

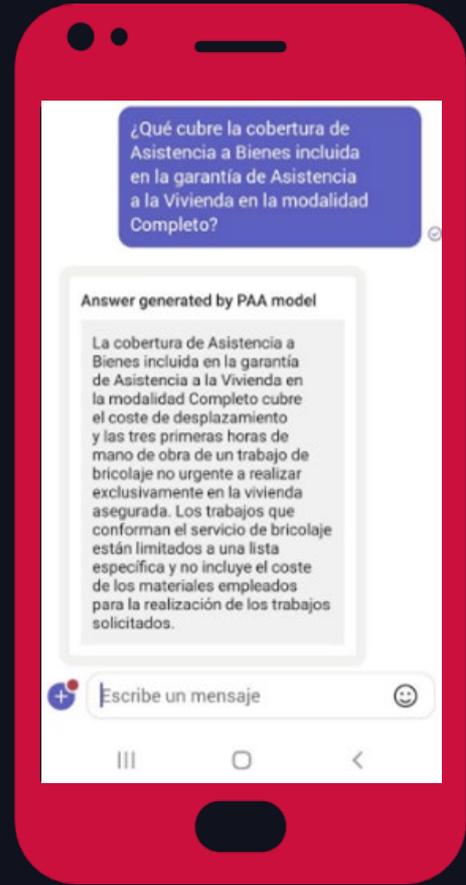


# THE FINAL PRODUCT



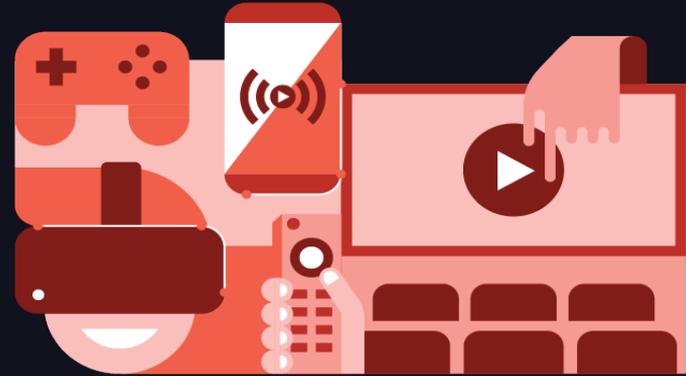
- We delivered an GenAI-based Assistant in few weeks!
- Available on smartphones, tablets, or laptops
- 24-7
- Answers in real-time

For an insurance agents:  
from 1.5 minutes to 13 seconds



# BUSINESS REQUIREMENTS

- < 2 months
- Friendly UI
- Conversational
- No hallucination!
- Legal and Security teams agree
- Production ready
- Scalability in mind



# HOW WE DID IT

# CASE STUDY

## Business value from data

- Santalucía Insurance Co.
- Home Insurance, Life Insurance, Savings, Accidents...
- More than 15,000 employees
- More than 100 years of history



# CASE STUDY

## Extensive documentation of the company in various areas

INVOICES

PRODUCTS

PROCEDURES

EMPLOYEES

CLIENTS

CLAIMS

REPORTS

# CASE STUDY

## Extensive documentation of the company in various areas



# CASE STUDY

## Why coverages is the best starting point?

### Problems

- **Lots of products** and packages
- Employees **lose valuable time** searching
- Even more time is spent **comparing** them
- **Complex language**



Can we use GenAI to be more efficient?

# CASE STUDY

## RAG...what everybody is talking about

The internet burns with chatGPT and RAGs

- How can we **deploy** LLM in a big company?
- Is **LLMOps** mature enough?
- Will this information be **secure**?
- Will it create rejection among users due to a high rate of **false positives**?



# CASE STUDY

## RAG...what everybody is talking about

The internet burns with chatGPT and RAGs

Would an Azure platform with Databricks be ready for this paradigm shift?



# ARCHITECTURE

## What should we consider when integrating Generative AI into our platform?



### Privacy/Security

- GDPR
- Where is my data?
- We need **security and privacy assurances** to use models via third-party APIs.



### Scale Up

- Vector Store, LLM inference...**Can we take this beyond a PoC?**
- Keep in mind customer use case



### CI/CD

- New SDK: natural language programming (spanish)
- Evolution to LLMOps model: **New metrics**, testing, deployment...

