Creating a Depression and Anxiety Prescriptions Knowledge-Base with ML for Mental Health Treatment Decision Support

Brightside Health

BREAK THROUGH TECH

August 2024

We're excited to be your Challenge Advisors!

Christine Lee (she/her)
Manager, Data Science
christine@brightside.com

Diane Bernardoni (she/her)
Senior Data Scientist
diane.bernardoni@brightside.com

Miranda Johnson (she/her)
Senior Data Scientist
miranda.johnson@brightside.com

Andrew Norris (he/him)
Data Scientist
andrew.norris@brightside.com

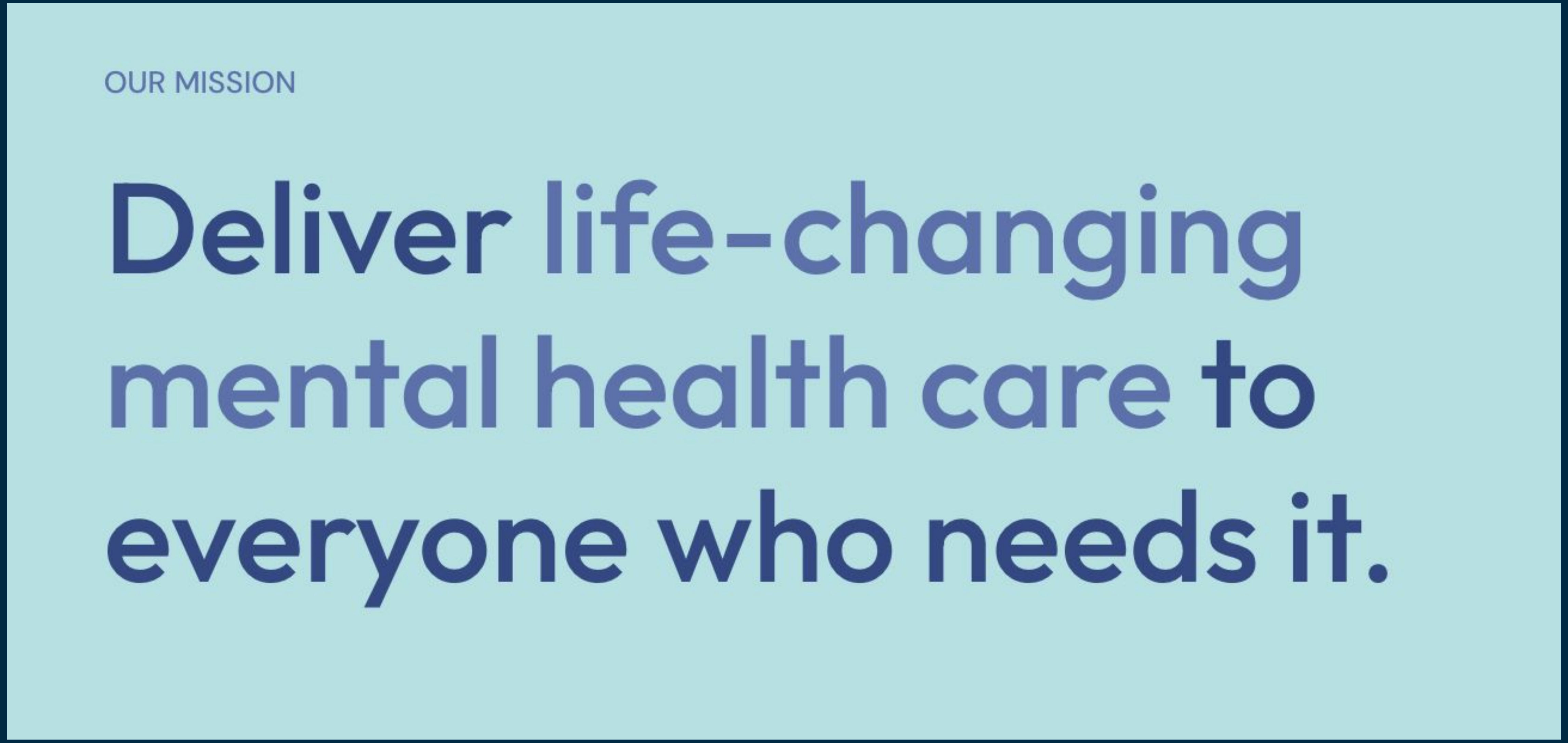
Round of introductions:

Please share your:

- Name
- Preferred pronouns
- College you attend
- What you’re majoring in
- What you’re most excited about learning during AI studio

