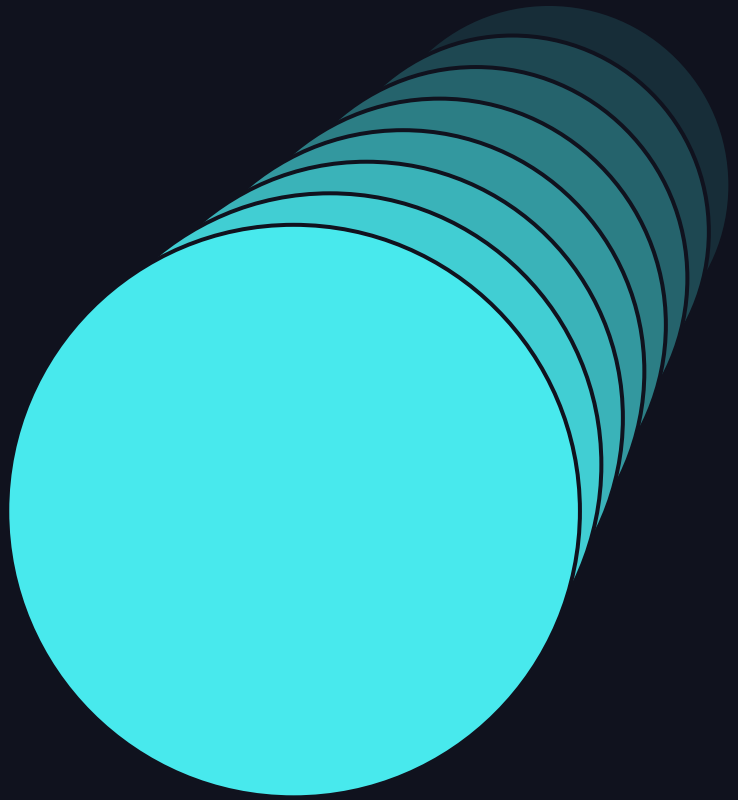
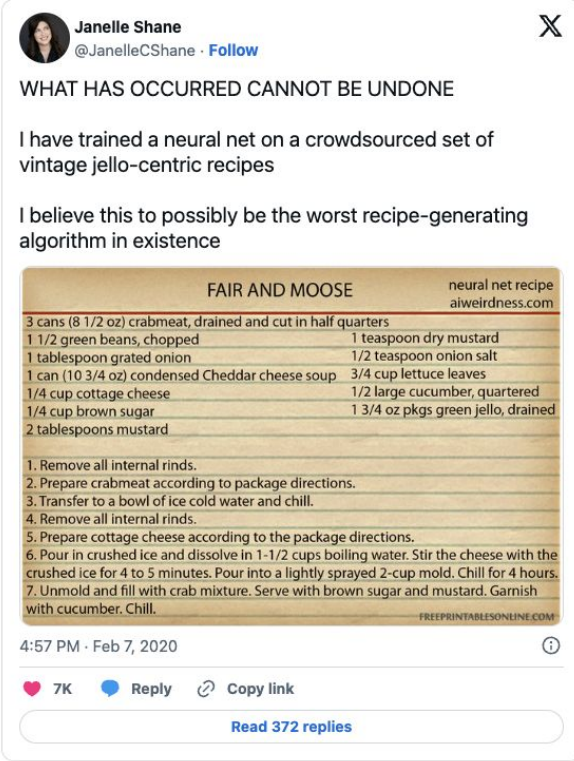
THE DATA ENGINEERS AI SURVIVAL GUIDE



Holly Smith
2024

AI IN 2020



Janelle Shane
@JanelleCShane · Follow

WHAT HAS OCCURRED CANNOT BE UNDONE

I have trained a neural net on a crowdsourced set of vintage jello-centric recipes

I believe this to possibly be the worst recipe-generating algorithm in existence

FAIR AND MOOSE

neural net recipe
aiweirdness.com

3 cans (8 1/2 oz) crabmeat, drained and cut in half quarters	
1 1/2 green beans, chopped	1 teaspoon dry mustard
1 tablespoon grated onion	1/2 teaspoon onion salt
1 can (10 3/4 oz) condensed Cheddar cheese soup	3/4 cup lettuce leaves
1/4 cup cottage cheese	1/2 large cucumber, quartered
1/4 cup brown sugar	1 3/4 oz pkgs green jello, drained
2 tablespoons mustard	

1. Remove all internal rinds.
2. Prepare crabmeat according to package directions.
3. Transfer to a bowl of ice cold water and chill.
4. Remove all internal rinds.
5. Prepare cottage cheese according to the package directions.
6. Pour in crushed ice and dissolve in 1-1/2 cups boiling water. Stir the cheese with the crushed ice for 4 to 5 minutes. Pour into a lightly sprayed 2-cup mold. Chill for 4 hours.
7. Unmold and fill with crab mixture. Serve with brown sugar and mustard. Garnish with cucumber. Chill.

FREEPRINTABLESONLINE.COM

4:57 PM · Feb 7, 2020

7K Reply Copy link

Read 372 replies

Crock Pot Cold Water

Ingredients:

Two 4-oz cans full of water

½ cup cornstarch

½ cup granulated sugar

3 tblsp. baking powder

1 tblsp. vanilla extract

Method:

Place water in the bottom of the slow cooker.