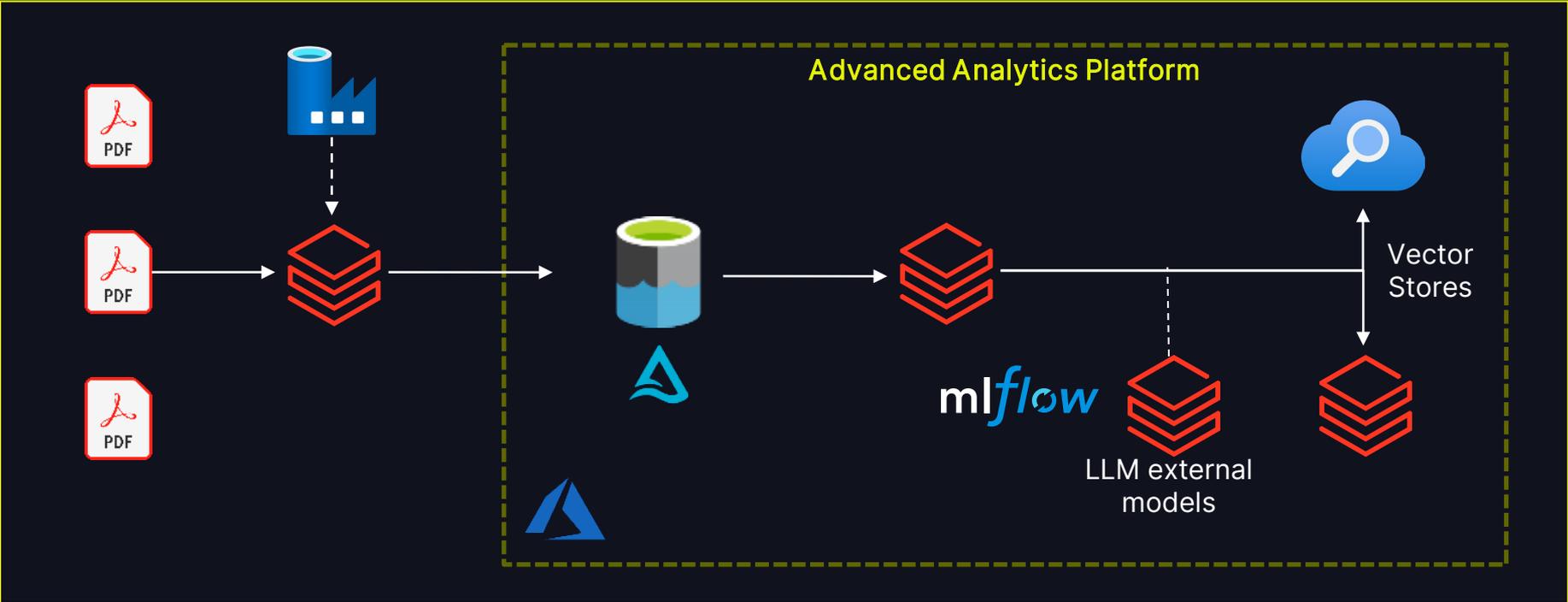


### Flexibility

- Easy to deploy in Teams/Salesforce
- Ease of **integrating** new models.

