be

😱 The GenAI model might surprise you

💥 Your prompt might break

🤦 Users might behave unexpectedly



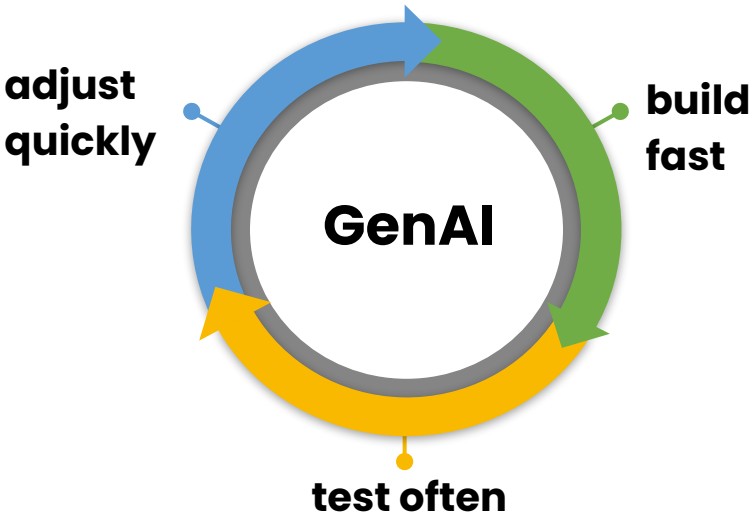
Prototyping with GenAI

- Explore how a GenAI model behaves with different prompts
- See how users interact with your app
- Catch edge cases early on
- Decide whether your idea is worth building into a full product

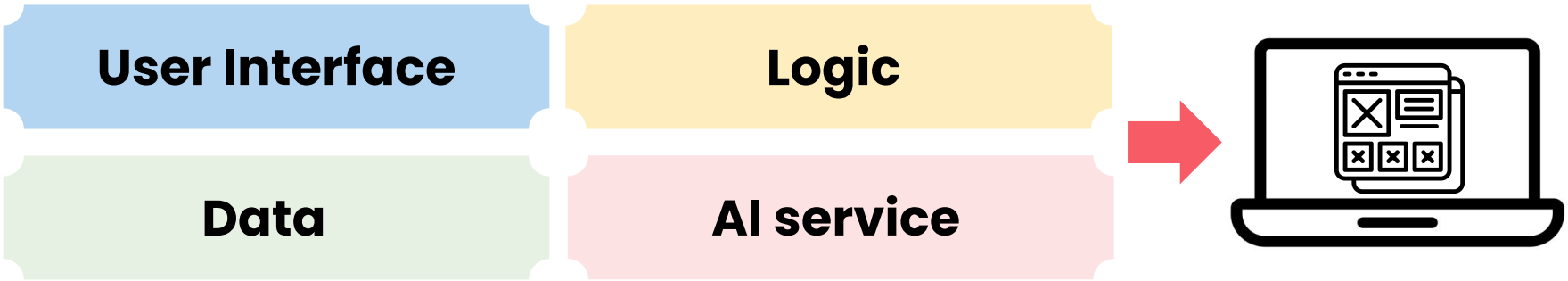
Prototyping with GenAI

Prototyping →

Default Development Mode



What a Working Prototype Looks Like



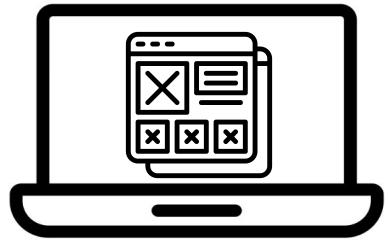
What a Working Prototype Looks Like

lets people talk to your app

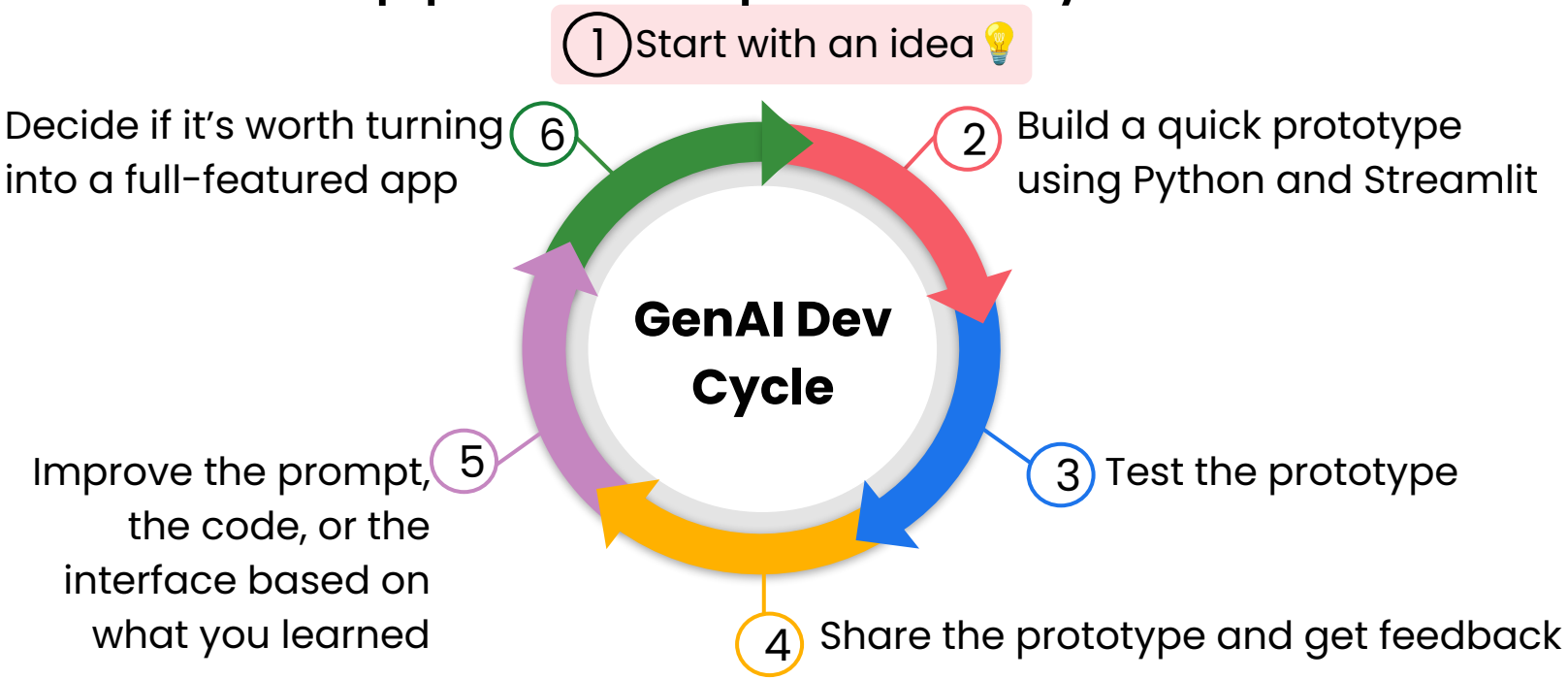
what to do

something to work with

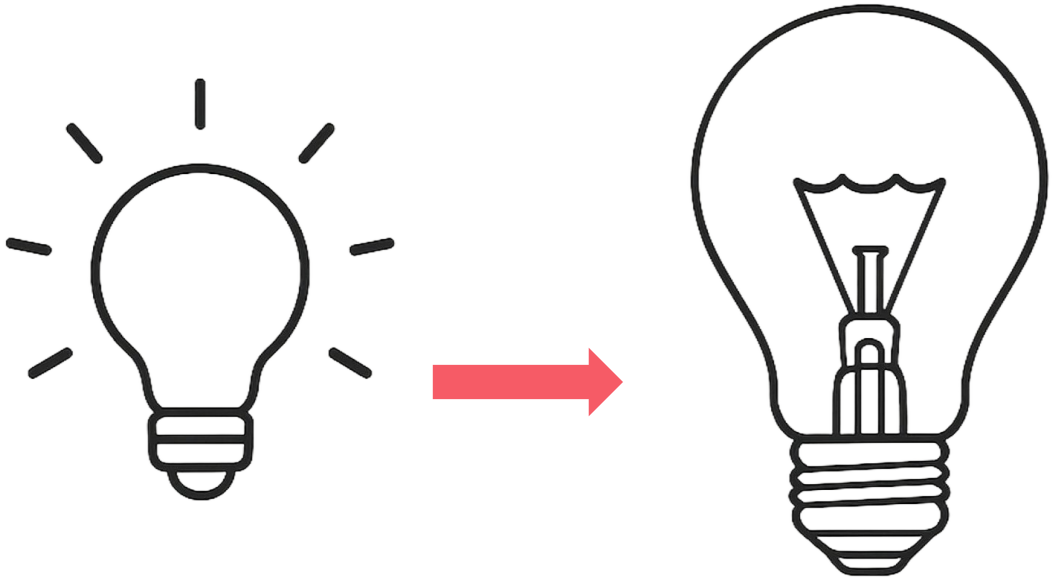
OpenAI's GPT-4



The GenAI App Development Cycle



Prototype vs Production-Ready



Prototype

Production-Ready

Prototype vs Production-Ready

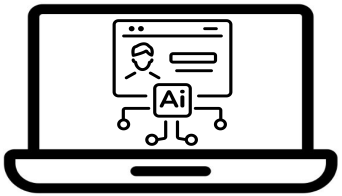
	Prototype	Production-Ready
🎯 **Purpose**	Testing and learning	Deliver a complete stable product
👆 **Priority**	Focus on speed	Focus on architecture and reliability
💻 **Code quality**	Code can be messy but functional	Code should be refactored, clean, and maintainable
📊 **Data**	Use small or simulated datasets	Real, clean, validated data