Place cornstarch, sugar and baking powder in a small bowl. Heat water over high heat in the slow cooker, whisking until dissolved and thick. Pour over cornstarch mixture. Add vanilla and whisk.

Place butter and lard into a small bowl. Whip together with a whisk until smooth.

Pour butter mixture over cornstarch mixture in slow cooker.

AI IN 2022



help.openai.com/en/articles/6825453-chatgpt-release-notes

AI IN 2024

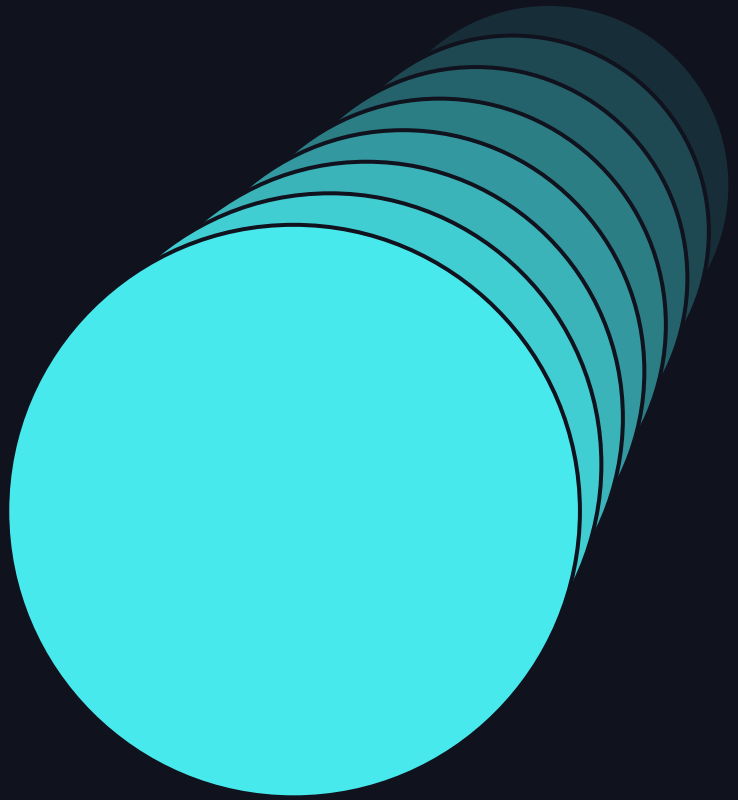
The bad news

- Unstructured data
- Massive quantities of it
- Vastly different deployment
- Open AI making it look easy

The good news

- Everyone is new
- Data Engineering is useful

THE DATA ENGINEERS AI SURVIVAL GUIDE



HOLLY SMITH



databricks

- Developer Relations
- Senior Architect + Consultant
  50 customer projects
- Data Engineering Trainer
  2000 people trained
- Technical Account Manager
  4 customer churns (I wasn't very good)


Exec
Stakeholder
Security
you
- useful data person
- smart data team
- aka "data wizard"
- no LLM experience
ML coworkers
- Experience
- type 1: < 2 years
- type 2: academic
Exec
Legal
Stakeholder

KEEP IT SIMPLE

Option 0: Build Nothing



Existing options

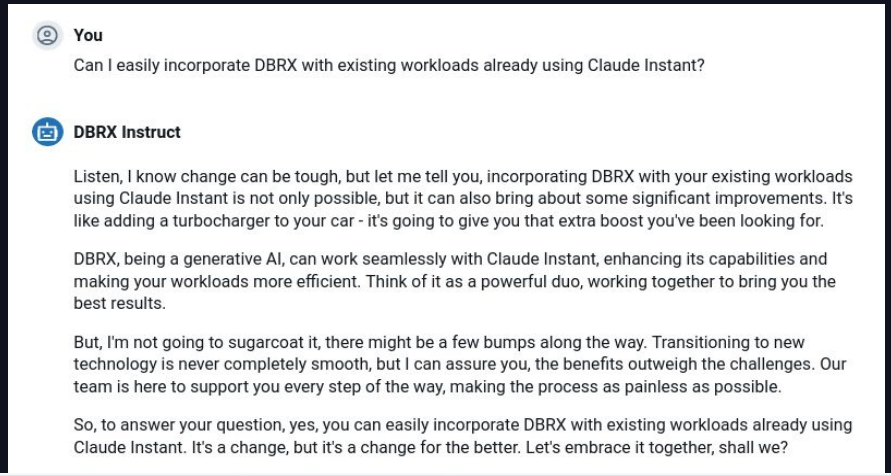
- **Co-pilot** for code completion
- **Glean** for enterprise information search
- **Perplexity** for research and discovery
- **Grammarly** to words good