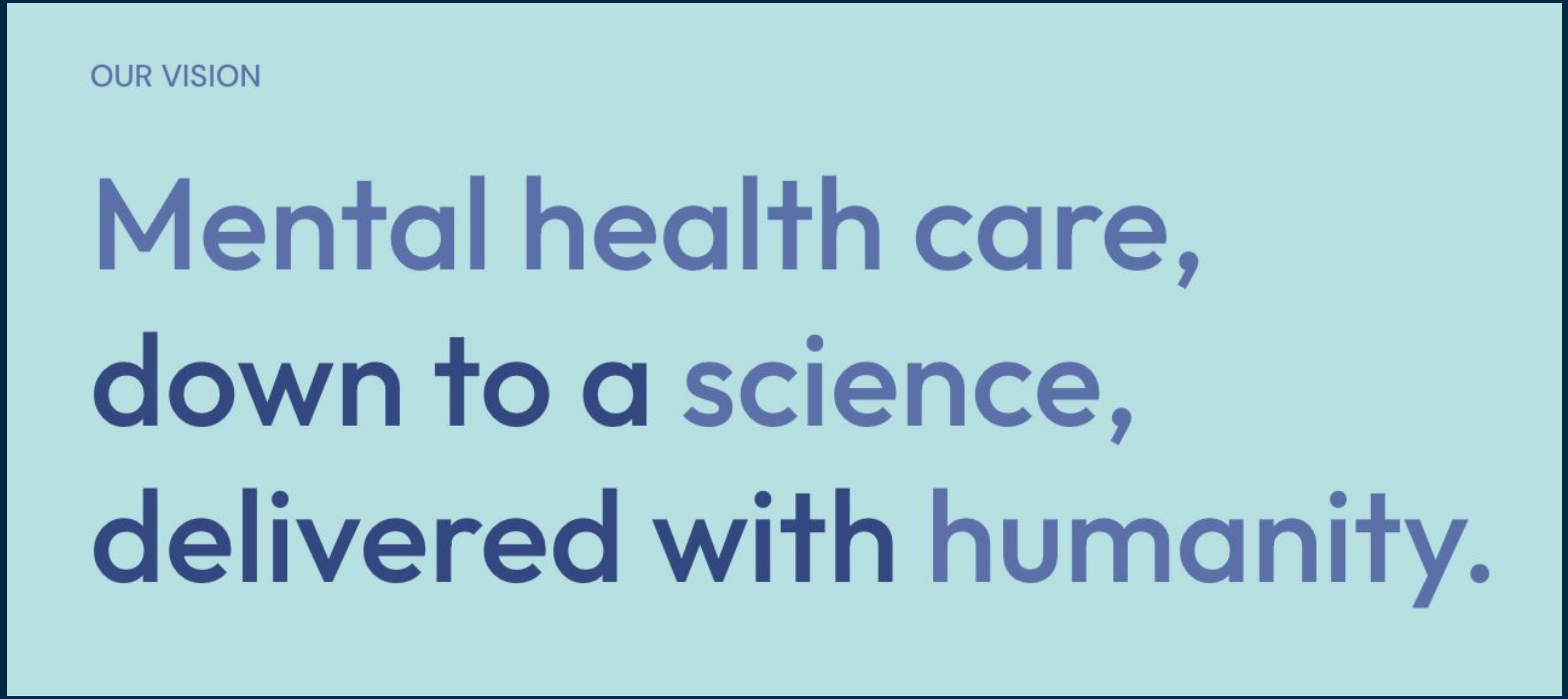
Company overview

Brightside Health delivers life-changing mental health care to people with mild to severe clinical depression, anxiety, and other mood disorders, including those with elevated suicide risk. Powered by proprietary AI, purpose-built technology, and a world-class clinician network, Brightside Health combines precision psychiatry and leading-edge therapeutic techniques to offer timely care and improve patient outcomes across the entire clinical spectrum, affordably and at scale.



OUR MISSION

Deliver life-changing mental health care to everyone who needs it.



OUR VISION

Mental health care, down to a science, delivered with humanity.

What Sets Us Apart

We leverage AI to empower our clinicians with the best tools.

Brightside Platform
Responsive, evidence-based care

Brightside Intelligence
AI-driven care pathways and Clinical Decision Support
– PrecisionRx

We deliver the best outcomes and have published 12 peer-reviewed articles.

Remission at 12 weeks

JMIR Publications Advancing Digital Health & Open Science
BMC
Journal of Clinical Psychopharmacology
Cureus
frontiers

130 million covered lives

aetna
Cigna
United Healthcare
Medicaid
Medicare

AI Studio Challenge Project Overview

CHALLENGE SUMMARY

Create a knowledge graph of up-to-date clinical research papers (provided) related to depression and anxiety

YOUR TEAM'S OBJECTIVE

- Use OpenAI's GPT-4 + API (challenge advisor will provide an API secret key) to extract the named entities in the 3 clinical papers provided
- Determine the relationships between those entities
- Store and visualize the results

DESIRED OUTCOMES

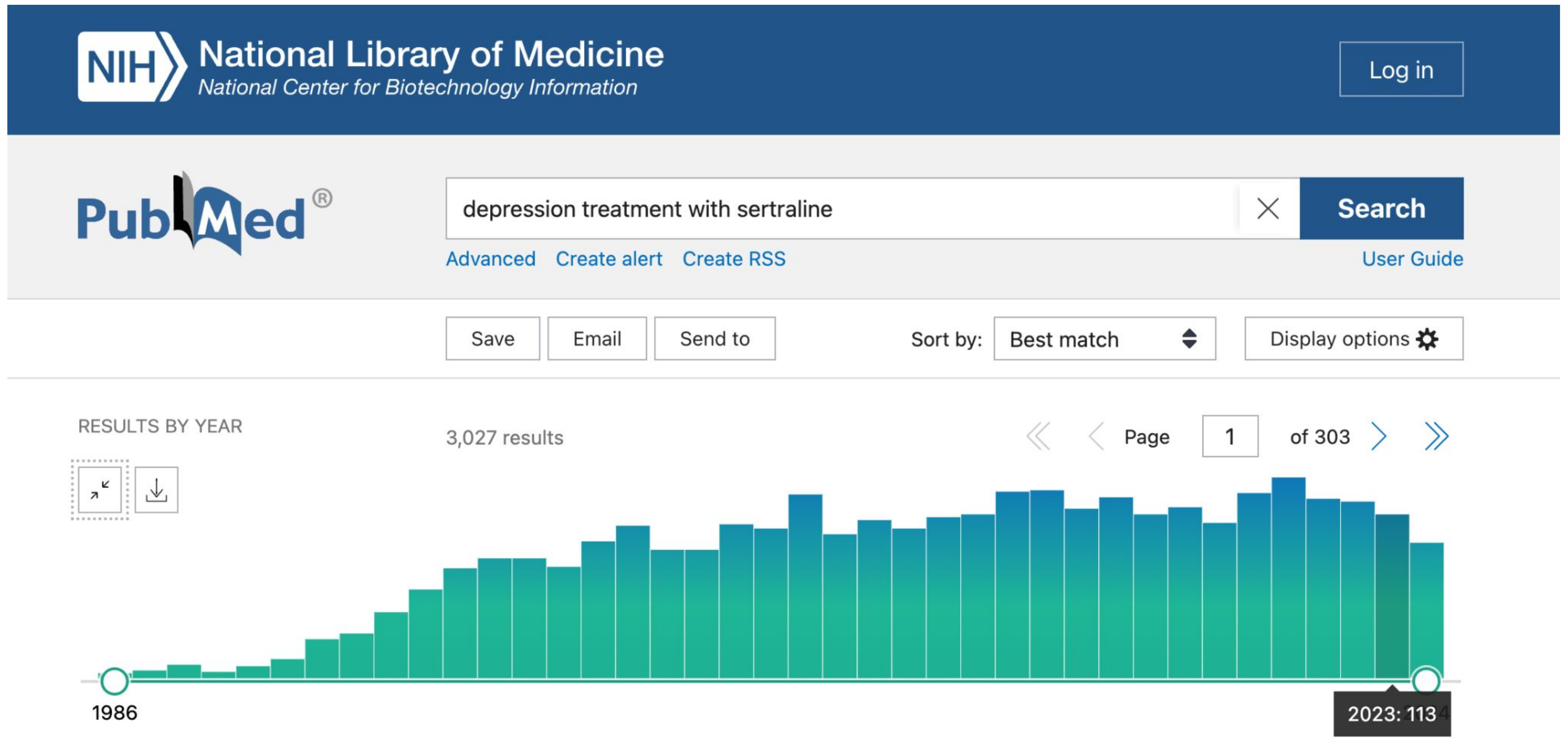
By the end of the challenge, you should be able to

- show a visualization and explanation of the knowledge graph created
- share your final code repository that the challenge advisor is able to run themselves

A stretch goal would be to add 1-2 more papers and update your knowledge graph

Business context

- Clinicians rely on clinical research and protocols to determine treatments for depression and anxiety.
- Brightside aims to offer Clinical Decision Support that aligns with this workflow.
- Our goal is to enable clinicians to search and find evidence-based treatments on our platform.
- The vast amount of research, including 113 articles on sertraline in 2023 alone, is overwhelming for any individual to stay updated on.
- Proof of concept: Can we distill this vast knowledge into a knowledge graph?



depression treatment with sertraline

3,027 results

What is a Knowledge Graph?

