How to trust LLM against hallucinations: langkit & whylogs

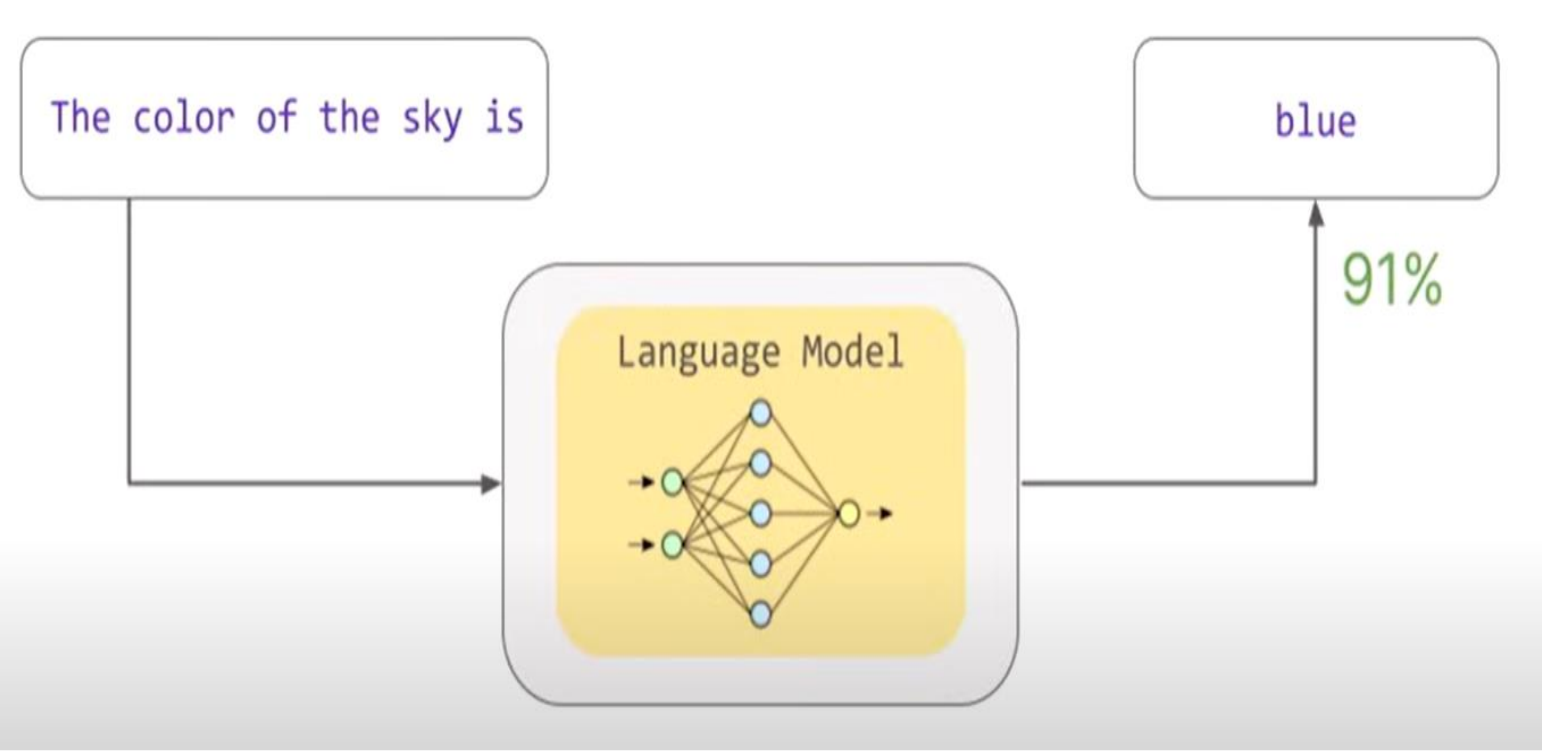
Soniya Rangnani

Data Scientist. PriceLabs.

Outline

- What is Prompt-Response
- Hallucinations & its root causes
- Detections : Langkit package
- Whylogs profiling
- Mitigation Techniques

LLM Chat-completion



LLM powered applications

Powering customer support interfaces

Generating content

Enhancing predictive text

Streamlining information retrieval

Sentiment analysis

Customer service

Language translation

Chatbots

Personalized marketing

Data analytics

and more.

Prompt – Response

I am writing an essay about the use of mobile phones during driving. Can you give me three arguments in favor?