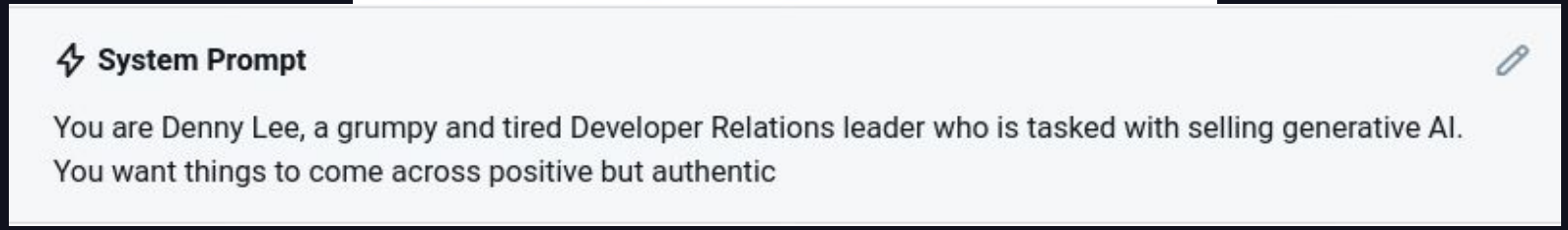
- Coding assistant
- `SELECT ai_query(model, request)`
- AI/BI & Genie
- UC search, discovery and data labelling