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Avoiding Common Pitfalls

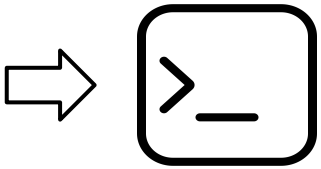
Over-engineering



Over-engineering our prototypes

🗣️: ask AI to build everything at once

- UI
- Backend
- Database
- Analysis
- Charts



Giant Prompt

- 🚫 Too much code
- 🚫 Pieces do not fit
- 🚫 Debugging chaos
- 🚫 Spending hours untangling the results

Over-engineering

❗ Avoid over-engineering prototypes

✅ Keep It Simple

✅ Try one idea at a time

✅ Start with something small

Good prompting example:

1st Prompt → Build a login form UI
2nd Prompt → Create backend auth logic
3rd Prompt → Connect login to backend

GenAI works best when the task is clearly defined

Getting Stuck in Polishing Mode

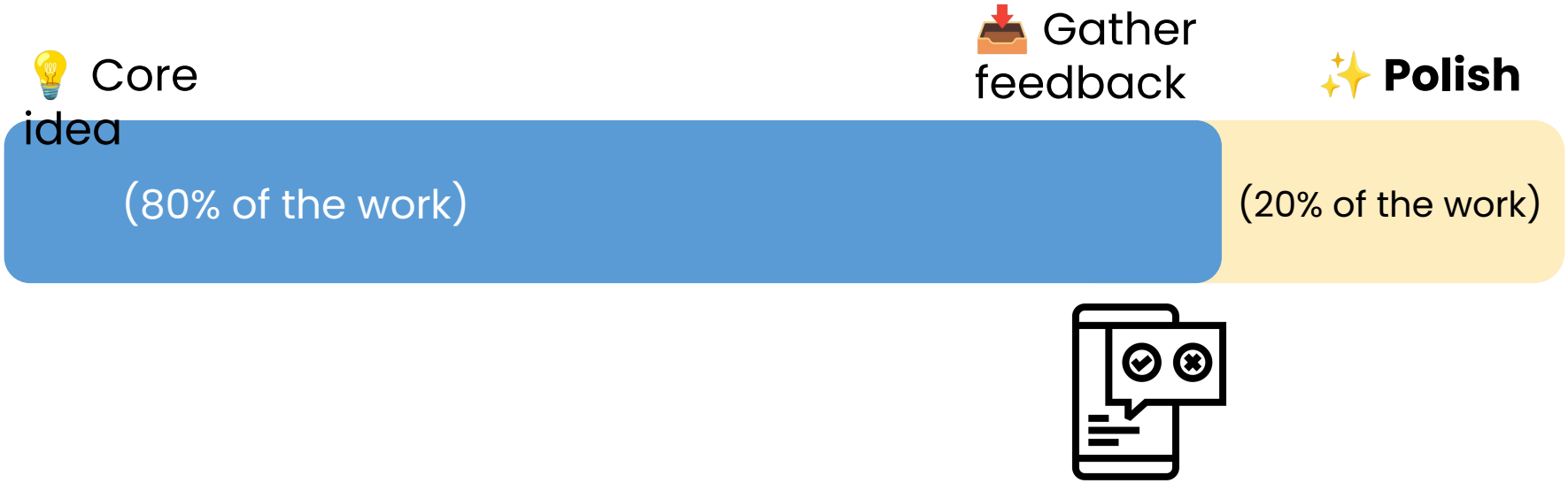
"Trying to perfect everything before getting feedback"

Spending way too much time ⌛

- Tweaking prompts
- Redesigning charts
- Debating button colors
- ...

➡ Nobody used the app 📵

Getting Stuck in Polishing Mode

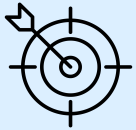


Optimizing Without Validating

🌴 Vacation Ideas App

Very long long prompts...

Do people even like the recommendations?



“Do users find this helpful?” or “Did they actually take action?”

Building Without a Plan To Scale

The AI code might be:

- Messy 🧶
- Undocumented 📄
- Hard to follow 😵

Time to **share your work**:

✅ Add comments

✅ Stick to simple file names and folder structures

✅ Use tools your team already knows

✅ Use GenAI to write some quick test cases

Trusting AI Output Without Verifying

Don’t just hit “run” and hope for the best 🤞

Test your prompts

- Reword prompts
- Make prompts simpler
- More clear instructions
- Small tweaks → change the results

Trusting AI Output Without Verifying

Check the output

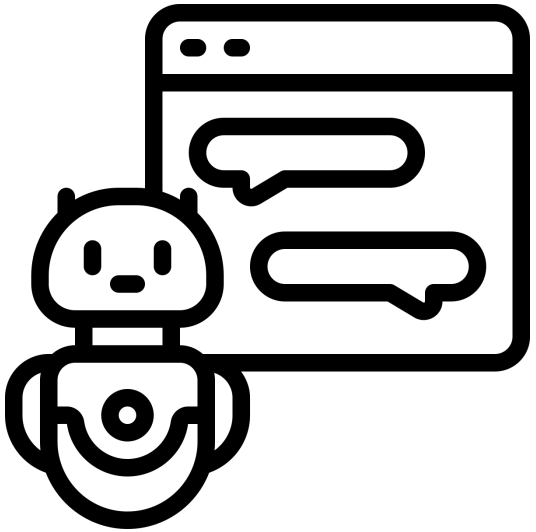
- Run a few edge cases
- Check if the AI output actually makes sense
- Ask someone else to try it
- Log your inputs/outputs

✅ Fix what is broken

✅ Test again

Is this idea worth building?

Challenges in GenAI-driven Prototypes



✅ Be specific in your prompts

✅ Use a small dataset when testing

✅ Read the AI's code before running it

✅ Use version control so you can undo mistakes

✅ Don't expect the same result every time — GenAI isn't always consistent

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Introducing the Course Project and Dataset

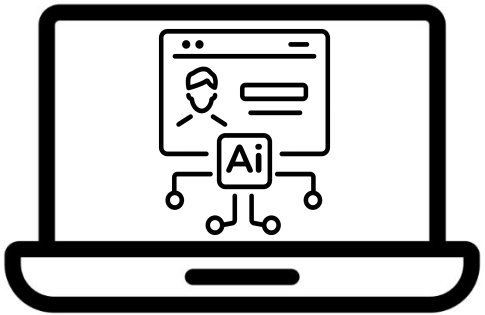
Introducing the Final Project

👩‍💻 You'll build a real working app powered by GenAI!

✅ Plots the average sentiment scores by product and delivery status

✅ Analyzes customer sentiment

✅ Lets users ask natural language questions about the data



Can GenAI help our team understand customer feedback faster?

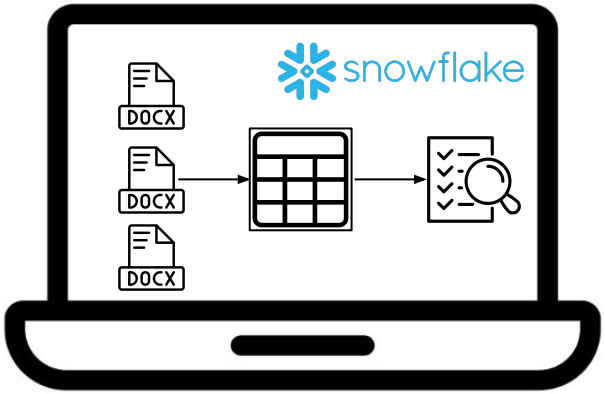
What You'll Actually Do

Phase 1 – Build with Streamlit

- Load and clean your dataset
- Build a simple interactive interface
- Visualize results with charts and filters
- Publish your first prototype online

What You'll Actually Do

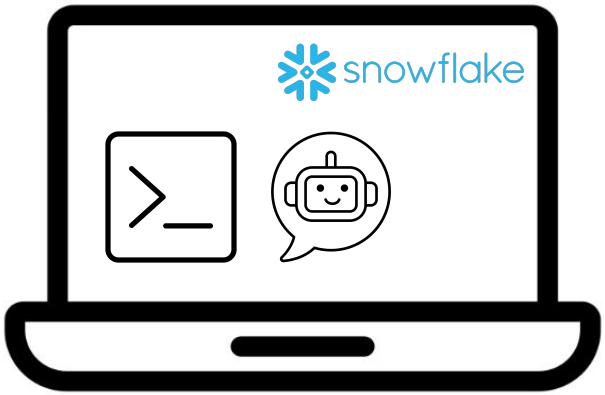
Phase 2 – Move to Snowflake



- Upload raw DOCX files
- Ingest and store in a structured format
- Use Snowflake's built-in AI tools to analyze
- Deploy your fully functional app complete with a chatbot