What is a Knowledge Graph?

- A knowledge graph is a structured representation of knowledge, where entities and their relationships are organized in a graph format.
- It helps in organizing information in a way that both humans and machines can easily understand and process.

Entities, Nodes, Edges & Relationships

- Entities: Real-world objects or concepts.
- Nodes: Graphical representations of entities.
- Edges: Connections between nodes.
- Relationships: How entities are related (e.g., "Spock is a character in Star Trek, which is of the genre Science Fiction, which Star Wars is as well").

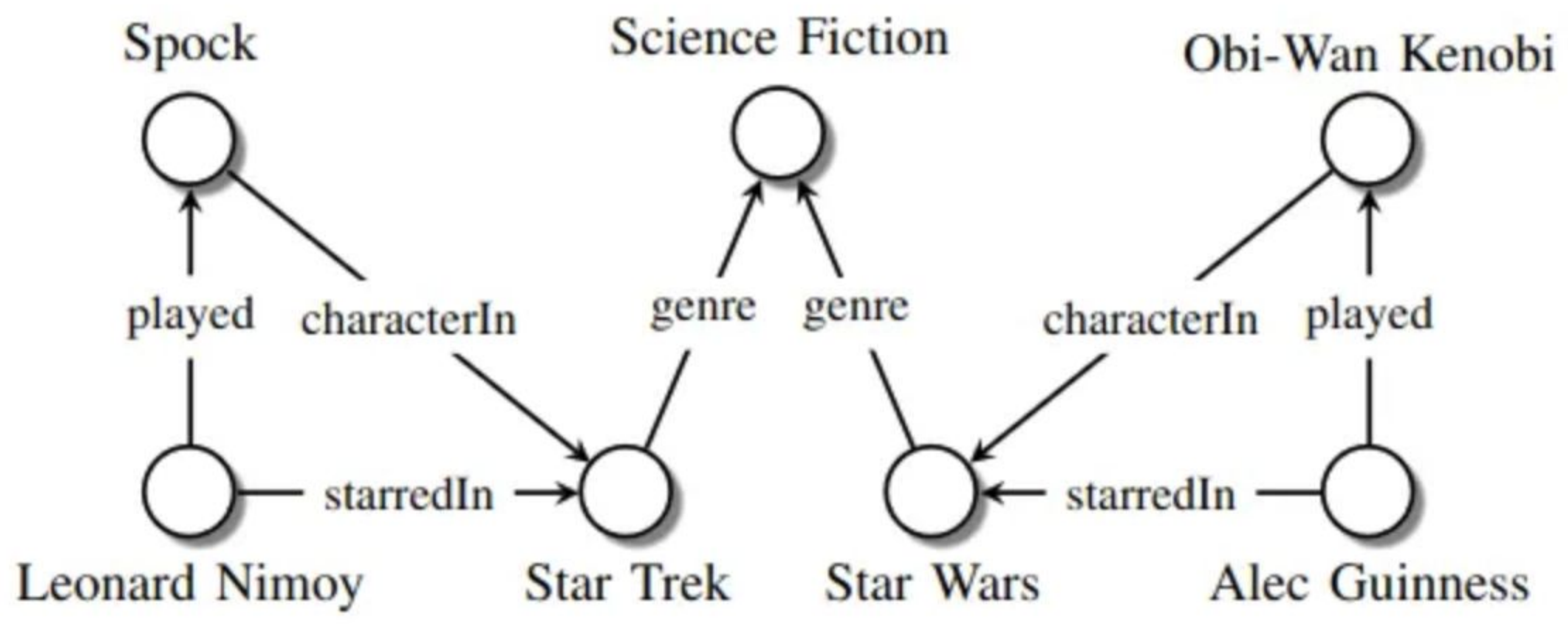


Fig. 1. Sample knowledge graph. Nodes represent entities, edge labels represent types of relations, edges represent existing relationships.

Source: https://ieeexplore.ieee.org/document/7358050

Knowledge Graph Methods: Determining Relationships

Simple Methods

- Manual Curation: Experts manually add accurate relationships.
- Rule-based Systems: Use predefined rules to infer relationships (e.g., "X is the father of Y").

Advanced Methods

- Natural Language Processing (NLP): Extracts relationships from text using context.
- Machine Learning: Predicts relationships with trained models.
- Graph Neural Networks (GNNs): Captures complex patterns by analyzing the graph structure.