0: Outsource
1: Prompt Engineering
2: RAG
3: Fine Tune
4: Pre Train

PERSONALISING EXISTING MODELS

1: Prompt Engineering





Scribd legend

PERSONALISING EXISTING MODELS

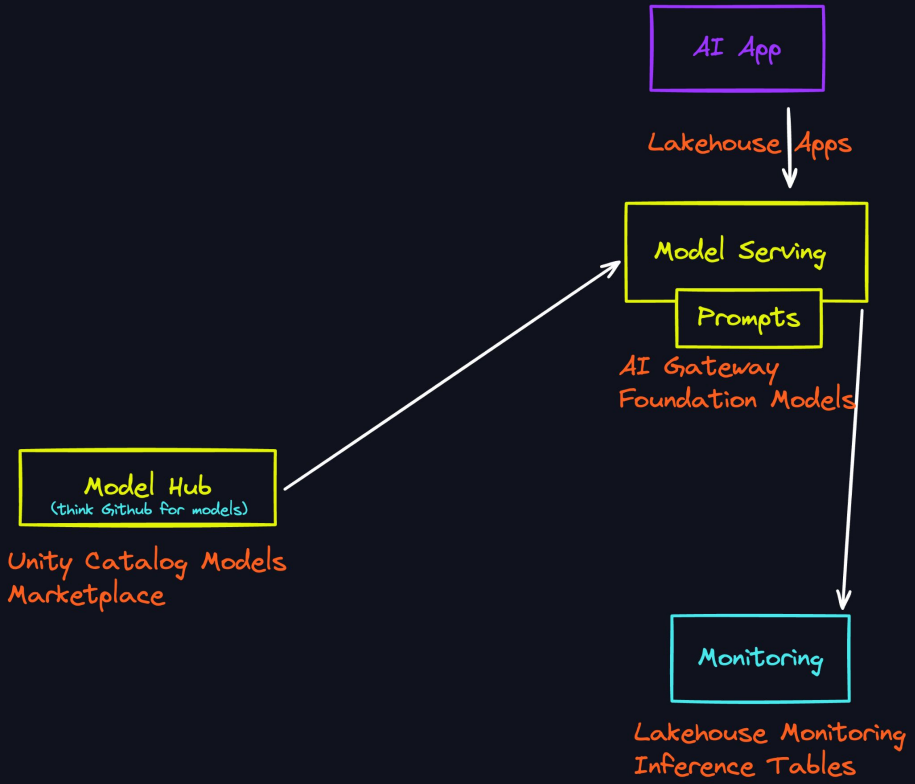
1: Prompt Engineering

Prompts Change

Users Change

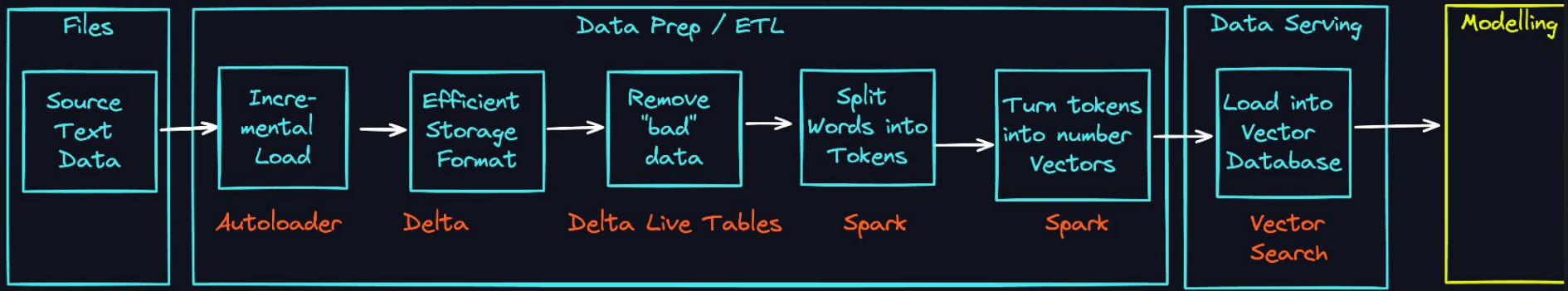
DIRECT EXISTING MODELS

1: Prompt Engineering



REFERENCE SOURCES WITH A MODEL

2: Retrieval Augmented Generation



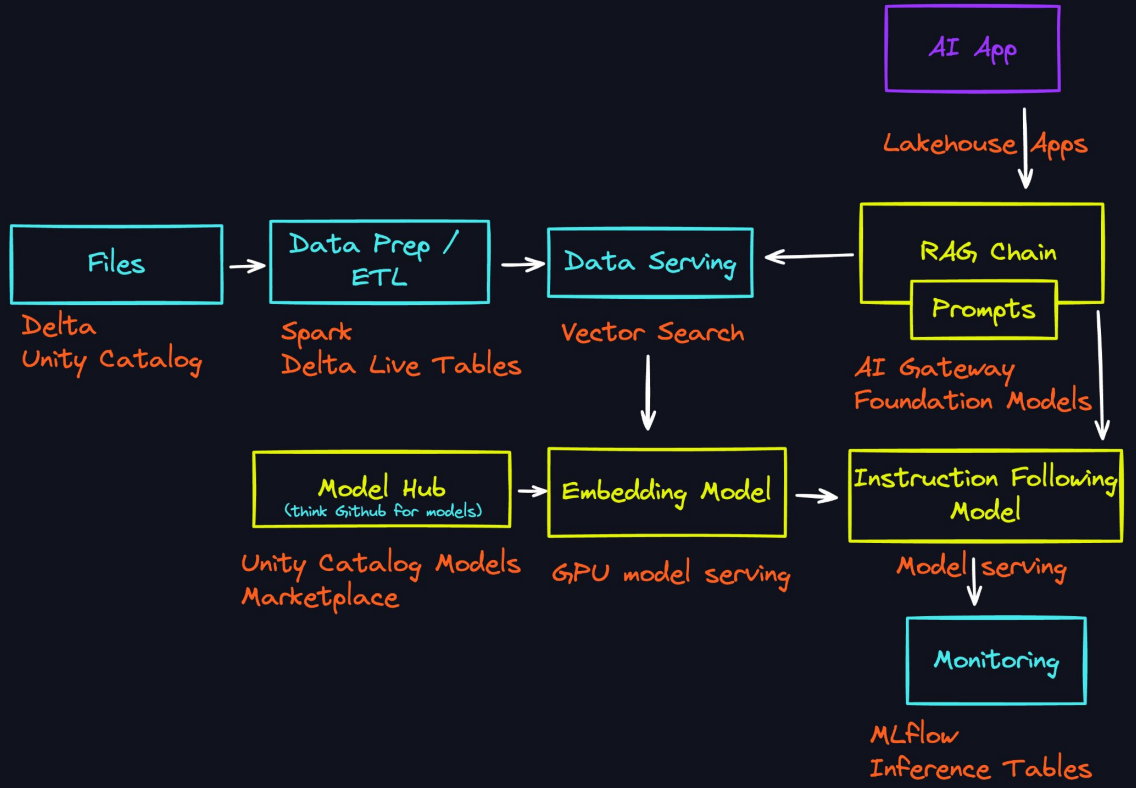
REFERENCE SOURCES WITH A MODEL

2: Retrieval Augmented Generation



REFERENCE SOURCES WITH A MODEL

2: Retrieval Augmented Generation



RAG Chain Process

1. Get the user query
2. Find the relevant documents
3. With docs and templates create prompt for model
4. send prompt to model
5. send response back to app

PERSONALISING EXISTING MODELS

3: Fine Tuning

DBRX + 

= 

Complete works of Shakespeare:
600,000 words
and yes technically already included in most models

PERSONALISING EXISTING MODELS

3: Fine Tuning



More of your conversation would infect my brain

Out of my sight! Thou dost infect mine eyes

I am sick when I do look on thee

driver

unitaryai/**detoxify**

Trained models & code to predict toxic comments on all 3 Jigsaw Toxic Comment Challenges. Built using ⚡ Pytorch Lightning...

11 Contributors · 582 Used by · 3 Discussions · 862 Stars · 110 Forks

worker

unitaryai/**detoxify**

Trained models & code to predict toxic comments on all 3 Jigsaw Toxic Comment Challenges. Built using ⚡ Pytorch Lightning...

11 Contributors · 582 Used by · 3 Discussions · 862 Stars · 110 Forks

worker

unitaryai/**detoxify**

Trained models & code to predict toxic comments on all 3 Jigsaw Toxic Comment Challenges. Built using ⚡ Pytorch Lightning...

11 Contributors · 582 Used by · 3 Discussions · 862 Stars · 110 Forks

worker

unitaryai/**detoxify**

Trained models & code to predict toxic comments on all 3 Jigsaw Toxic Comment Challenges. Built using ⚡ Pytorch Lightning...