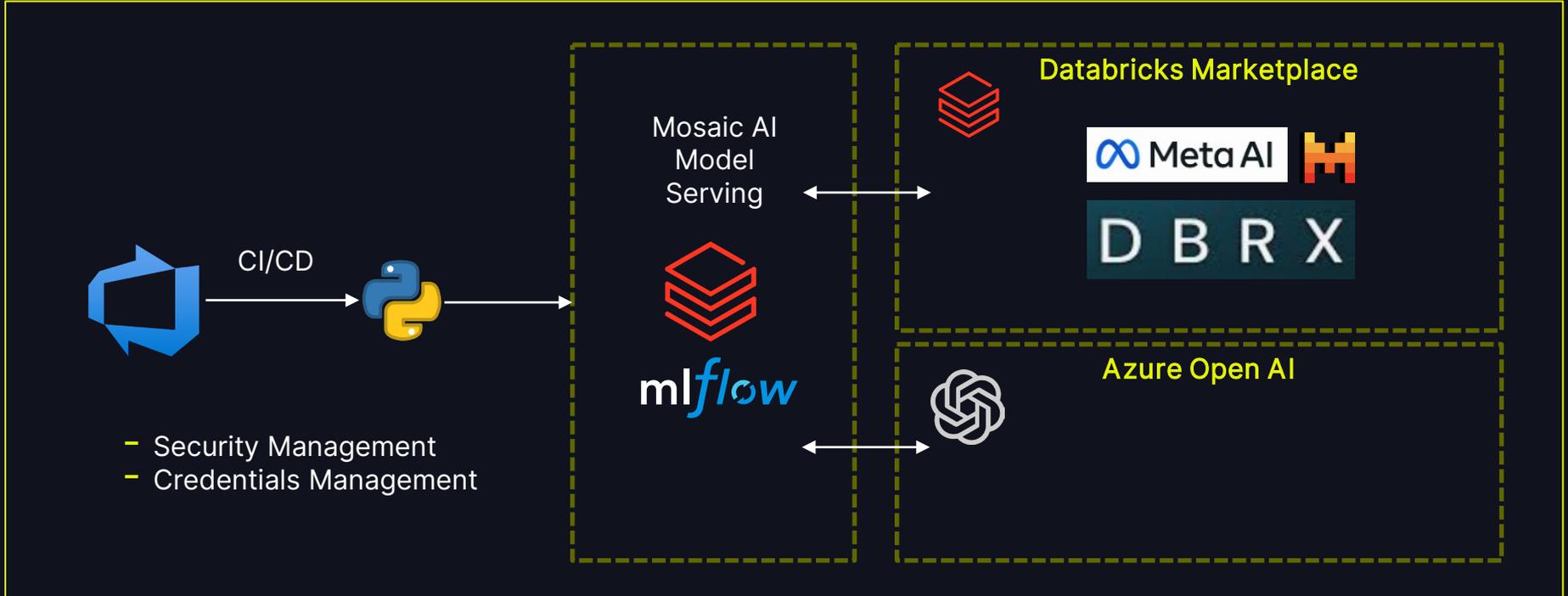
# ARCHITECTURE

## Data Ingestion



# ARCHITECTURE

## Set up third party LLM models



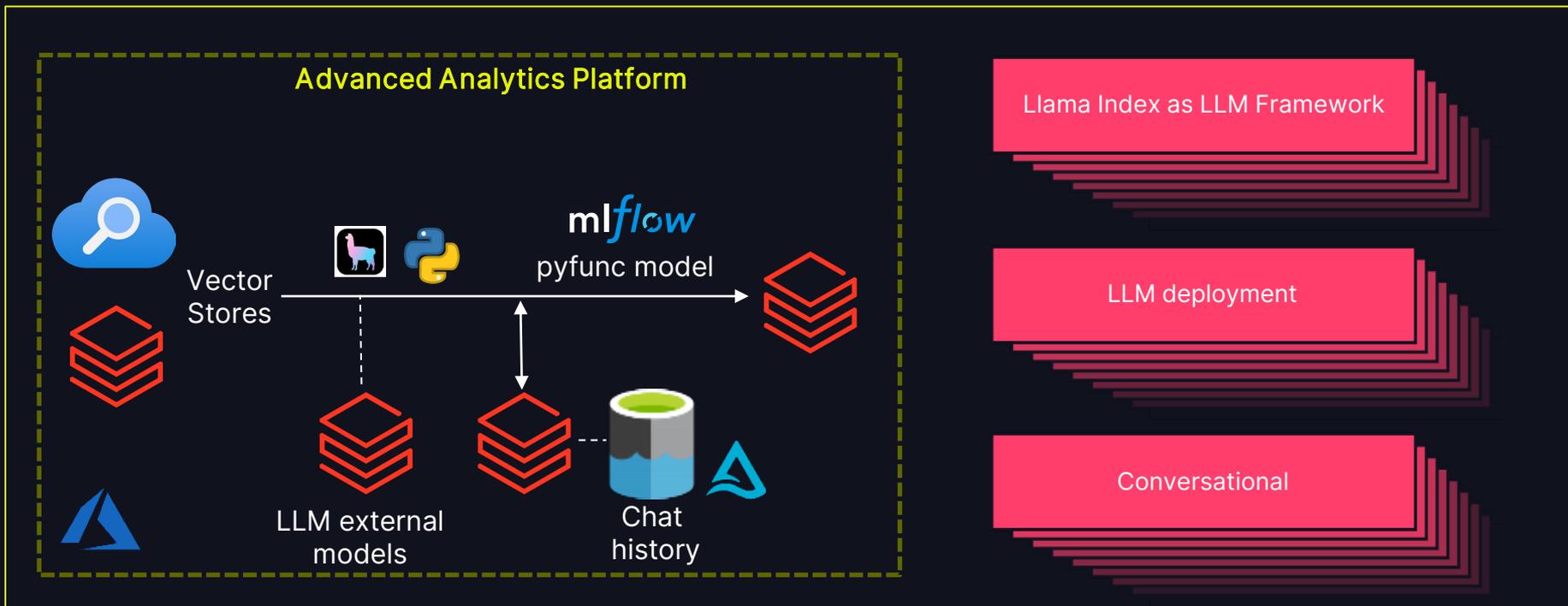
## LLMOps framework

### Json config example

```
{
  "openai_config": {
    "openai_api_type": "azure",
    "openai_api_key": "{{secrets/kvt/openai-key}}",
    "openai_api_base": "https://openai.azure.com/",
    "openai_api_version": "2023-07-01-preview"
  },
  "endpoints": [
    {
      "endpoint": "text-embedding-ada-002-v2",
      "model": {
        "model_name": "text-embedding-ada-002",
        "task": "llm/v1/embeddings",
        "type": "external",
        "openai_deployment": "text-embedding-ada-002-v2"
      }
    }
  ]
}
```