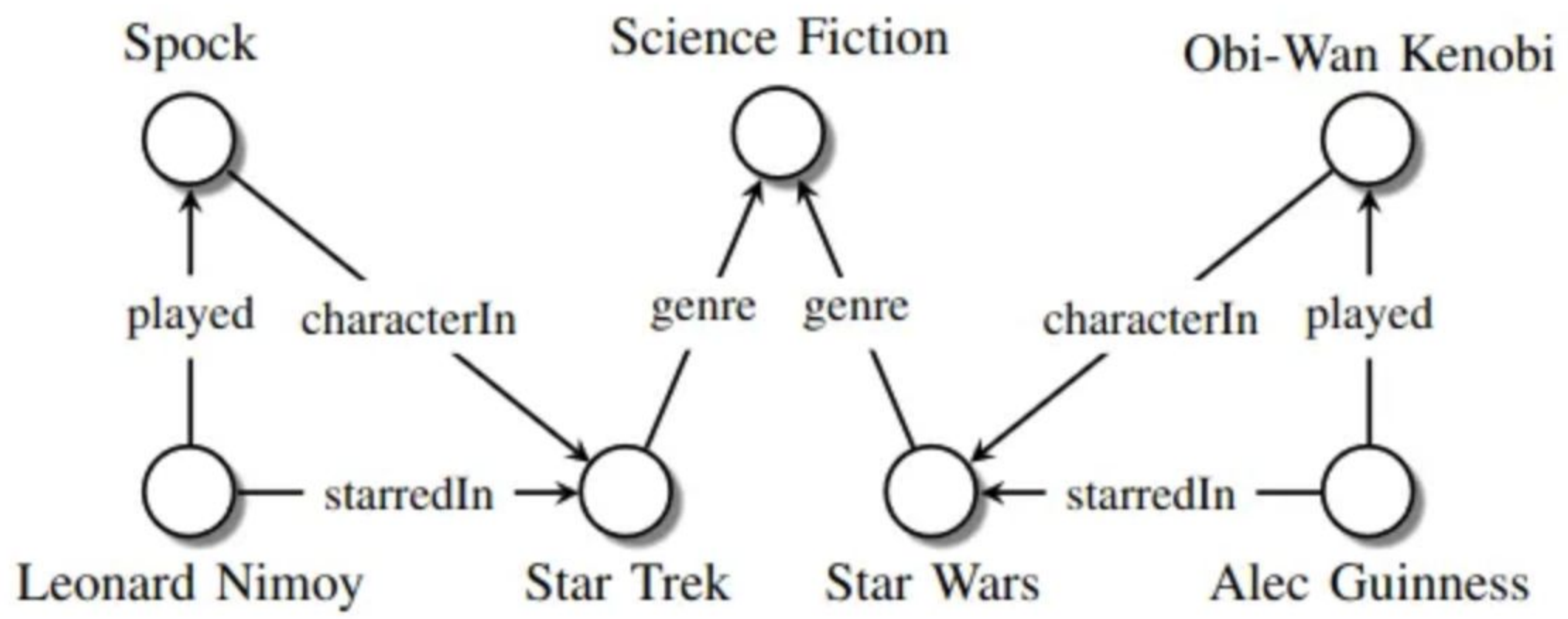


Fig. 1. Sample knowledge graph. Nodes represent entities, edge labels represent types of relations, edges represent existing relationships.

Source: https://ieeexplore.ieee.org/document/7358050

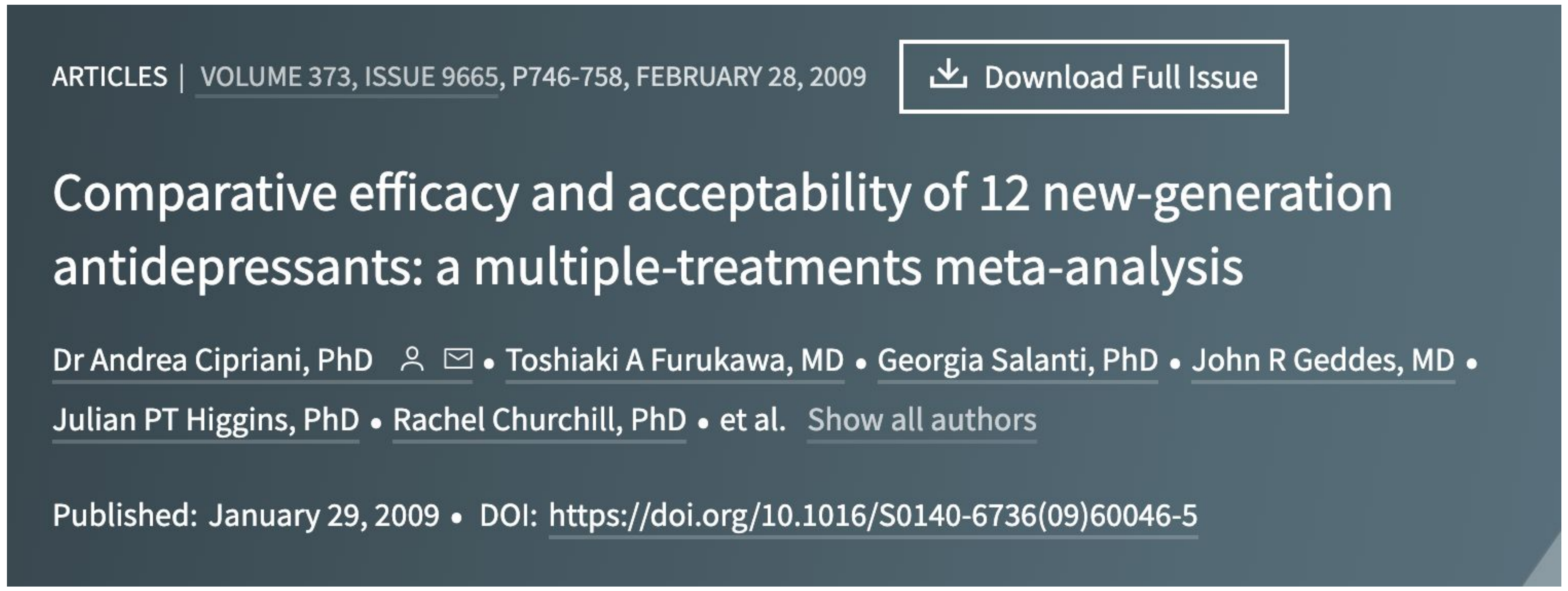
Large Language Models

Some things to keep in mind if this is your first time working on a project involving LLMs:

- **Pre-trained Models**
  Start with pre-trained models (e.g., GPT, BERT).
- **Preprocessing**
  Perform text preprocessing (e.g., tokenization, normalization).
- **Fine-Tuning**
  Fine-tune the model on your specific dataset.
- **Evaluation**
  Use metrics like BLEU, ROUGE, and human evaluation.
- **Deployment**
  Plan for integration and scalability.

Data overview

We will provide 3 clinical papers related to depression, antidepressants and treatment outcomes as PDFs