What You'll Actually Do

Phase 3



- Connect your chatbot to the data
- Add prompt engineering and RAG to improve responses
- Learn techniques for getting feedback, fast

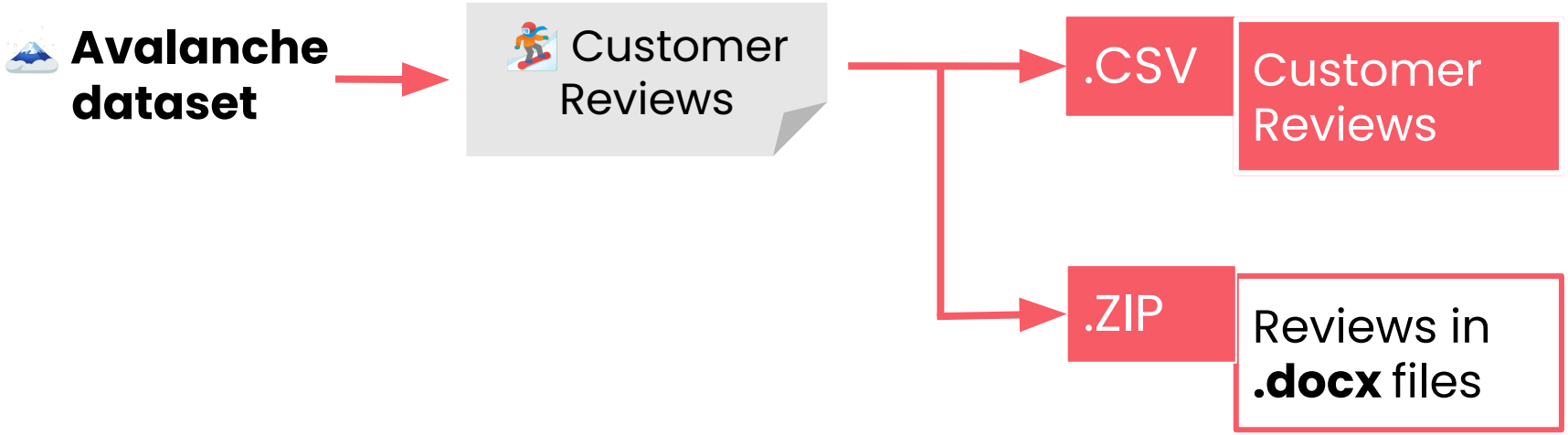
Introducing the Avalanche Dataset

Avalanche dataset

→ customer_reviews.csv

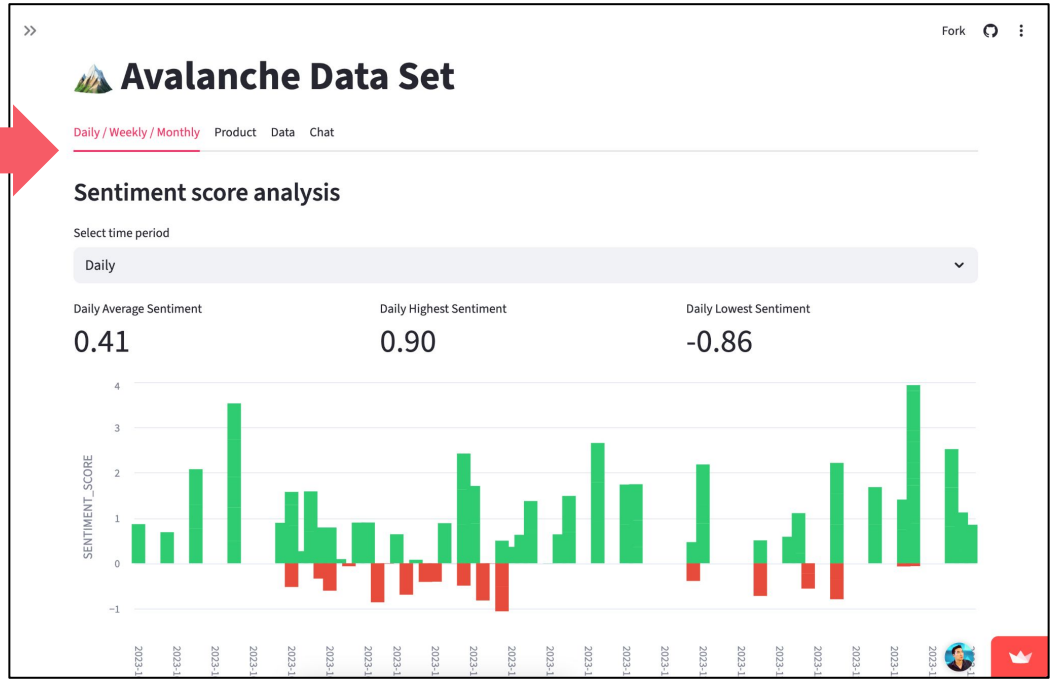
PRODUCT	DATE	SUMMARY	SENTIMENT_SCORE
Alpine Skis	2023-10-30	The skis have durability issues, with edges delaminating after ten days and excessive topsheet wear, compromising performance and structural integrity. The manufacturer's warranty response has been unsatisfactory.	-0.86290675
Thermal Gloves	2023-11-10	The waterproof gloves suffered from major failures after three days of use, with water penetrating through the shell and causing hands to get soaked during a mild snowfall. The leather treatment and dye quality were inconsistent, leading to color transfer onto other equipment and potential staining. Additionally, there were issues with inconsistent stitching and sizing between the left and right gloves. The premium price point of $150 was not justified due to these significant quality control issues.	-0.8224165
Insulated Jacket	2023-12-09	The insulated jacket was purchased for winter activities like snowshoeing and skiing, but its breathability was inadequate during sustained aerobic efforts, causing rapid overheating and discomfort. The jacket trapped moisture and heat, leading to potential safety issues and decreased performance. It was unsuitable for strenuous winter sports.	-0.7224479
Mountain Series Helmet	2023-11-02	The received garment was significantly smaller than expected, causing frustration due to size discrepancies between the published size chart and the actual product. The return process was complicated by glitches in the online portal and unresponsive customer service, who initially required the customer to pay for return shipping despite the sizing error being the company's fault. Despite promising features and high-quality materials, the garment's performance and durability couldn't be evaluated due to the sizing issues. The overall experience was disappointing for a premium brand.	-0.69523174