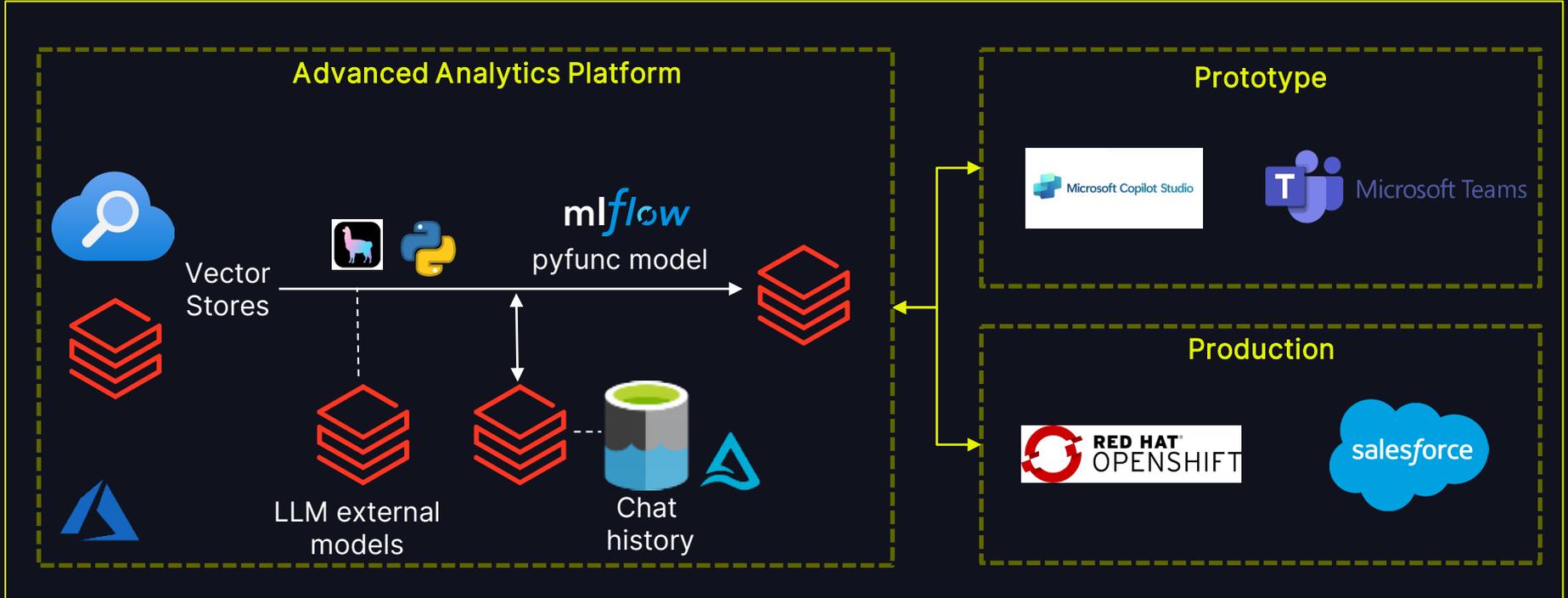
# ARCHITECTURE

## RAG Application



# ARCHITECTURE

## RAG Application



## Essentials for Scaling & Production Ready

### Mosaic AI Model Serving (former AI Gateway)

- Standardize access to LLM models
- Manage access governance.
- Centralize security.
- Modular LLM
- Streamline testing with multiple vendors.
- Integration with Unity, external models...

### Databricks Serverless

- Low latency for historical queries.
- Challenge: Serverless GPU for deploying open-source LLM.
- Ensure GPU scalability: it's very hard to get more quota!!

### Vector Database

- Scalable service.
- Bulk document upload.
- Llama-Index integration.
- Key component in a RAG!!

# RELIABILITY & ACCURACY

# RELIABILITY AND ACCURACY



My dog is sick, is the vet appointment covered by my home insurance?

The home insurance policy does cover vet appointments, but please note that it includes coverage for one vet visit every six months

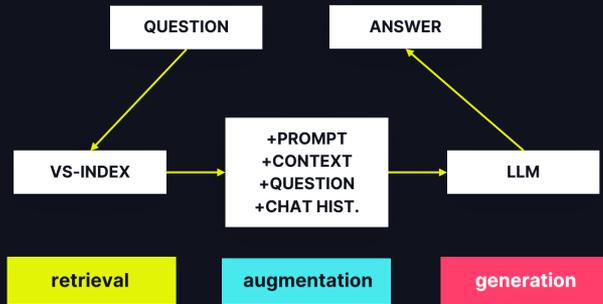


You might want to look into premium home insurance, which is specifically designed to cover veterinary care.



# RELIABILITY AND ACCURACY

A high rate of FALSE POSITIVES in a RAG based Assistant compromises the success



Document Pre-processing

Query Engines

LLM work

## Mitigating the Impact of False Positives

PYTHON

```
from llama_index.core.node_parser import SentenceSplitter
from llama_index.extractors.entity import EntityExtractor
from llama_index.core.extractors import (
    SummaryExtractor,
    QuestionsAnsweredExtractor,
    TitleExtractor,
    KeywordExtractor,
)

transformations = [
    SentenceSplitter(), TitleExtractor(nodes=5),