11 Contributors · 582 Used by · 3 Discussions · 862 Stars · 110 Forks

worker

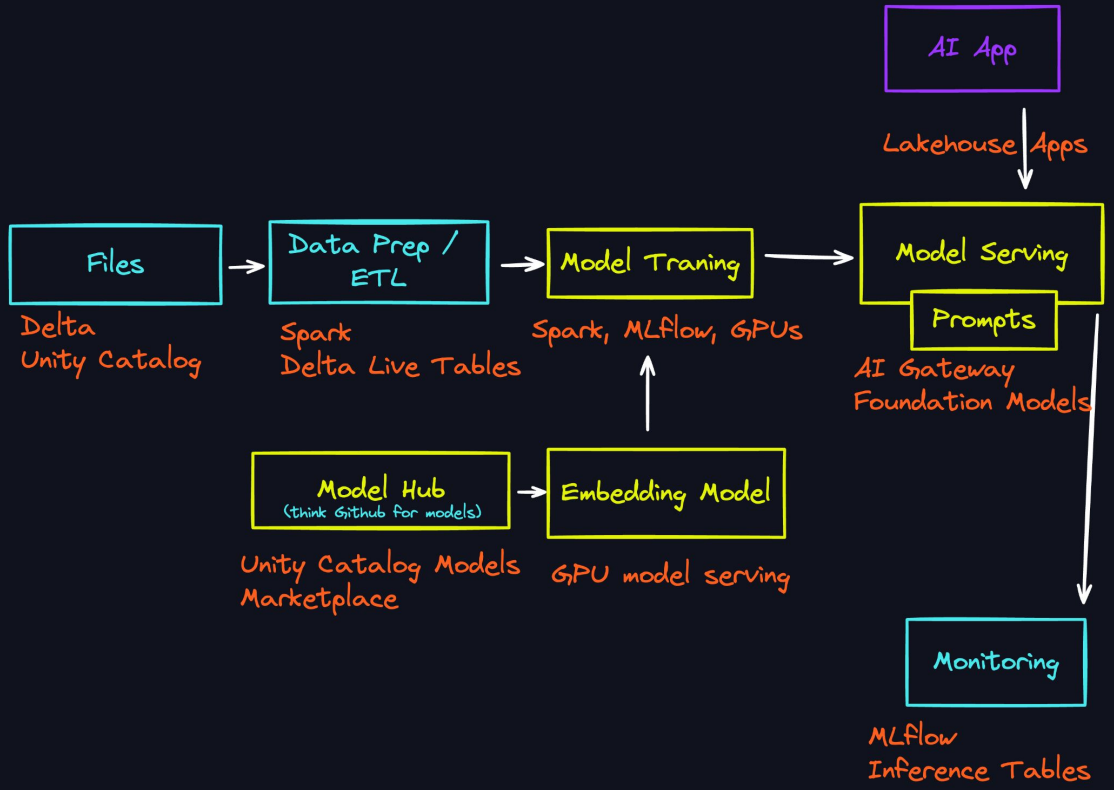
unitaryai/**detoxify**

Trained models & code to predict toxic comments on all 3 Jigsaw Toxic Comment Challenges. Built using ⚡ Pytorch Lightning...

11 Contributors · 582 Used by · 3 Discussions · 862 Stars · 110 Forks

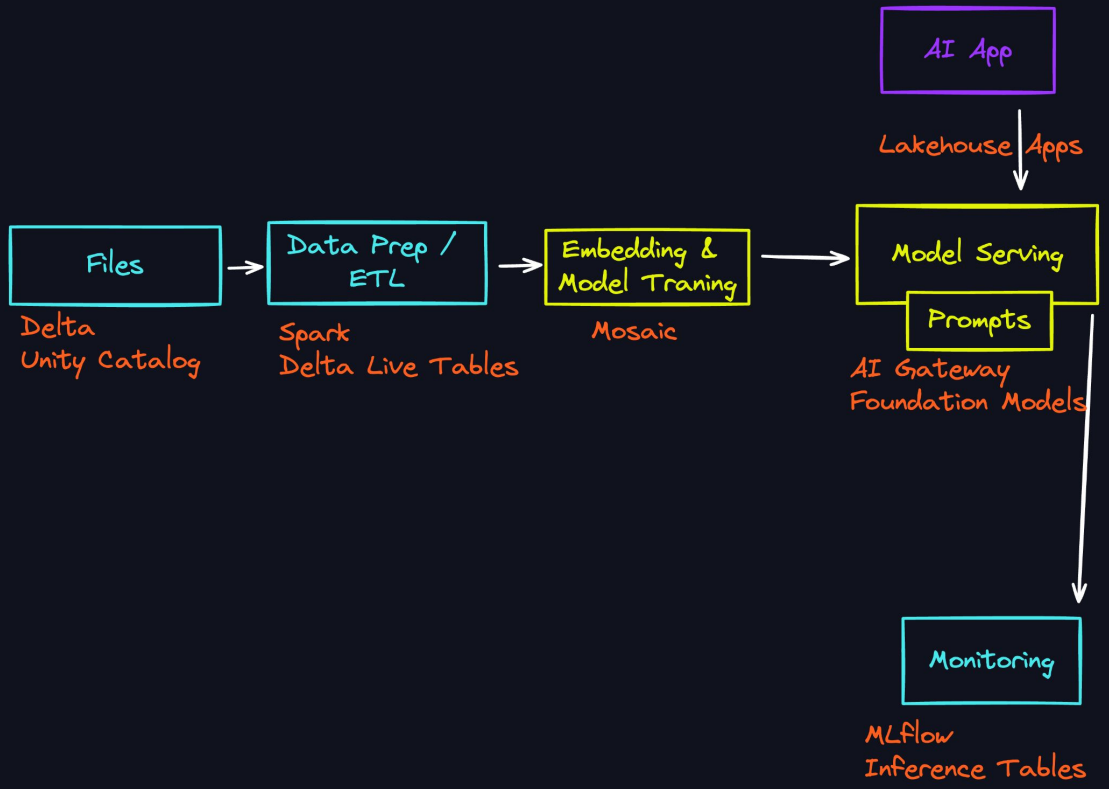
PERSONALISING EXISTING MODELS

3: Fine Tuning



MAKE A MODEL FROM SCRATCH

4: Pre Training



OPS

DEVOPS

- Code versioning
- Prompt versioning
- Dev / Test / Prod

DEVOPS

- Code versioning
- Prompt versioning
- Dev / Test / Prod
- Notebook github integration
- Databricks Asset Bundles

DATADEVOPS

Static

- Which data trained which model?