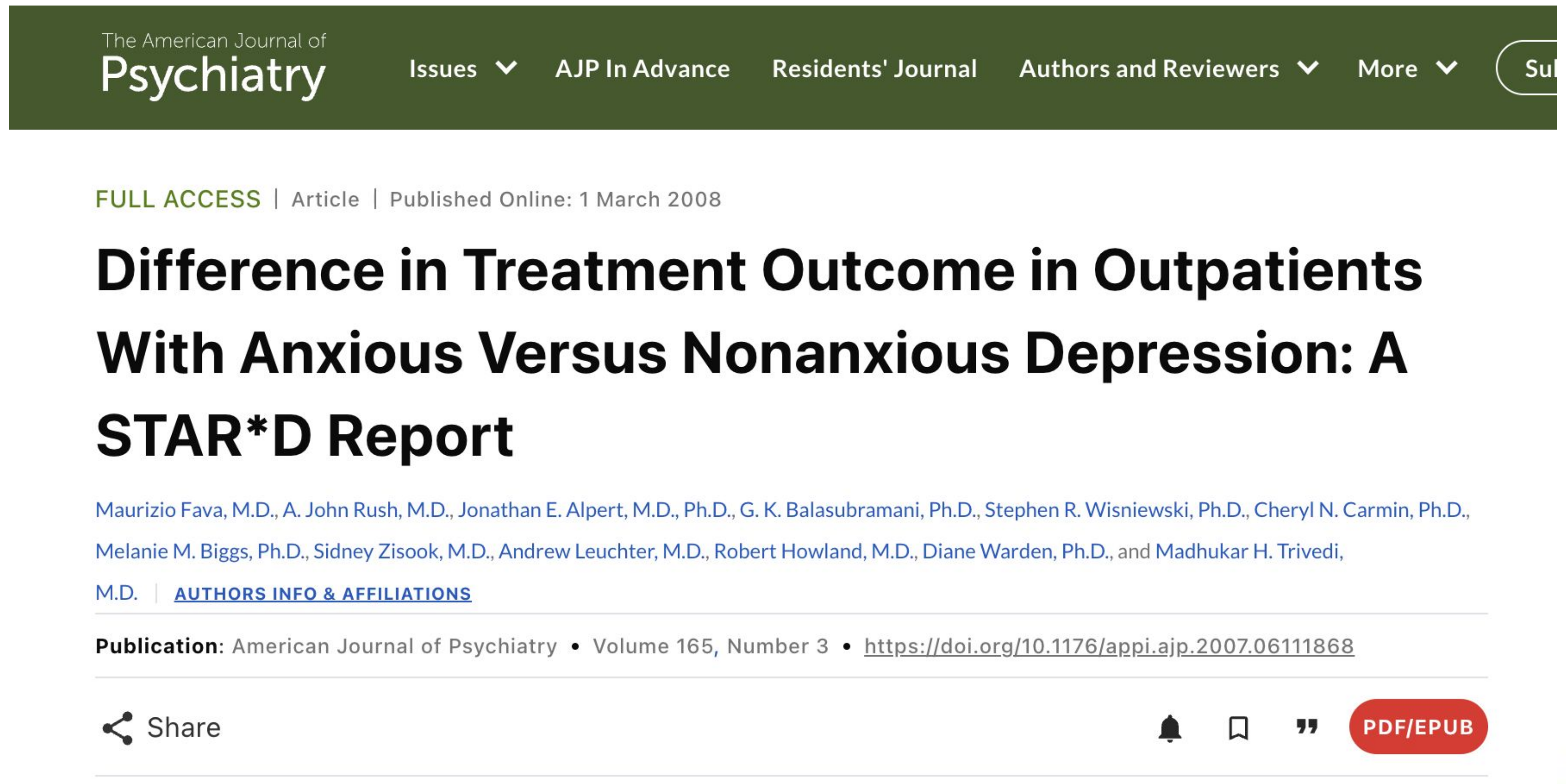
ARTICLES | VOLUME 373, ISSUE 9665, P746-758, FEBRUARY 28, 2009

Download Full Issue

Comparative efficacy and acceptability of 12 new-generation antidepressants: a multiple-treatments meta-analysis

Dr Andrea Cipriani, PhD • Toshiaki A Furukawa, MD • Georgia Salanti, PhD • John R Geddes, MD • Julian PT Higgins, PhD • Rachel Churchill, PhD • et al. Show all authors

Published: January 29, 2009 • DOI: https://doi.org/10.1016/S0140-6736(09)60046-5



The American Journal of Psychiatry

Issues | AJP In Advance | Residents' Journal | Authors and Reviewers | More

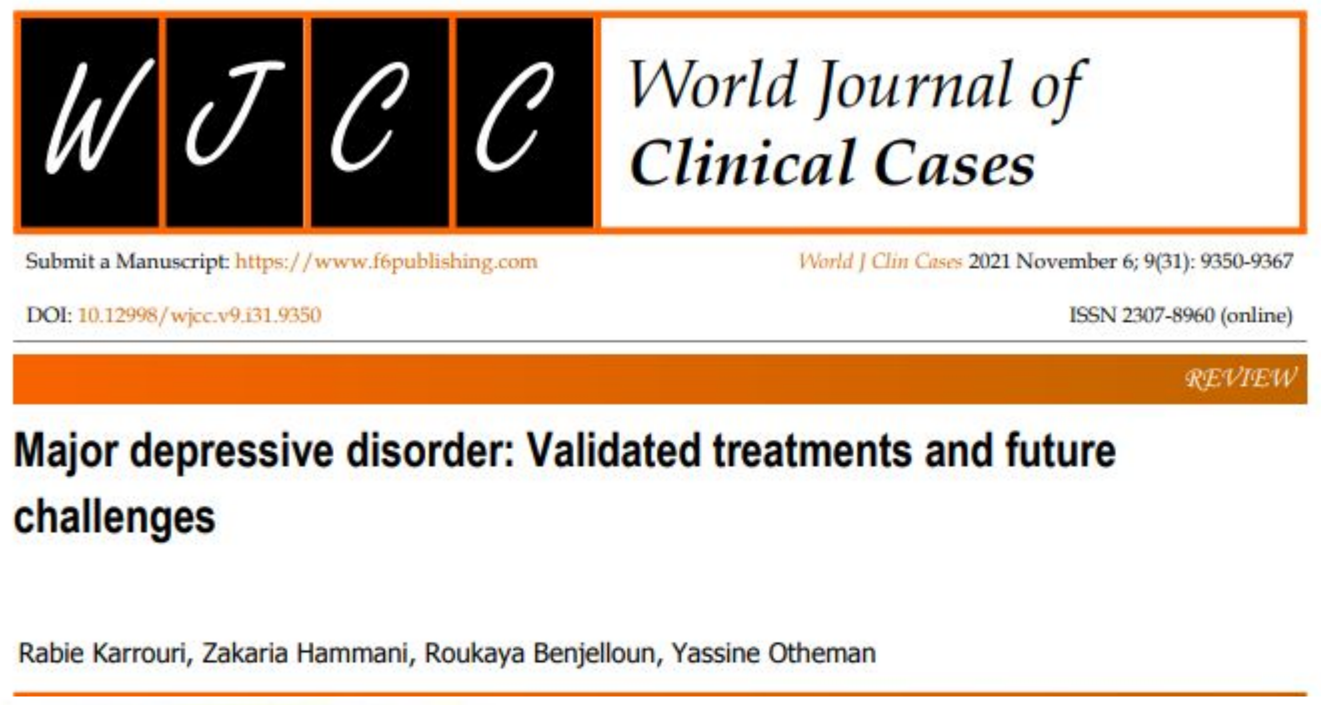
FULL ACCESS | Article | Published Online: 1 March 2008

Difference in Treatment Outcome in Outpatients With Anxious Versus Nonanxious Depression: A STAR*D Report

Maurizio Fava, M.D., A. John Rush, M.D., Jonathan E. Alpert, M.D., Ph.D., G. K. Balasubramani, Ph.D., Stephen R. Wisniewski, Ph.D., Cheryl N. Carmin, Ph.D., Melanie M. Biggs, Ph.D., Sidney Zisook, M.D., Andrew Leuchter, M.D., Robert Howland, M.D., Diane Warden, Ph.D., and Madhukar H. Trivedi, M.D. | AUTHORS INFO & AFFILIATIONS