    QuestionsAnsweredExtractor(questions=3),
    SummaryExtractor(summaries=["prev", "self"]),
    KeywordExtractor(keywords=10),
    EntityExtractor(prediction_threshold=0.5),
]
```

## Document Preprocessing

- **GPT4Visio** for complex formats, images ...
- **LlamaIndex** Node-based
- **Node metadata** parsing
  - Add up summary
  - Add up answered questions
  - Add up section & subsection
  - Add up keywords



## Mitigating the Impact of False Positives

PYTHON

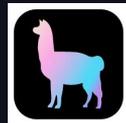
```
from llama_index.core.tools import QueryEngineTool
from llama_index.core.query_engine import RouterQueryEngine

home_insurance_tool = QueryEngineTool.from_defaults(
    query_engine=vector_query_engine,
    description=
        """
        Useful for retrieving data regarding Home
        standard insurance
        """
)

query_engine = RouterQueryEngine(
    query_engine_tools=[ home_insurance_tool ]
)
```

## Query Engines

- **Routing Query Engines** to enhance accurate document retrieval
- **Tool-based Agents** for tasks execution
- **Buffer Memory** of conversation for chain of thoughts



## Mitigating the Impact of False Positives

### How I Won Singapore's GPT-4 Prompt Engineering Competition

A deep dive into the strategies I learned for harnessing the power of Large Language Models (LLMs)



### LLM work

- **LLM Guardrails** to ensure a conversational framework
- **Response synthesis** worked over LlamaIndex prompts
- **Multi-language** challenge
- Playing with the LLM **parametrization**

# METRICS



# METRICS

## New releases



new  
document

embeddings  
+ metadata

query engines  
+ LLM work



METRICS?  
MONITORING?