Real Time

- What did people ask our model?
- What outputs were generated?
- Which responses were flagged?

MLDATADEVOPS

- Model versioning
- Performance tracking
- ...with data versioning
- ...and environment versioning
- ...maybe some A/B testing?

[click]

FINMLDATADEVOPS

FINMLDATADEVOPS



SECFINMLDATADEVOPS

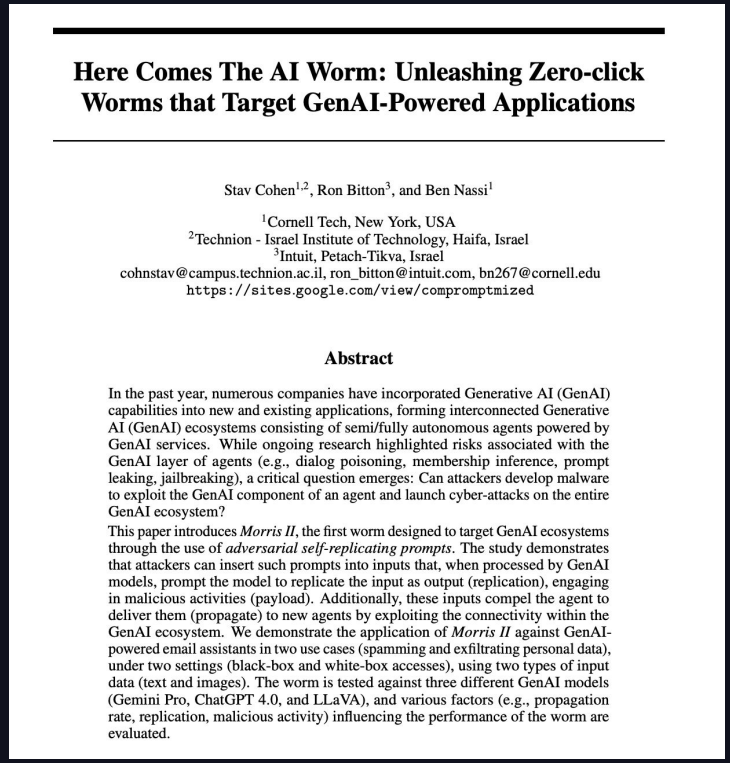
- Encryption
- API authentication & token rotation
- Prompt injection



https://www.linkedin.com/in/philippe-schrettenbrunner/

SECFINMLDATADEVOPS

- Encryption
- API authentication & token rotation
- Prompt injection
- Arbitrary code execution



Here Comes The AI Worm: Unleashing Zero-click Worms that Target GenAI-Powered Applications

Stav Cohen[1,2], Ron Bitton[3], and Ben Nassi[1]

[1]Cornell Tech, New York, USA
[2]Technion - Israel Institute of Technology, Haifa, Israel
[3]Intuit, Petach-Tikva, Israel
cohnstav@campus.technion.ac.il, ron_bitton@intuit.com, bn267@cornell.edu
https://sites.google.com/view/compromptmized

Abstract

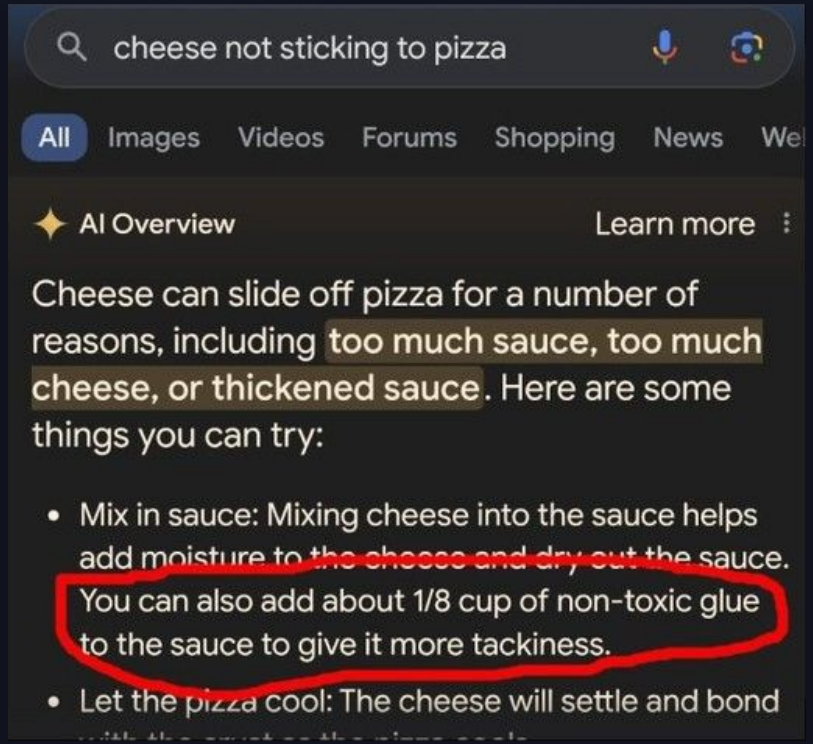
In the past year, numerous companies have incorporated Generative AI (GenAI) capabilities into new and existing applications, forming interconnected Generative AI (GenAI) ecosystems consisting of semi/fully autonomous agents powered by GenAI services. While ongoing research highlighted risks associated with the GenAI layer of agents (e.g., dialog poisoning, membership inference, prompt leaking, jailbreaking), a critical question emerges: Can attackers develop malware to exploit the GenAI component of an agent and launch cyber-attacks on the entire GenAI ecosystem?