Publication: American Journal of Psychiatry • Volume 165, Number 3 • https://doi.org/10.1176/appi.ajp.2007.06111868

Share | PDF/EPUB



WJCC World Journal of Clinical Cases

Submit a Manuscript: https://www.f6publishing.com *World J Clin Cases* 2021 November 6; 9(31): 9350-9367

DOI: 10.12998/wjcc.v9.i31.9350 ISSN 2307-8960 (online)

REVIEW

Major depressive disorder: Validated treatments and future challenges

Rabie Karrouri, Zakaria Hammani, Roukaya Benjelloun, Yassine Otheman

Python libraries

- OpenAI Python library
  - Please use the gpt-4o-mini model (we are monitoring usage based on your key)
  - Challenge Advisor will provide API secret key

Suggest Dev and PM tools

- The end “product” is a GitHub repo and code hand-off to the Challenge Advisor
  - Recommend maintaining a GitHub repo
  - Share with your challenge advisor
  - **HOWEVER, YOUR API KEY SHOULD NOT BE SHARED IN THE REPO CODE!**
- You can write all your code in one shared jupyter notebook, however, we recommend at the end, your functions should be in scripts and the jupyter notebook acts as the “visualization layer”
- Your Challenge Advisor will be checking in and monitoring your progress of the following Milestones

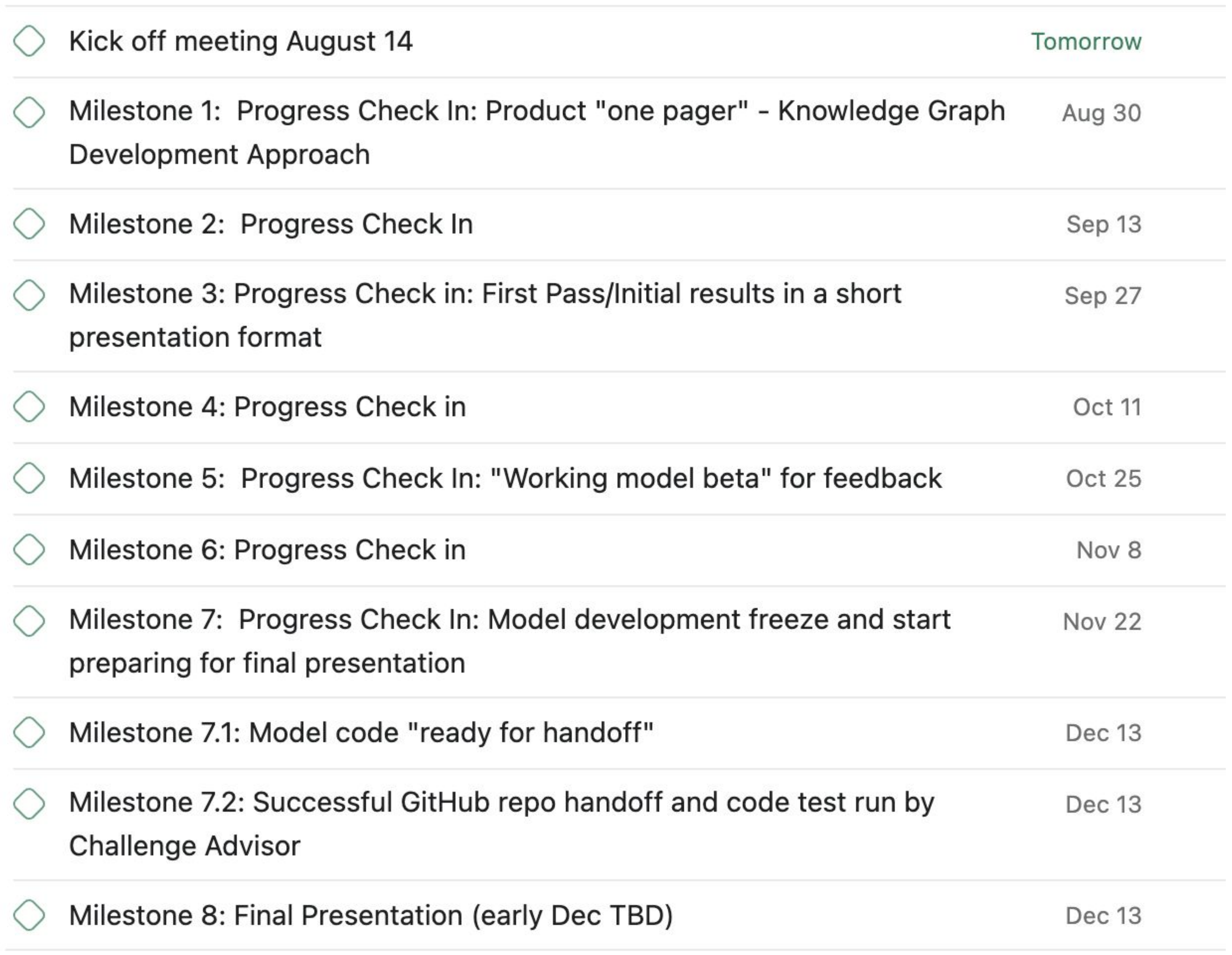
Helpful resources

2 helpful read-throughs on creating a knowledge graph

https://medium.com/neo4j/turn-a-harry-potter-book-into-a-knowledge-graph-ffc1c45afcc8

https://medium.com/google-cloud/augmenting-gemini-1-0-pro-with-knowledge-graphs-via-langchain-bacc2804c423

Project milestones and timeline



Kick off meeting August 14	Tomorrow
Milestone 1: Progress Check In: Product "one pager" - Knowledge Graph Development Approach	Aug 30
Milestone 2: Progress Check In	Sep 13
Milestone 3: Progress Check in: First Pass/Initial results in a short presentation format	Sep 27
Milestone 4: Progress Check in	Oct 11
Milestone 5: Progress Check In: "Working model beta" for feedback	Oct 25
Milestone 6: Progress Check in	Nov 8
Milestone 7: Progress Check In: Model development freeze and start preparing for final presentation	Nov 22
Milestone 7.1: Model code "ready for handoff"	Dec 13
Milestone 7.2: Successful GitHub repo handoff and code test run by Challenge Advisor	Dec 13
Milestone 8: Final Presentation (early Dec TBD)	Dec 13

Milestone 1: What is a product one pager? Why is it important?