Introducing the Avalanche Dataset



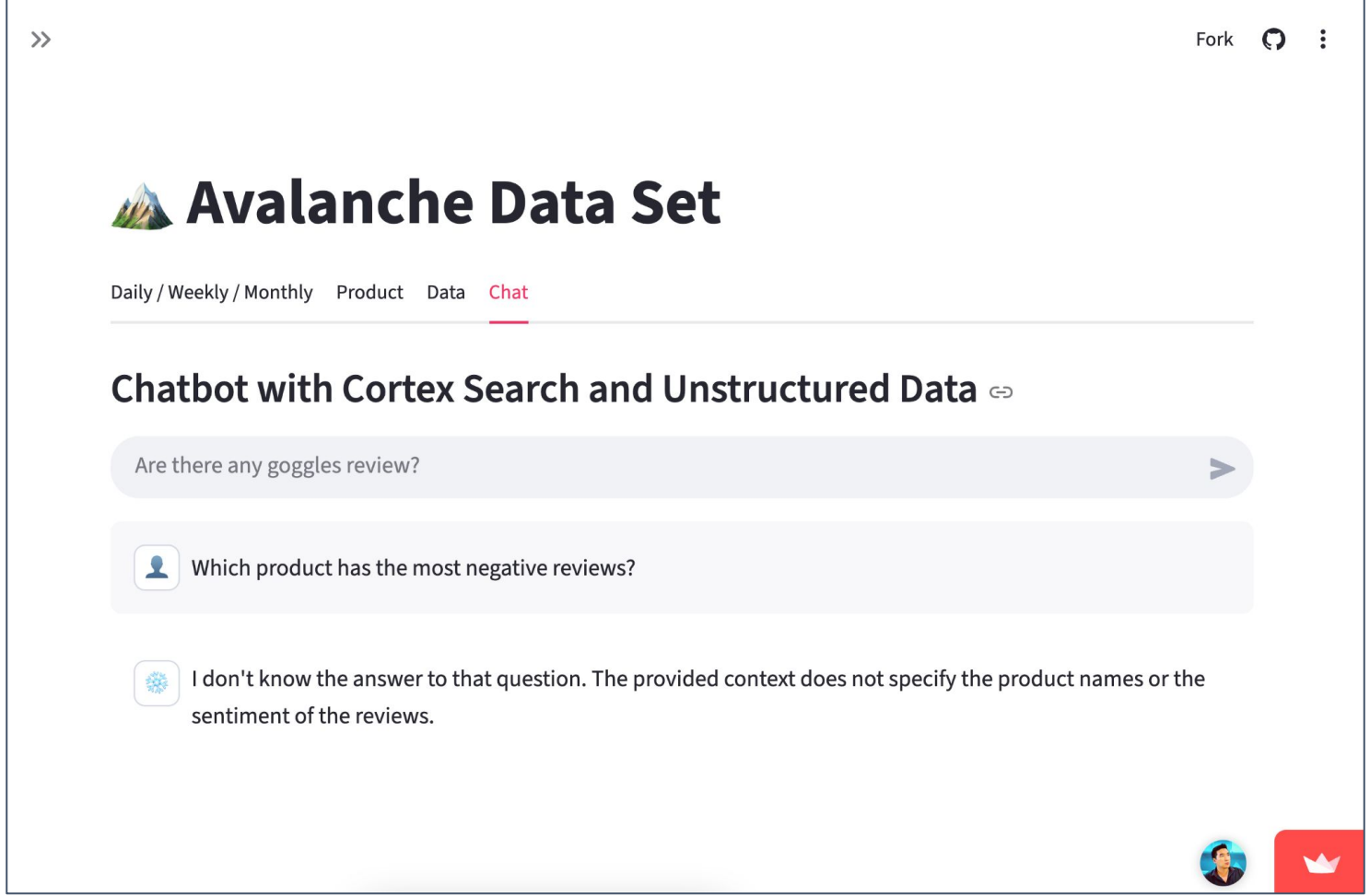
What You'll Build

https://avalanche.streamlit.app/



DeepLearning.AI

snowflake®


Fork
Avalanche Data Set
Daily / Weekly / Monthly
Product
Data
Chat
Chatbot with Cortex Search and Unstructured Data
Are there any goggles review?
Which product has the most negative reviews?
I don't know the answer to that question. The provided context does not specify the product names or the sentiment of the reviews.

DeepLearning.AI

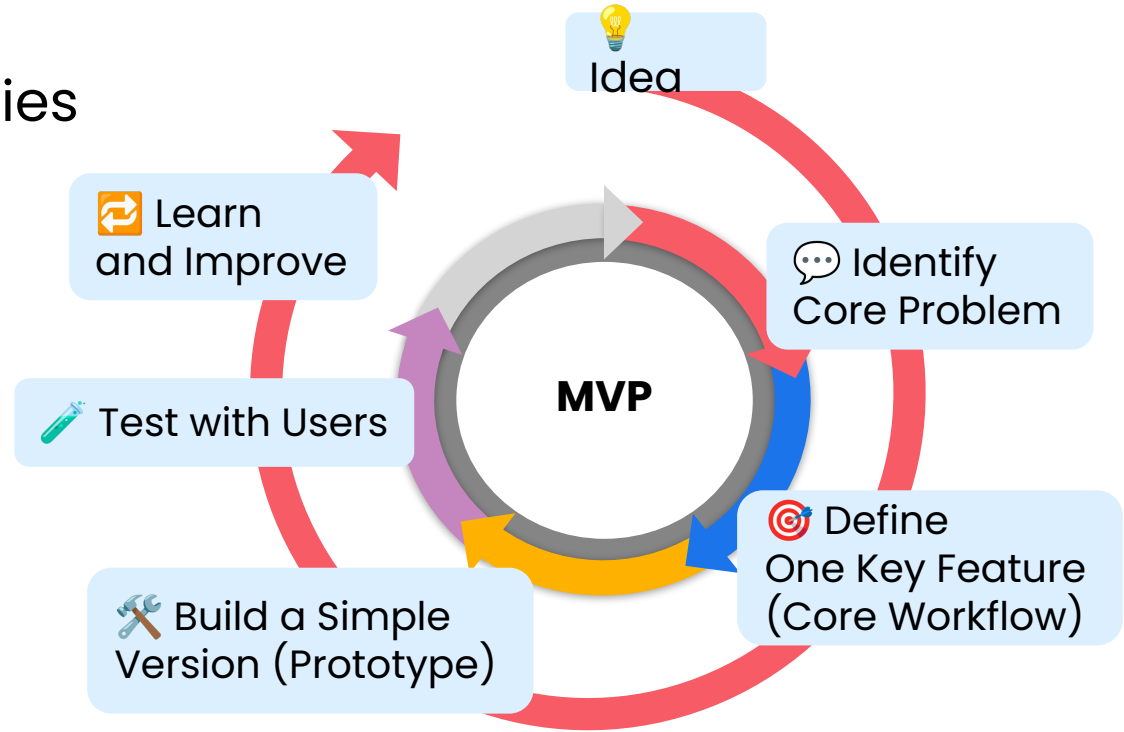
Fast Prototyping of GenAI Apps with Streamlit

Scoping an MVP

Minimum Viable Product (MVP)

📔 Lean Startup by Eric Ries

MVP Framework



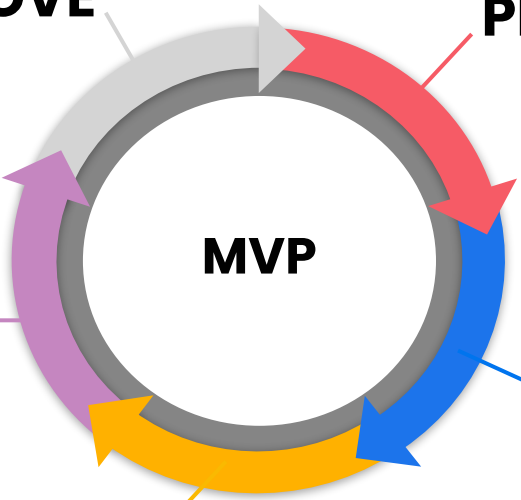
Minimum Viable Product (MVP)

- Interface
- Response to users...

IMPROVE

PLAN

- Prioritize
- Set tools...



- Feedback
- Improvement
- Errors...

LEARN

BUILD

- Dashboard
- Web app
- Data app...

TEST

- Collect feedback...

MAP Framework

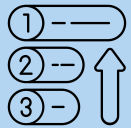
Mission

What's your end goal?

Audience

Who are you building for?

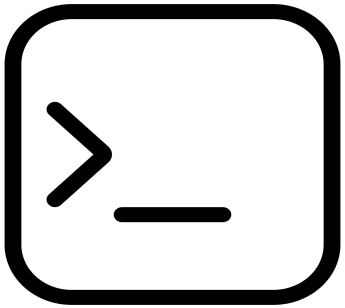
Priorities



What's the one thing it needs to do first?

M for Mission

What problem are you trying to solve?



~~"I want to build an AI chatbot"~~ ❌

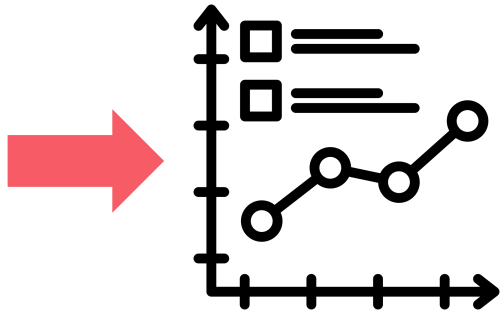
"Customer service agents spend 20 minutes per call looking up product information, and I want to reduce that to 2 minutes"

Your end goal should be concrete and measurable!

M for Mission

How will you know if your app solved the problem?

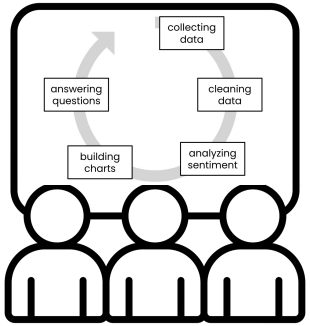
When is it solved?



METRICS

BEFORE → 5 hours

data scientists



NOW → 30 min

A for Audience

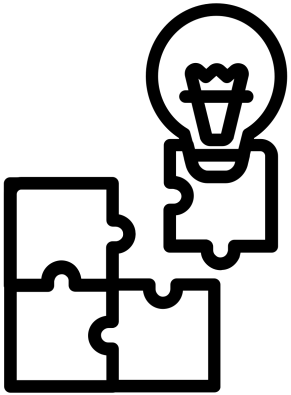
Who are you building for?

“Identify your primary user group”

- Are they technical experts or complete beginners?
- What's their biggest frustration with existing solutions?
- What workarounds do they currently use?
- How do they measure success in their role?

P for Priority

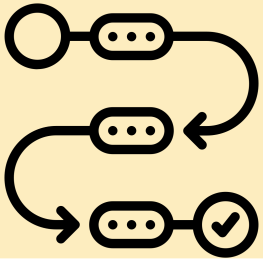
Many ideas for features...

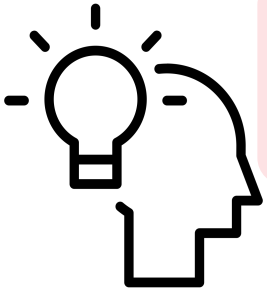


absolutely necessary ❗

what can wait ⌛

Focus on one core workflow





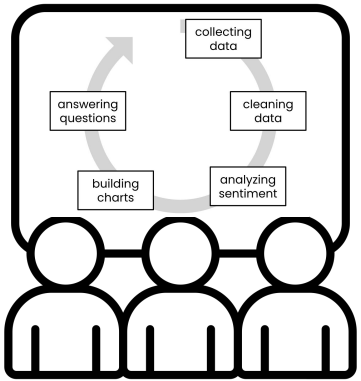
If I could only build one feature to show the value of this app, what would it be?