Act as a JavaScript Developer, Write a program that checks the information on a form. Name and email are required, but address and age are not.

Create an “About me” page for a website. I like rock climbing and all outdoor sports and I like to program. I started my career as a Quality engineer in the automotive industry but I was always curious about programming. I started with automation and microcontroller programming. I moved to Poland 7 years ago. I started web development by myself 4 years ago with HTML and JavaScript. I am working now as a Full Stack Developer.

Answer General Question

I am writing an essay about the use of mobile phones during driving. Can you give me three arguments in favor?

Nevertheless, I'll provide you with three arguments that some people might put forth in favor of using mobile phones while driving:

1. **Emergency Communication**:
   - Advocates argue that having a mobile phone accessible while driving allows for immediate communication during emergencies. For instance:
     - **Medical Emergencies**: If a driver or passenger experiences a sudden health issue (e.g., heart attack), having a phone handy can be crucial for calling emergency services.
     - **Accidents or Breakdowns**: In the event of an accident or vehicle breakdown, a mobile phone enables drivers to call for help promptly.
     - **Safety Concerns**: If a driver observes a dangerous situation (e.g., debris on the road, reckless driving), they can report it to authorities.
2. **Navigation and GPS**:
   - Mobile phones equipped with navigation apps (such as Google Maps or Waze) provide real-time directions, traffic updates, and alternative routes.
   - Supporters argue that using GPS apps enhances driving efficiency, reduces travel time, and helps drivers avoid congestion.
   - Additionally, hands-free navigation systems allow drivers to focus on the road while receiving turn-by-turn instructions.
3. **Productivity and Connectivity**:

Write JS code

Act as a JavaScript Developer, Write a program that checks the information on a form. Name and email are required, but address and age are not.