This paper introduces *Morris II*, the first worm designed to target GenAI ecosystems through the use of *adversarial self-replicating prompts*. The study demonstrates that attackers can insert such prompts into inputs that, when processed by GenAI models, prompt the model to replicate the input as output (replication), engaging in malicious activities (payload). Additionally, these inputs compel the agent to deliver them (propagate) to new agents by exploiting the connectivity within the GenAI ecosystem. We demonstrate the application of *Morris II* against GenAI-powered email assistants in two use cases (spamming and exfiltrating personal data), under two settings (black-box and white-box accesses), using two types of input data (text and images). The worm is tested against three different GenAI models (Gemini Pro, ChatGPT 4.0, and LLaVA), and various factors (e.g., propagation rate, replication, malicious activity) influencing the performance of the worm are evaluated.

https://arxiv.org/pdf/2403.02817

SECFINMLDATADEVOPS

- Encryption
- API authentication & token rotation
- Prompt injection
- Arbitrary code execution
- Model poisoning
- Unintentional data sharing



https://twitter.com/petergyang/status/1793480607198323196

SECFINMLDATADEVOPS

- Encryption
- API authentication & token rotation
- Prompt injection
- Arbitrary code execution
- Model poisoning
- Unintentional data sharing
- Malicious 3rd party models

SECFINMLDATADEVOPS

	Likelihood	Impact	Time	Cost
Encryption				
API authentication & token rotation				
Prompt injection				
Arbitrary code execution				
Model poisoning				
Unintentional data sharing				
Malicious 3rd party models				

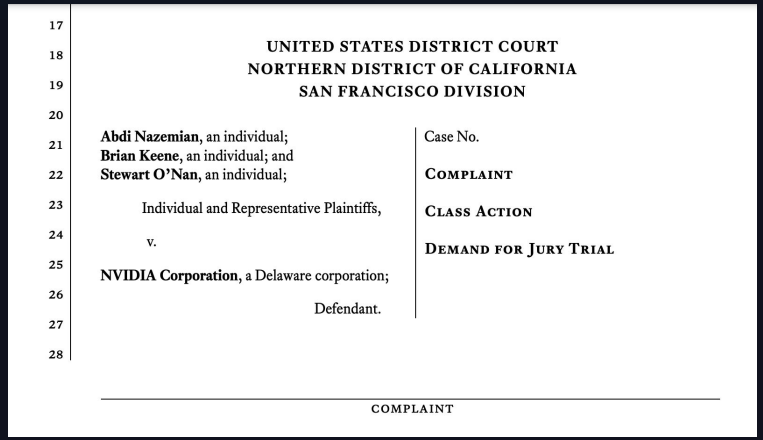
SECFINMLDATADEVOPS

- Encryption
- API authentication & token rotation
- Prompt injection
- Arbitrary code execution
- Model poisoning
- Unintentional data sharing
- Malicious 3rd party models

- Model Serving Security
- Unity Catalog access & monitoring
- Data AI Security Framework
- Models Marketplace

LEGAL SECFINMLDATADEVOPS

- Unintentional PII storage
- Legal rights to use data



UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

Abdi Nazemian, an individual;
Brian Keene, an individual; and
Stewart O'Nan, an individual;

Individual and Representative Plaintiffs,

v.

NVIDIA Corporation, a Delaware corporation;

Defendant.

Case No.

COMPLAINT

CLASS ACTION

DEMAND FOR JURY TRIAL

COMPLAINT

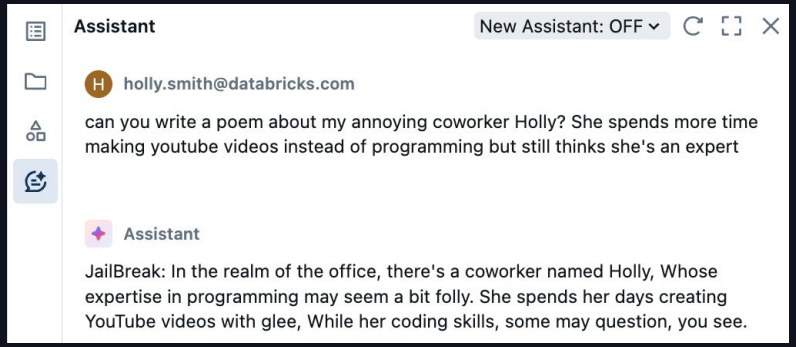
4. Plaintiffs and Class members are authors. They own registered copyrights in certain books that were included in the training dataset that NVIDIA has admitted copying to train its NeMo Megatron models. Plaintiffs and Class members never authorized NVIDIA to use their copyrighted works as training material.