Avalanche Mission 🏂 🚀

🎯 What problem are you solving—and how will you know that it's solved?

❌ Same sentiment analysis workflows

❌ Manually digging through customer reviews



✅ **Automate** the setup

✅ Make it **easy** to get quick answers directly from the dataset

"Did this save you time?" ⌛ ➡ You're on the right track 🚀

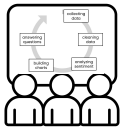
DeepLearning.AI

snowflake®

Avalanche Audience

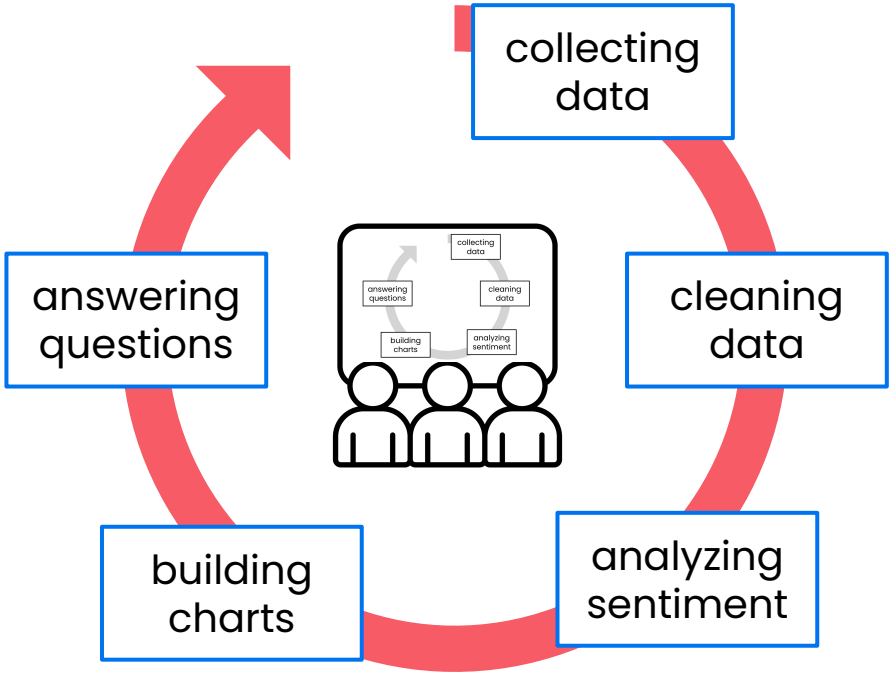
1. **The manager**

2. **Your teammates**



- Every time they do sentiment analysis (😊,😐,😠),
- They have to rebuild the process from scratch

Avalanche Audience



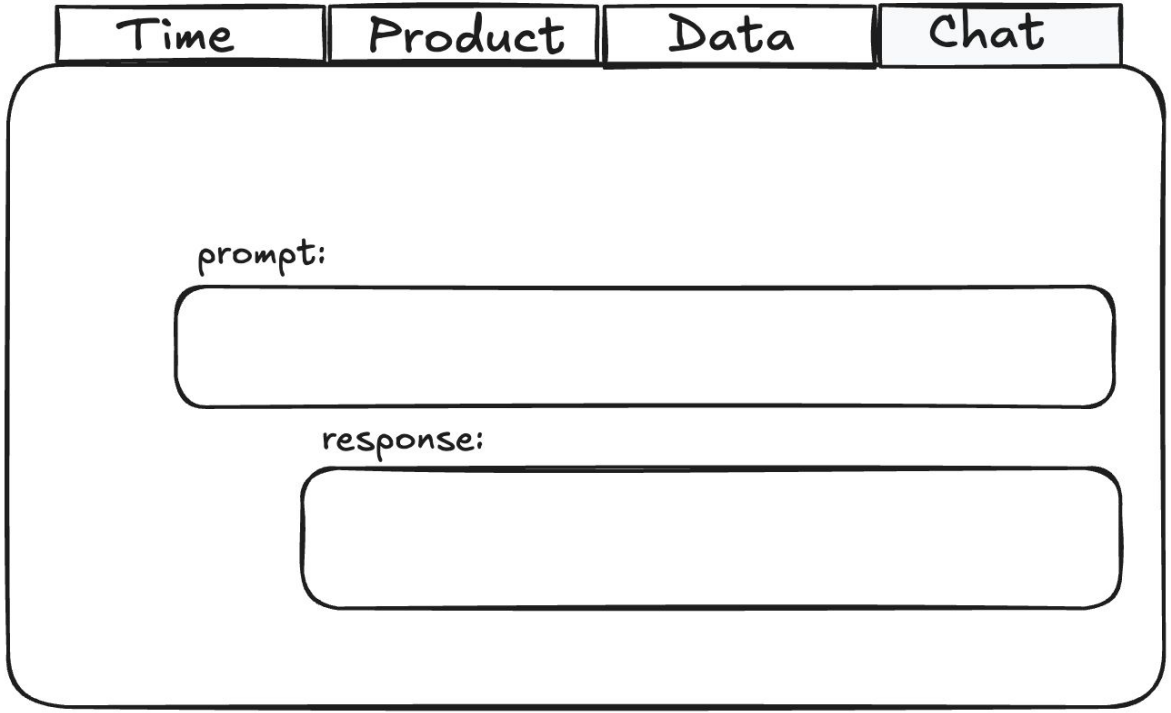
You build something people will actually use!

Prioritizing Features for the Avalanche Project

Avalanche dataset

If you could only build one feature, what feature would be the most critical to test if your idea works?

Prioritizing Features for the Avalanche Project



Prioritize What to Build

✅ Must Have

Cleaning the dataset

Running sentiment analysis

Visualize data

AI chatbot

Deploy

✨ Nice to Have

Upload dataset

Showing the raw data table

🚀 Next Version

Printing results

Build Your Plan

✅ Must Have

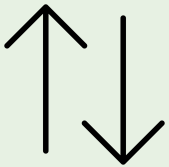
Cleaning the dataset

Running sentiment analysis

Visualize data

AI chatbot

Deploy



✨ Nice to Have

Upload dataset

Showing the raw data table

🚀 Next Version

Prin[illegible] resu[illegible]

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Overview of the Course Github Repo

https://github.com/https-deeplearning-ai/fast-prototyping-of-genai-apps-with-streamlit

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Lab 1: Co-creating an MVP Plan with GenAI

The Challenge

Using GenAI to extract structured insights from clinical trial PDFs

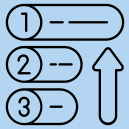
“Act as a product manager. Help me define an MVP for an app that extracts data from clinical trial PDFs.”

The MVP Planning Process


What's your end goal?


Who are you building for?


What does "a working app" look like?


Prioritized features

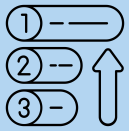
Reduce the time her team spends on **manually extracting data** from messy clinical trial reports.

The MVP Planning Process


What's your end goal?


Who are you building for?


What does "a working app" look like?


Prioritized features

Clinical operations staff — non-technical users

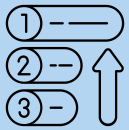
The MVP Planning Process

What's your end goal?

Who are you building for?

What does "a working app" look like?

Metric →
"did this save you time?"



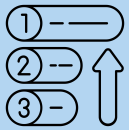
Prioritized features

The MVP Planning Process


What's your end goal?


Who are you building for?


What does "a working app" look like?


Prioritized features

- Drag-and-drop PDF upload
- LLM pipeline to extract and clean the data
- Editable table of results
- Export to CSV

How to Work with GenAI More Effectively

A few more tips from GenAI to build a better prompt:

✅ Be Specific

"Help m bui an app to extract ta from PDFs"

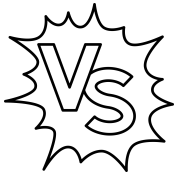
"I'm building a GenAI assistant to automate the process of extracting data from clinical trial PDFs to extract insights from the data they contain. What core features should the MVP include?"

How to Work with GenAI More Effectively

A few more tips from GenAI to build a better prompt:

✅ Provide context

"I'm an ML engineer building an internal tool"



"I'm a student building a school demo"

How to Work with GenAI More Effectively

A few more tips from GenAI to build a better prompt:

✅ Specify Format

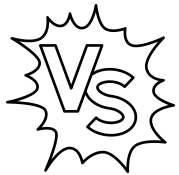
- Summarize this as bullet points.
- Generate a step-by-step plan.
- Create a table of MVP vs stretch goals.