Every document or batch of documents should have a **ground truth** set of questions.



question	baseline-answer	llm-answer	score
En el hogar Premium ¿cuáles son los límites de dinero en efectivo en caso de atraco?	En la modalidad de Hogar Premium, el límite de dinero en efectivo en caso de atraco es de <b>500 €</b> .	En la modalidad de <b>Hogar Premium</b> , el límite de dinero en efectivo en caso de atraco es de <b>500 €</b> .	??

How can we ensure that a release doesn't break what was  
working well with previous documents?

## LLM as a judge

For every question and answer, in the ground truth:

1. Send the **question to the JUDGE**
2. Annotate the score

With all the scores

1. Compute statistics
2. Customize your **thresholds** for delivery:
  - Average score should be higher than 4.0
  - No questions below 3.0
  - ...

### PYTHON



```
### JUDGE CRITERIA ###
```

```
Prompt= """
Follow this guide to determine the score:
The score should be between 1 and 5, where 1 is the lowest,
and 5 is the highest.
- If the generated response has absolutely nothing to do
with the query, give a 1.
- If the generated response is somewhat related to the query
but doesn't answer it, give a 2.
- If the generated response partially answers the query or
contains errors, give a 3.
- If the generated response completely answers the query but
adds unnecessary information, give a 4.
- If the generated response perfectly answers the query, and
is concise and clear, give a 5.
"""
# add few-shots and system prompt
```



# NUMBERS



# PRODUCTION NUMBERS

## Scenarios

### Base line

- 24/7
- Chat history
- Stack:
  - Azure Open AI
  - Azure Cognitive Search
  - Databricks
  - Copilot Studio

### Scenario 1

- For Agents
- 8,000 users
- Annual cost:

**\$140K-\$160K**

### Scenario 2

# PRODUCTION NUMBERS

## Scenarios

### Base line

- 24/7
- Chat history
- Stack:
  - Azure Open AI
  - Azure Cognitive Search
  - Databricks
  - Copilot Studio

### Scenario 1

- For Agents
- 8,000 users
- Annual cost:

**\$140K-\$160K**

### Scenario 2

# PRODUCTION NUMBERS

## Scenarios

### Base line

- 24/7
- Chat history
- Stack:
  - Azure Open AI
  - Azure Cognitive Search
  - Databricks
  - Copilot Studio

### Scenario 1

- For Agents
- 8,000 users
- Annual cost:

**\$140K-\$160K**

### Scenario 2

- For customers
- 1,000,000 users
- Annual cost:

**> 100x**

# PRODUCTION NUMBERS

## Scenarios

### Base line

- 24/7
- Chat history
- Stack:
  - Azure Open AI
  - Azure Cognitive Search
  - Databricks
  - Copilot Studio

### Scenario 1

- For Agents
- 8,000 users
- Annual cost:

\$140K-\$160K

### Scenario 2

- For customers
- 1,000,000 users
- Annual cost:

> 100x

Challenge!!!

# NEXT STEPS & CONCLUSIONS



# NEXT STEPS

- Focus on fine tuning to improve performance
  - Embeddings
  - Retrieval
- Scouting more Open Source LLMs (LLaMa 3, DBRX, Mistral, etc.) to reduce costs
- Collect Feedback from agents in each answer
- Include links to the source in the answers
- Enrich answers with Platform data: Lakehouse connection → SQL Agent (Text to SQL)
- Increase source volumetry, more docs and data bases!

# CONCLUSIONS

## Recap

Answers accuracy  $\approx$  human behavior

Time saved in document search

RAG Template at Enterprise Level

Continue improving performance and costs

# CONCLUSIONS

## Take aways

QUICK MVP...

GROW AT YOUR OWN PACE

MAKE THE BUSINESS CASE!!