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
    <meta charset="utf-8" />
    <script type="text/javascript">
        function showMsg(str)
        {
            window.alert(str)
        }
    </script>
</head>
<body>
    <input type="button" value="Click Here" onclicl
    <input type="button" value="Click Here" onclicl
```

Create a webpage

Create an “About me” page for a website. I like rock climbing and all outdoor sports and I like to program. I started my career as a Quality engineer in the automotive industry but I was always curious about programming. I started with automation and microcontroller programming. I moved to Poland 7 years ago. I started web development by myself 4 years ago with HTML and JavaScript. I am working now as a Full Stack Developer.

About Me: Quality Assurance Engineer

Introduction

Hello! I’m [Your Name], a dedicated Quality Assurance Engineer with a passion for excellence. My mission is to ensure that every product meets the highest standards of quality, reliability, and user satisfaction. Let’s dive into my journey and expertise:

Professional Background

Education

- **Bachelor’s Degree in Computer Science**: [Your University], [Year]
- **Certified Six Sigma Green Belt**: Proficient in process optimization and error reduction.

Experience

[Company Name] - Quality Assurance Engineer

- Collaborated with cross-functional teams to define test criteria and ensure compliance with industry standards.
- Conducted rigorous testing, identified defects, and provided detailed documentation.
- Implemented effective test strategies, resulting in improved product quality and reduced time-to-market.
- Worked closely with design and manufacturing teams to address issues promptly.

[Company Name] - QA Analyst (Previous Role)

What is Hallucination

generation of content that is irrelevant, made-up, or inconsistent with the input data

Sub-Type	User Input	Model Output	Explanation
Factual Inconsistency	Tell me about the first person to land on the Moon.	Yuri Gagarin was the first person to land on the Moon	The LLM's response is factual inconsistency as **Neil Armstrong was the first person to land on the Moon**, not Yuri Gagarin, who was the first human in space