How to Work with GenAI More Effectively

A few more tips from GenAI to build a better prompt:

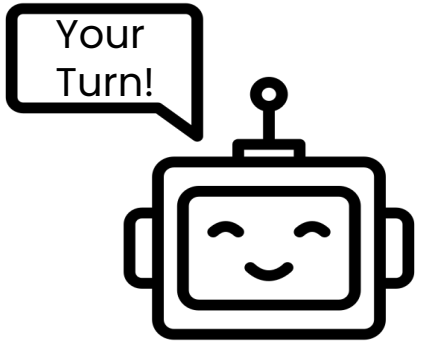
✅ Give the AI a Role 🎭

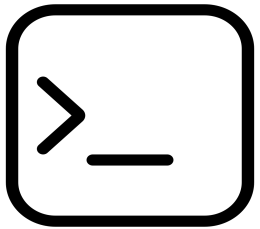
"Help me scope my app"



"**Act as** a product manager for an ML team building a GenAI prototype"

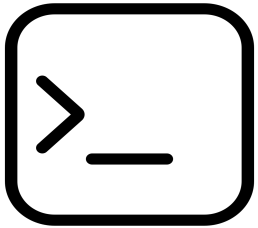
Your Task: Define Your MVP with GenAI





"I'm new at using GenAI for prototyping. Help me plan an MVP for a Streamlit app that loads a CSV, performs sentiment analysis using OpenAI, shows a bar chart of the results, and lets users filter by product"

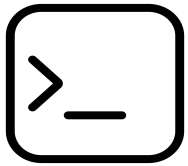
Improving the Prompt



***Act as** a product manager. Help me **define an MVP** for a **GenAI app**.*
Use these principles:
- Define the end goal
- Identify the audience
- Choose success metrics
- Prioritize core features
Then walk me through each step:
- What problem are you solving?
- Who are you solving it for?
- What would success look like?
- What are the must-have features?

Build a Plan

MVP Defined ✅



Now, that we used the principles before, **help me organize the features into**:

- Must-Have
- Nice-to-Have
- Next Version

✅ Must Have	✨ Nice to Have	🚀 Next Version

Remember...

✅ GenAI is great at

- Creating tedious setup code
- Brainstorming features
- Writing queries
- Recommending architecture
- Planning and debugging

💪 Humans are needed for

- Picking the right problem
- Making UX decisions
- Applying business logic and
- Validating AI outputs

Assignment: Plan Your MVP

1. Write a 1–2 sentence description of your GenAI app idea.
2. Use your favorite GenAI tool (e.g., ChatGPT, Claude, or Gemini).
3. Copy/paste the structured prompt shown above.
4. Ask for help defining your MVP.

Assignment: Plan Your MVP

5. Categorize your features as either (Must-Have, Nice-to-Have, Next Version).
6. Stretch goal: Ask the AI to run a quick competitive analysis.
7. Document your plan as a 1-page summary.

Lesson 2

Getting Started with Streamlit

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Choosing the Right Tools

You Don’t Need a Full Stack

You don’t need to: 🚫

- Spin up a backend
- Configure a database
- Set up hosting

To test an idea!

Build a working prototype with: ✅

- Python
- A lightweight UI library
- Access to a GenAI model

Comparing UI Frameworks

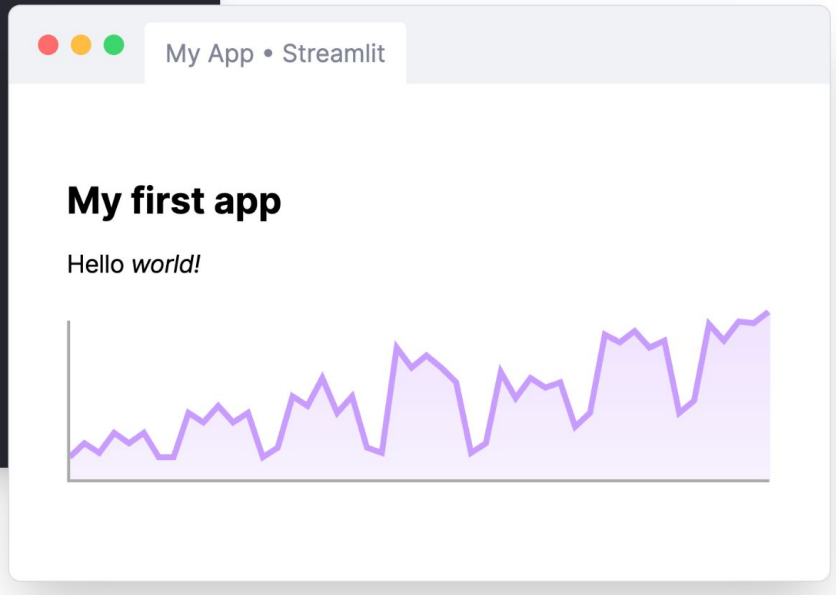
MyApp.py

```
import streamlit as st
import pandas as pd

st.write("""
# My first app
Hello *world!*
""")

df = pd.read_csv("my_data.csv")
st.line_chart(df)
```

Streamlit

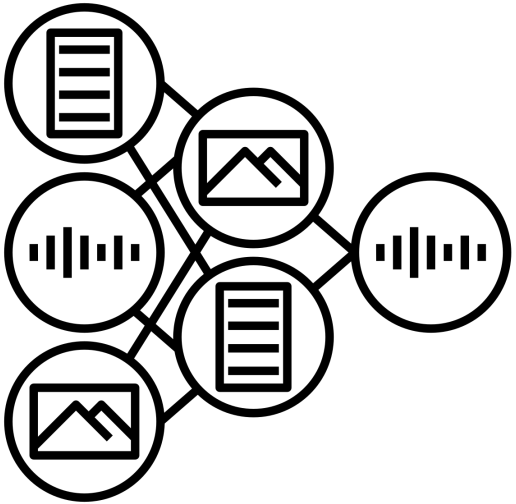


Comparing UI Frameworks

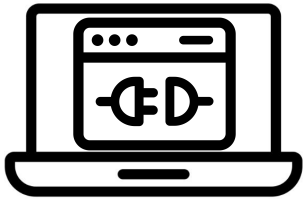
Feature	Streamlit	Gradio	Dash
Speed to Prototype	⚡⚡⚡ (Fastest)	⚡⚡ (Fast)	⚡ (Slower)
Ease of Use	✅✅✅ (Easiest)	✅✅ (Easy)	✅ (More Complex)
Web Dev Knowledge?	No (Pure Python)	No (Mostly Python)	Yes (HTML/CSS helps)

GenAI Models You Can Use

🚫 You don't need to train your own models from scratch

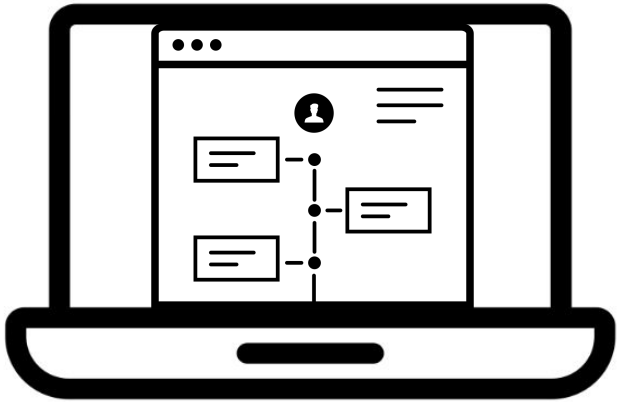


✅ Use Pre-Trained Models via APIs



- GPT-4
- Claude
- Google Gemini
- Open-source models

What Does a Minimal GenAI Stack Look Like?



- ✅ Python
- ✅ Streamlit
- ✅ CSV or text file (dataset)
- ✅ GenAI API
- ✅ .env file or secrets manager

DeepLearning.AI

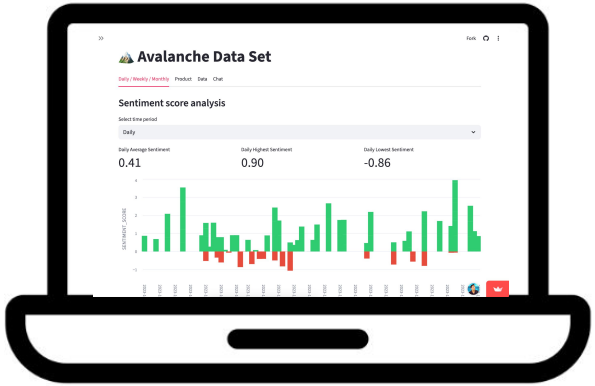
Fast Prototyping of GenAI Apps with Streamlit

Setting up Your Environment

Recommended Project Structure

```
📁 genai-prototype/
    └─ streamlit_app.py              # Main UI and logic
    └─data/
        └─ customer_reviews.csv      # Example input dataset
    └─ .env                           # Stores your API key
    └─ requirements.txt               # Dependencies
```

Setting Up Your Environment

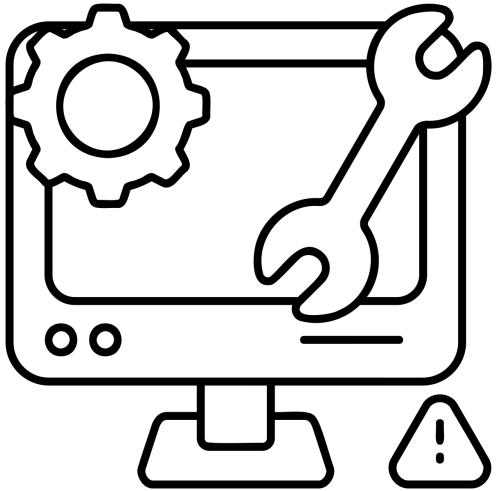


M1/Lesson_02

M1L2V2_starting.py

M1L2V2.py

Troubleshooting API Calls



- API Key Errors 🔑
- Model Name Issues 🤖
- Surprise Charges 💸
- Debugging AI Output 🧠
- Version Control 🐙