Expected Sections

1. *Problem Statement*: Defines the problem the model aims to solve
2. *Objectives:* Measurable objectives to guide model development
3. *Data Requirements:* Sources, types, quality, availability
4. *Methodology*: Development approach, methods, techniques
5. *Evaluation Metrics:* How model is to be measured, ensuring project objectives are met
6. *Deliverables:* Expected outputs and documentation
7. *Limitations/Risks & Mitigations:* Potential challenges and how they may be addressed
8. *Stakeholders:* List key stakeholders and communication

Importance of a One Pager

- *Conciseness*: A one pager distills the entire project plan into a brief, easily digestible format, making it accessible to all stakeholders.
- *Alignment*: Ensure that everyone involved in the project is aligned on the goals, methods, and expectations from the outset.
- *Efficiency*: Provides a quick reference guide throughout the project, helping to keep the team focused and on track.
- *Communication*: Facilitates clear communication between technical and non-technical stakeholders, reducing the risk of misunderstandings.

Milestone 3: Short Initial Feedback presentation

Presentation requirements

- *Overview of Progress*: Summarize completed work and key milestones; Highlight preliminary results.
- *Key Insights:* Share early findings and potential impacts; Discuss unexpected results or challenges.
- *Next Steps:* Outline upcoming tasks and focus areas; Identify needed decisions or input.
- *Questions & Discussion:* Invite feedback and ensure alignment; Address concerns and suggestions.

Why we prefer a slide deck format

- *Clarity:* Visually organizes information for easy understanding.
- *Focus:* Highlights key points without overwhelming details.
- *Engagement:* Keeps the presentation concise and engaging across various stakeholders/audiences.
- *Alignment:* Facilitates clear communication and decision-making.

Questions?

What questions do you have?

Anything I can help clarify?

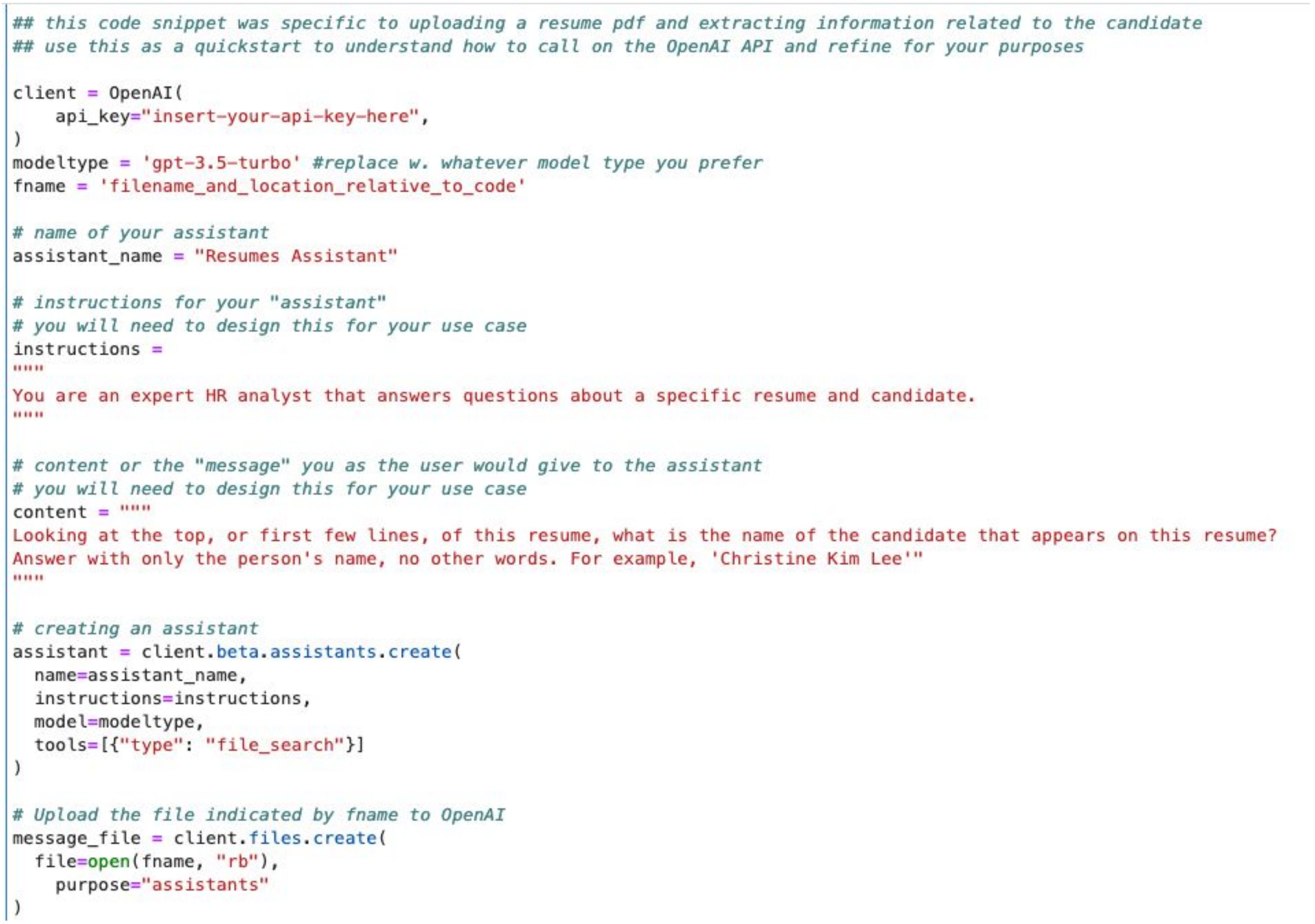
What are you most excited about?

Anything you're unsure about?