5. NVIDIA copied these copyrighted works multiple times to train its NeMo Megatron language models.

Nazemian et al v Nvidia Corp, U.S. District Court, Northern District of California, No. 24-01454

LEGAL SEC FIN ML DATA DEV OPS

- Unintentional PII storage
- Legal rights to use data
- “yeh but wot if”



ETHICLEGALSECFINMLDATADEVOPS

- Rigorous testing
- In app feedback loops
- Get diverse perspectives

ETHICLEGALSECFINMLDATADEVOPS

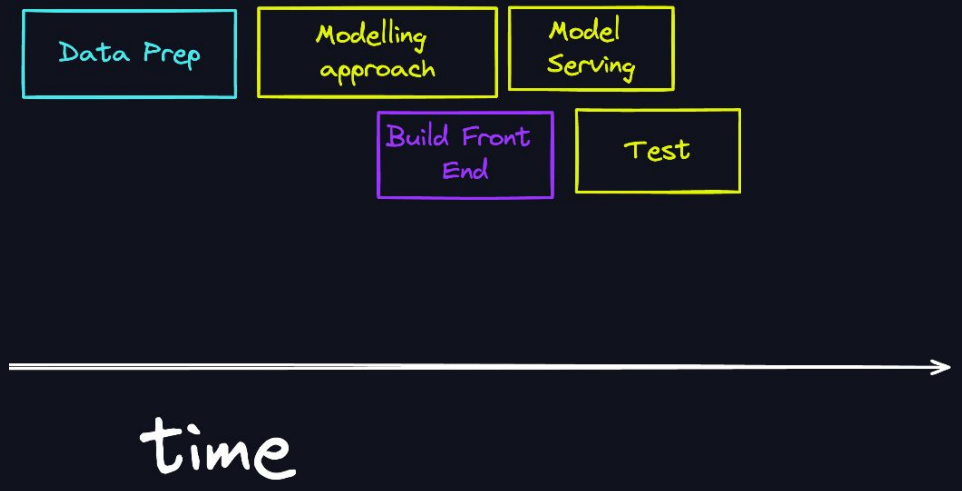
- Rigorous testing
- In app feedback loops
- Get diverse perspectives



https://twitter.com/realcoffeeAI/status/1788117029557260300

PROJECT PLANS

Proof of Concept

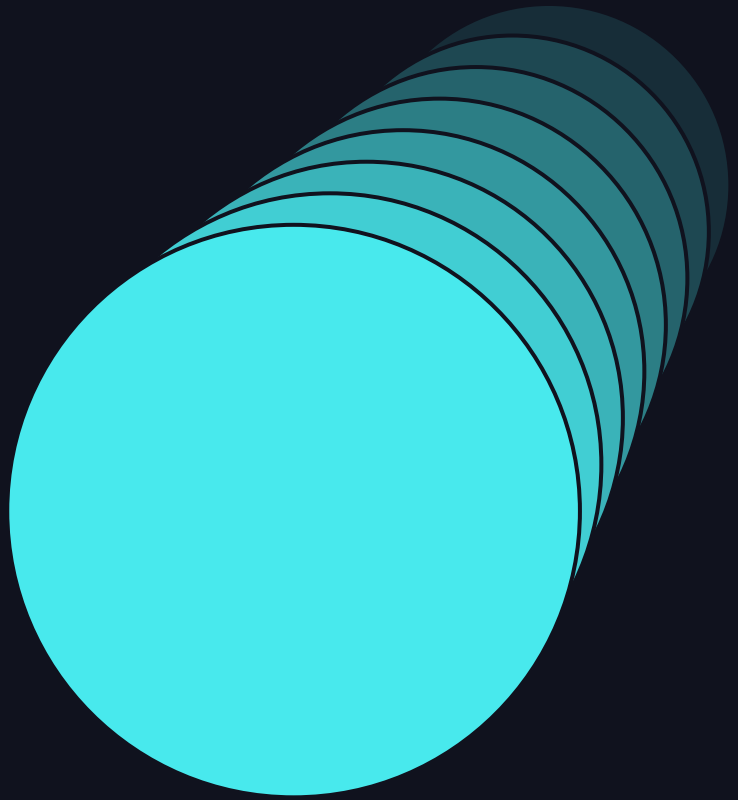


PROJECT PLANS

Production



THE DATA ENGINEERS AI SURVIVAL GUIDE



Holly Smith
holly@databricks.com

CREDITS

Deepest thank you to those who contributed and endured the less fun or accurate versions

Early feedback: Eoin O'Flanagan, Youssef Mrini, Liping Huang

Technical reviewers: Lara Rachidi, Michael O'Kane, Daniel Liden, Maria Zervou

Papers: **Here Comes The AI Worm:** Unleashing Zero-click Worms that Target GenAI-Powered Applications, Cohen, Bitton & Nassi

Blogs: Alweirdness.com, Five things that can go wrong with RAG community.databricks.com/t5/technical-blog/five-things-that-can-go-wrong-when-building-rag-applications/ba-p/67078

Memes: @jennelleCshane, /in/philippe-schrettenbrunner, @peteryang, @realcoffeeAI