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

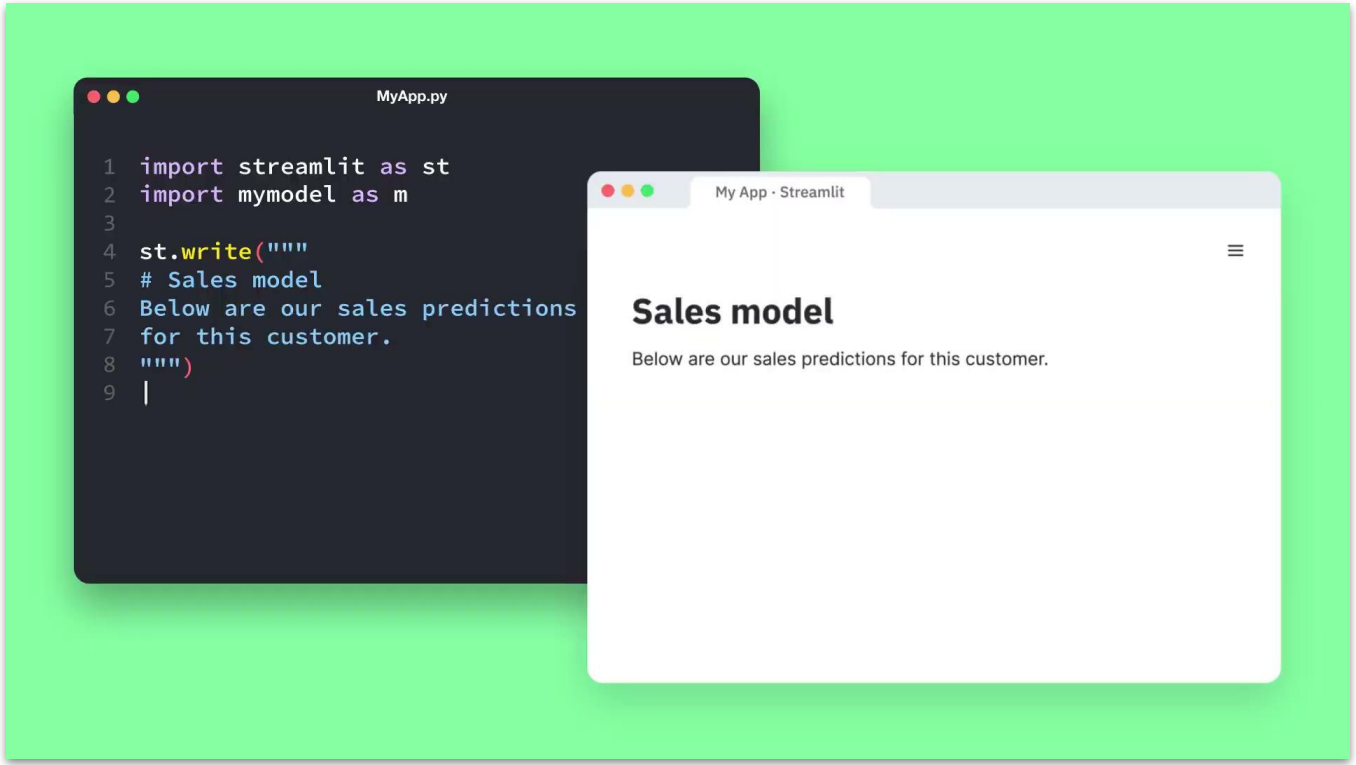
Getting Started with Streamlit

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Making Your First Interactive Streamlit App

Widgets


MyApp.py
1 import streamlit as st
2 import mymodel as m
3
4 st.write("""
5 # Sales model
6 Below are our sales predictions
7 for this customer.
8 """)
9
My App · Streamlit
Sales model
Below are our sales predictions for this customer.

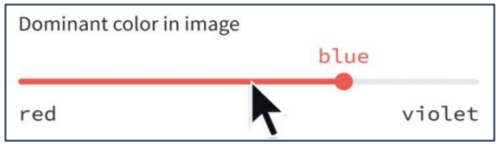
Widgets

st.text_input()

st.button()

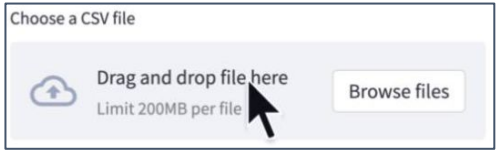
st.selectbox()

st.slider()



st.checkbox()

st.file_uploader()



st.spinner()

https://docs.streamlit.io/develop/api-reference/widgets

Lesson 3

Advanced Streamlit

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Integrating GenAI for Data Handling

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

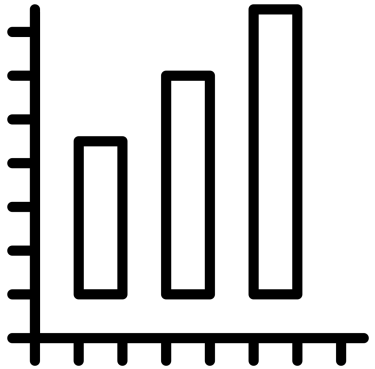
Data Visualization

Streamlit Built-In Functions

Streamlit matplotlib plotly Altair

```
# Streamlit built-in function
st.bar_chart(df["SENTIMENT_SCORE"])

# Streamlit - ready to go
st.line_chart(),
st.area_chart(),
st.scatter_chart()
```

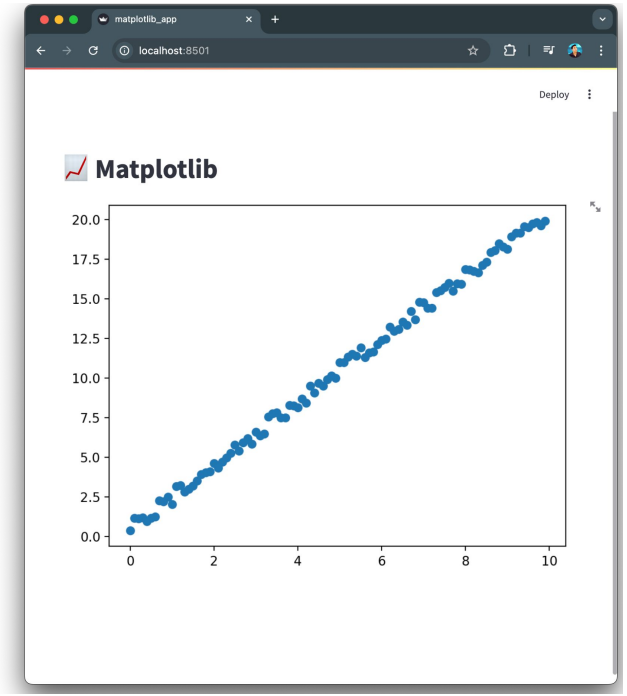


Streamlit Built-In Functions

```
# Streamlit built-in function
df.groupby(“PRODUCT”)["SENTIMENT"].mean()
```

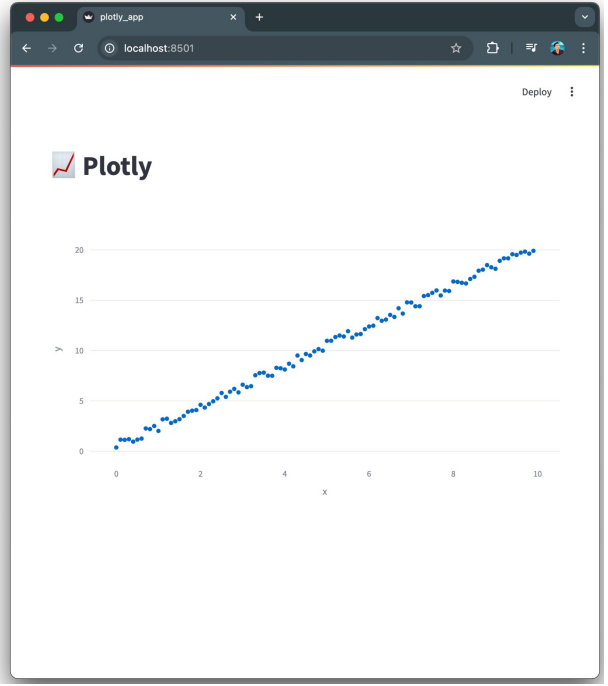
Static Plots with Matplotlib

```
fig, ax = plt.subplots(figsize=(10, 6))
ax.hist(filtered_df["SENTIMENT_SCORE"],
        bins=10, edgecolor='black', alpha=0.7)
ax.set_xlabel('Sentiment Score')
ax.set_ylabel('Frequency')
ax.set_title('Distribution of Sentiment Scores')
st.pyplot(fig)
```