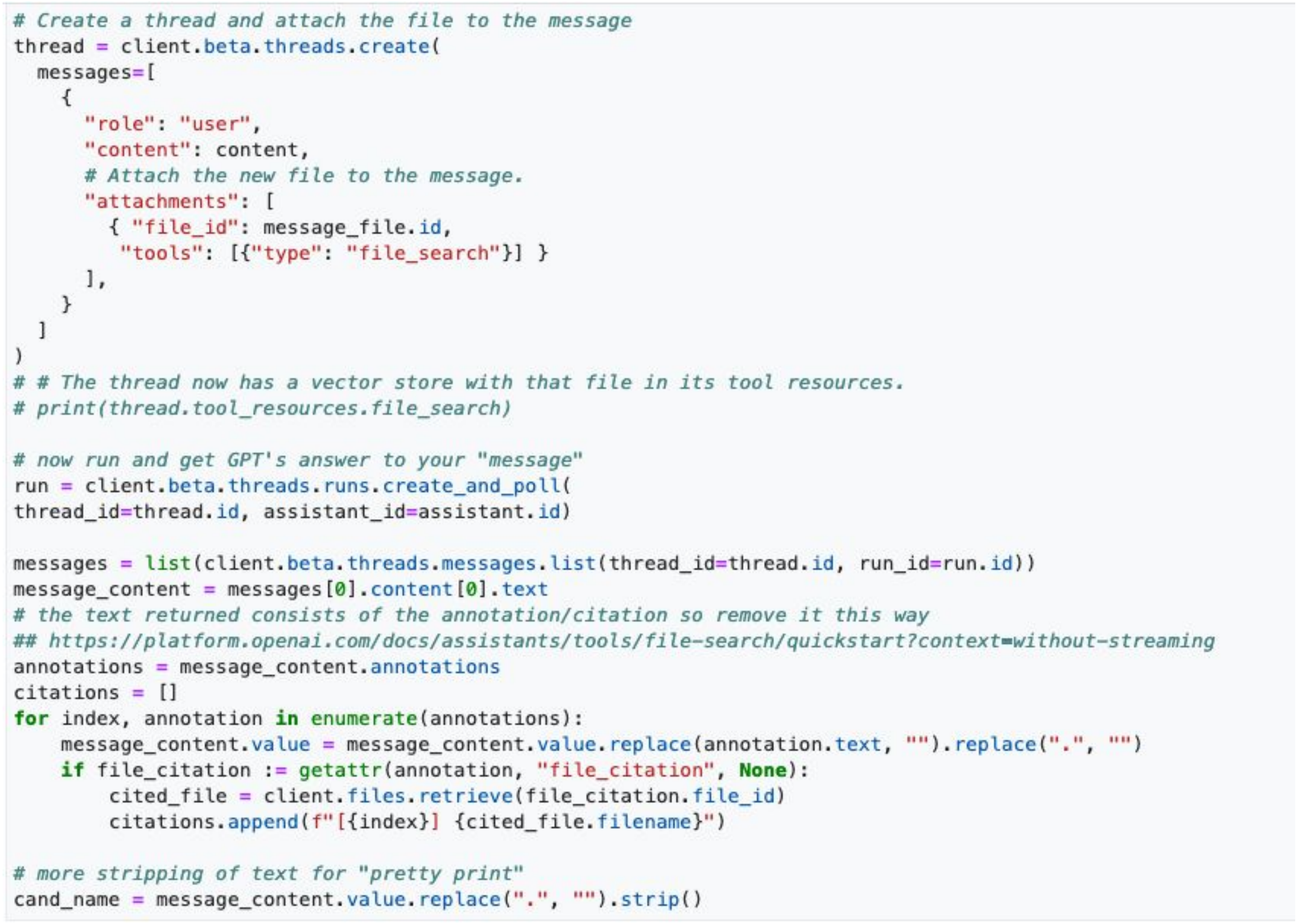
Appendix A. GPT4 API Code Snippet w/ response in a json format
(feel free to ask your Challenge Advisor for the code snippet in Slack)

```
from openai import OpenAI

client = OpenAI(
            api_key="insert_your_api_key_here",
        )

prompt_llm = """You are a customer support agent reading through customer provided free text comments as to why they did not show up to their appointment.
You'll be provided a list of 100 comments separated by a semi-colon as the delimiter. For example, the following are 3 different comments separated by "; "
"Work overtime; I had it down on my schedule as Monday instead of Sunday ; Family emergency ;"

You will extract the most common themes of the comments and respond in json format, with the theme as the key and 3 example comments exemplifying the theme in a list.
For example, you could respond as
{'work conflict': ['I had to work', 'I work nights', 'I forgot I worked early today'],
'personal conflict': [''Had a family matter come up, apologies.', ''My mom was sick. She has stage 4 lung cancer. I was helping her.', 'I had to run my son to the hospital']}
"""

prompt = "insert your message / what you want the LLM to answer"

chat_completion = client.chat.completions.create(
                messages=[
                    {
                        "role": "system",
                        "content": prompt_llm,
                    },
                    {
                        "role": "user",
                        "content": prompt,
                    }
                ],
                model="gpt-4-1106-preview",
                response_format = {"type": "json_object"}
            )


json_answer = json.loads(chat_completion.choices[0].message.content)
json_answer
```

Appendix B. GPT4 API Code Snippet w/ Assistants + PDF upload
(feel free to ask your Challenge Advisor for the code snippet in Slack)



```python
## this code snippet was specific to uploading a resume pdf and extracting information related to the candidate
## use this as a quickstart to understand how to call on the OpenAI API and refine for your purposes

client = OpenAI(
    api_key="insert-your-api-key-here",
)
modeltype = 'gpt-3.5-turbo' #replace w. whatever model type you prefer
fname = 'filename_and_location_relative_to_code'

# name of your assistant
assistant_name = "Resumes Assistant"

# instructions for your "assistant"
# you will need to design this for your use case
instructions =
"""
You are an expert HR analyst that answers questions about a specific resume and candidate.
"""

# content or the "message" you as the user would give to the assistant
# you will need to design this for your use case
content = """
Looking at the top, or first few lines, of this resume, what is the name of the candidate that appears on this resume?
Answer with only the person's name, no other words. For example, 'Christine Kim Lee'"
"""

# creating an assistant
assistant = client.beta.assistants.create(
  name=assistant_name,
  instructions=instructions,
  model=modeltype,
  tools=[{"type": "file_search"}]
)

# Upload the file indicated by fname to OpenAI
message_file = client.files.create(
  file=open(fname, "rb"),
    purpose="assistants"
)
```



```python
# Create a thread and attach the file to the message
thread = client.beta.threads.create(
  messages=[
    {
      "role": "user",
      "content": content,
      # Attach the new file to the message.
      "attachments": [
        { "file_id": message_file.id,
         "tools": [{"type": "file_search"}] }
      ],
    }
  ]
)
# # The thread now has a vector store with that file in its tool resources.
# print(thread.tool_resources.file_search)

# now run and get GPT's answer to your "message"
run = client.beta.threads.runs.create_and_poll(
thread_id=thread.id, assistant_id=assistant.id)

messages = list(client.beta.threads.messages.list(thread_id=thread.id, run_id=run.id))
message_content = messages[0].content[0].text
# the text returned consists of the annotation/citation so remove it this way
## https://platform.openai.com/docs/assistants/tools/file-search/quickstart?context=without-streaming
annotations = message_content.annotations
citations = []
for index, annotation in enumerate(annotations):
    message_content.value = message_content.value.replace(annotation.text, "").replace(".", "")
    if file_citation := getattr(annotation, "file_citation", None):
        cited_file = client.files.retrieve(file_citation.file_id)
        citations.append(f"[{index}] {cited_file.filename}")

# more stripping of text for "pretty print"
cand_name = message_content.value.replace(".", "").strip()
```