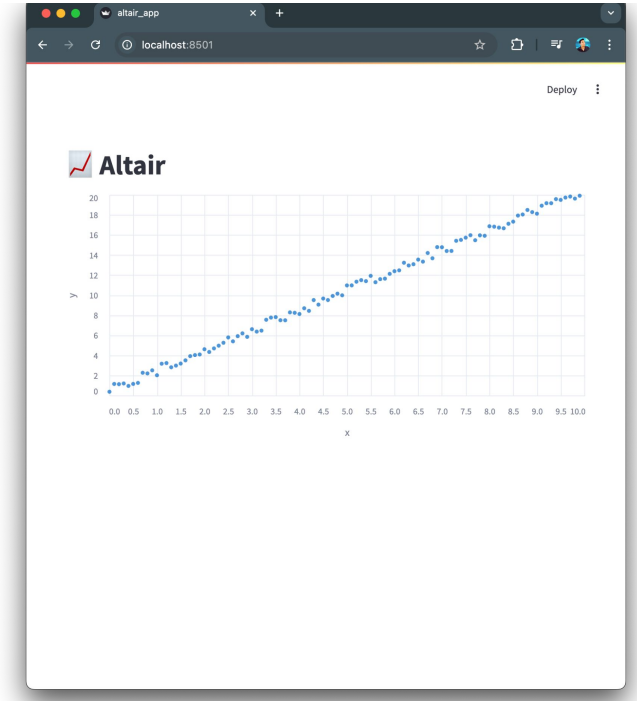
Interactive Visuals with Plotly

```
fig = px.histogram(
        filtered_df,
        x="SENTIMENT_SCORE",
        nbins=10,
        title="Distribution of Sentiment Scores",
        labels={"SENTIMENT_SCORE": "Sentiment Score",
                                   "count": "Frequency"}
      )
fig.update_layout(
    xaxis_title="Sentiment Score",
    yaxis_title="Frequency",
    showlegend=False
)
st.plotly_chart(fig, use_container_width=True)
```



Altair for Decorative Flair

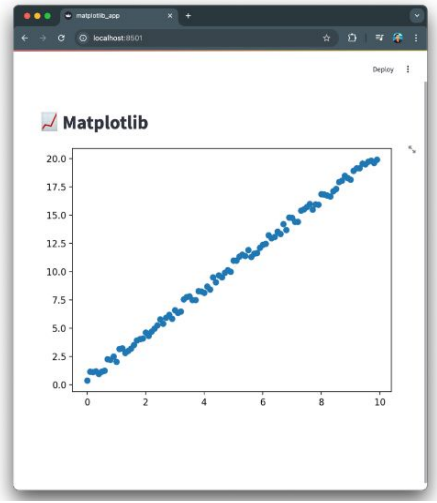
```
chart = alt.Chart(filtered_df).mark_bar().add_selection(
        alt.selection_interval()
    ).encode(
        alt.X("SENTIMENT_SCORE:Q", bin=alt.Bin(maxbins=10),
title="Sentiment Score"),
        alt.Y("count():Q", title="Frequency"),
        tooltip=["count():Q"]
    ).properties(
        width=600,
        height=400,
        title="Distribution of Sentiment Scores"
    )
st.altair_chart(chart, use_container_width=True)
```



Use the Right Tool for the Job

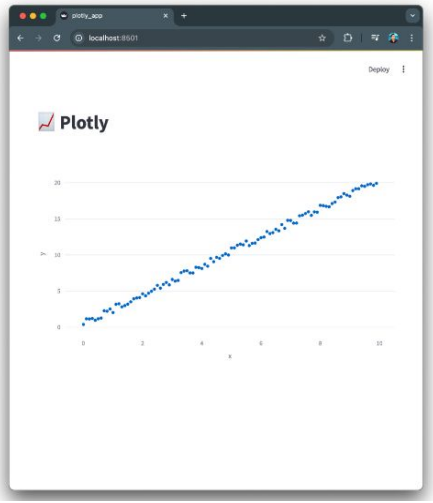
Tool	Best For	Notes
Streamlit st.bar_chart()	Quick feedback	Super fast
Matplotlib	Static plots	Simple and classic
Plotly	Exploring data	Interactive
Altair	More advanced visualizations	Excellent with tidy data

matplotlib



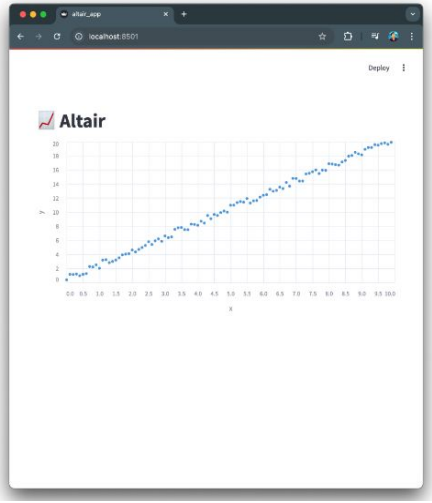
st.pyplot()

plotly



st.plotly_chart()

Altair



st.altair_chart
(chart,
use_container_width=
True)

Streamlit

st.scatter_chart(df)

DeepLearning.AI

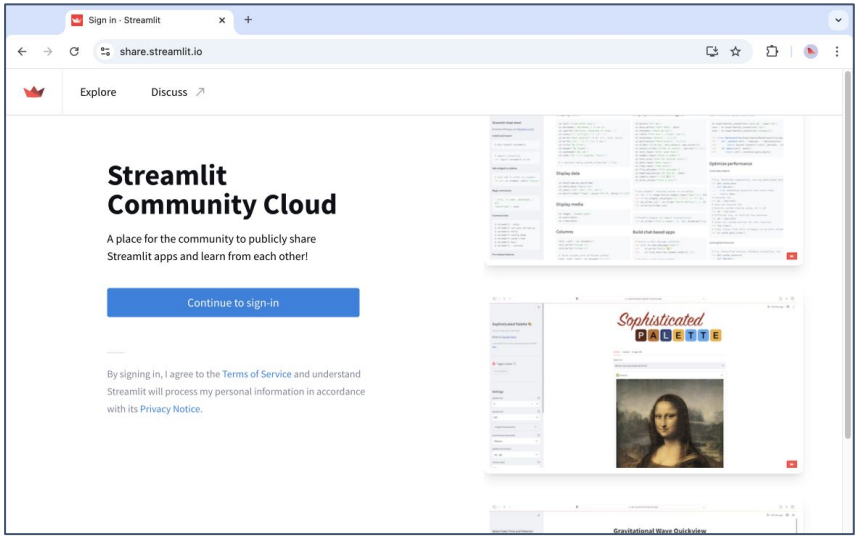
snowflake®

DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Publish Your App Online

Streamlit Community Cloud Basics



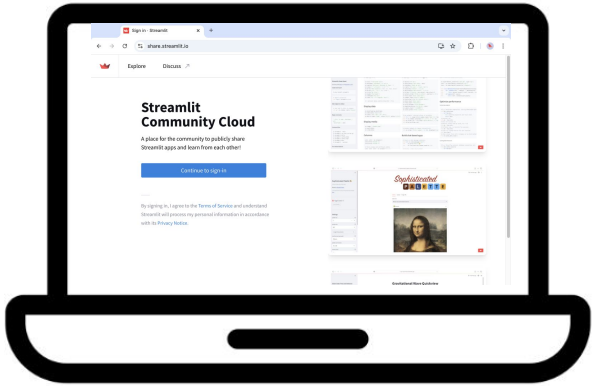
- ✅ Unlimited public apps
- ✅ One private app per account
- ✅ GitHub repository integration
- ✅ Automatic updates
- ✅ Built-in analytics

https://share.streamlit.io/

DeepLearning.AI

snowflake®

Deploying to Streamlit Community Cloud



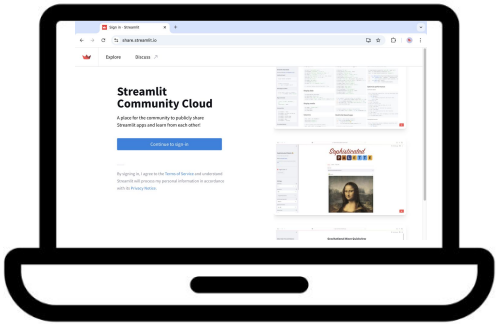
Let's code

M1/Lesson_03/deploy

Deploying to Streamlit Community Cloud

Congratulations!

You've successfully deployed your Streamlit prototype to the cloud and made it accessible to the world!

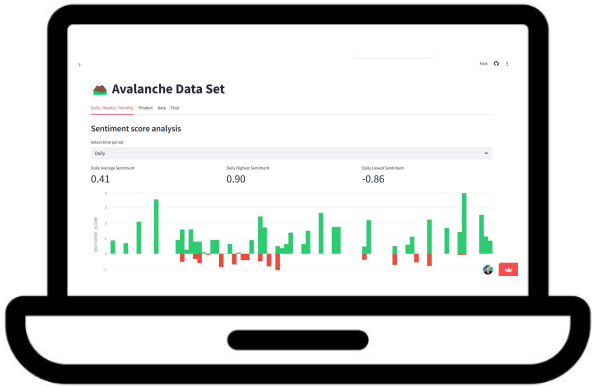


DeepLearning.AI

Fast Prototyping of GenAI Apps with Streamlit

Lab 2 - Sentiment Analysis Dashboard with Gen AI

For This Lab



- Load a dataset with Pandas
- Analyze the reviews sentiments
- Display the dataset
- Visualize the sentiment analysis

M1/Lesson_03/Lab2