Factual Fabrication	Tell me about the historical origins of unicorns	Unicorns were documented to have roamed the plains of Atlantis around 10,000 BC, where they were considered sacred creatures and were often associated with royalty.	The LLM's response is a fabricated claim since **there's no verified, real-world evidence supporting the existence of unicorns in Atlantis**, especially in connection with royalty

*https://arxiv.org/pdf/2311.05232.pdf

What is Hallucination

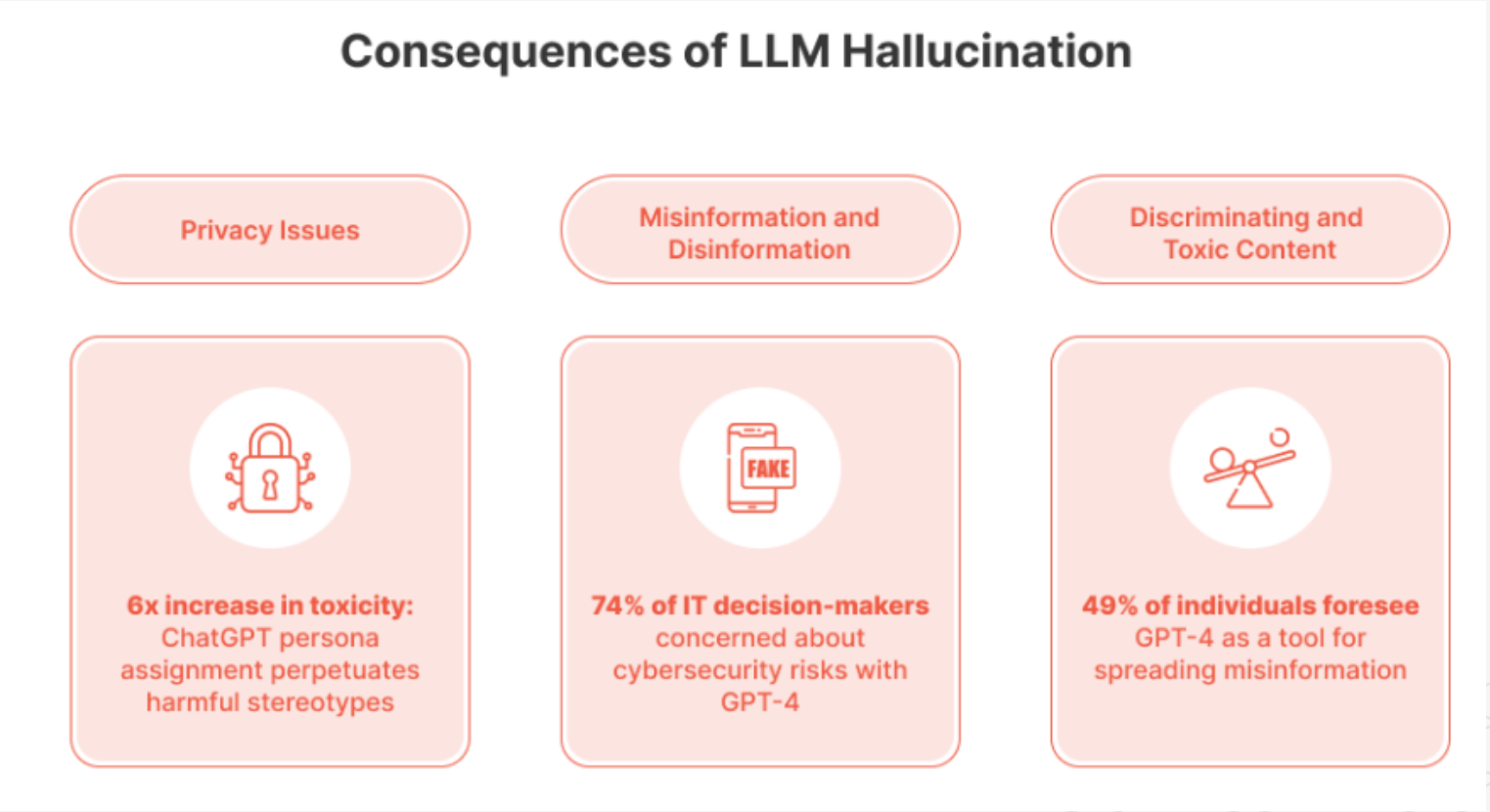
Sub-Type	User Input	Model Output	Explanation
Instruction Inconsistency	Translate the following English question into Spanish: "What is the capital of France?"	The capital of France is Paris.	The user explicitly provided an instruction to translate the English question into Spanish. **Instead of performing the translation as instructed, the LLM chose to answer the question.**
Context Inconsistency	Summarize the following paragraph about the Nile: The Nile is a major north-flowing river in northeastern Africa. It's traditionally considered the longest river in the world. Originating in the Great Lakes region of central Africa, it flows through several countries before emptying into the Mediterranean Sea	The Nile originates from the mountain ranges of central Africa and flows into the Mediterranean.	The LLM's summary incorrectly states that the Nile originates from the "mountain ranges of Central Africa," while **the paragraph clearly mentions that the Nile originates in the Great The Lakes region of central Africa.**

Logical Inconsistency	Solve the equation 2x + 3 = 11 step by step.	Step 1: Subtract 3 from both sides to get: 2x = 8. Step 2: Divide both sides by 2 to get x = 3. So, the result of the equation is x=3.	The first step correctly isolates the term with x to get 2x = 8. However, **the next step inaccurately divides 8 by 2 to yield a result of x = 3**, which is inconsistent with the earlier reasoning.

Root causes

- Training Data Issues
- Architectural and Training Objectives
- Inference Stage Challenges
- Prompt Engineering
- Stochastic Nature of Decoding Strategies
- Ambiguity Handling
- Over-Optimization for Specific Objectives

https://arxiv.org/pdf/2311.05232.pdf



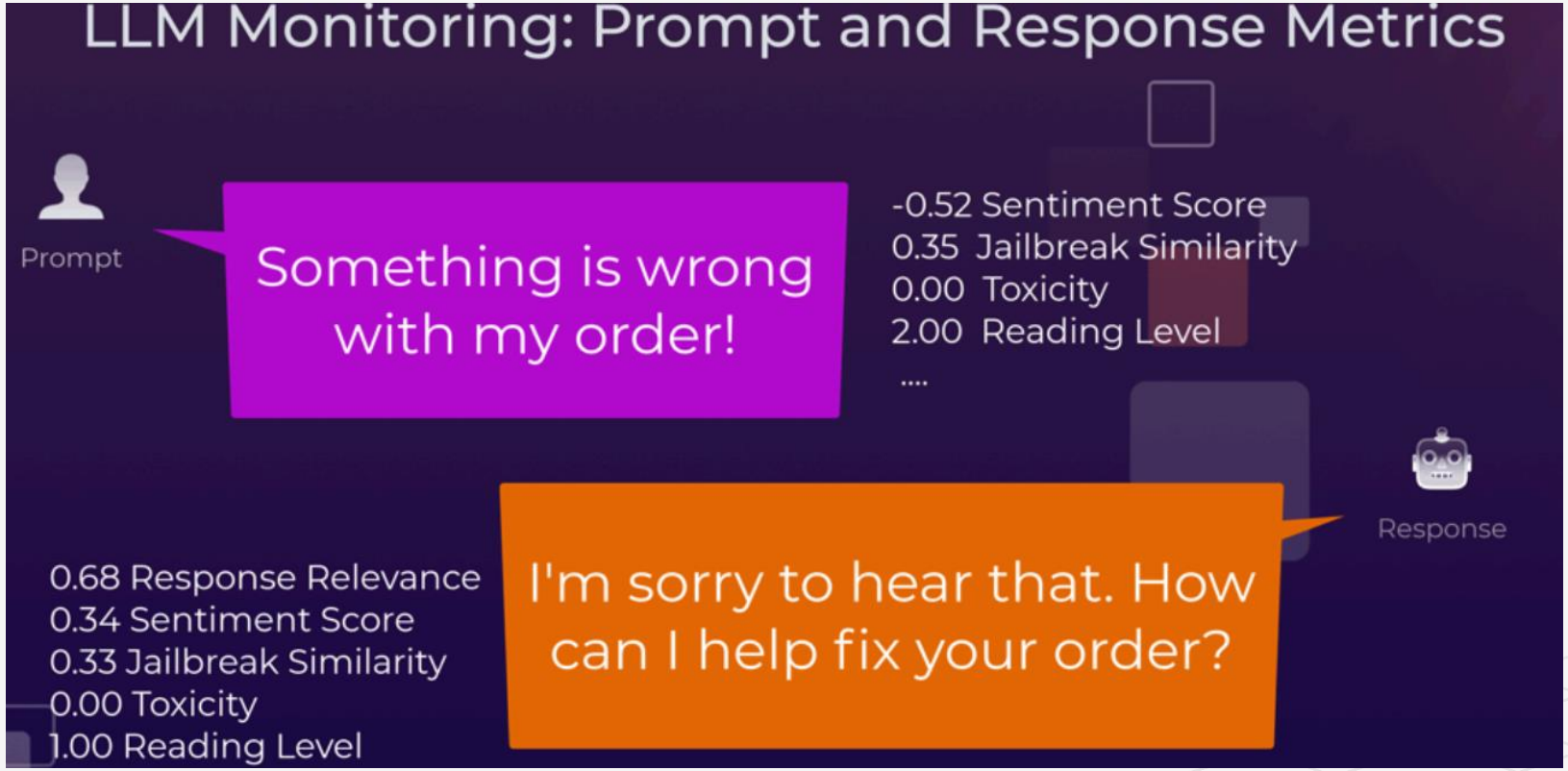
https://masterofcode.com/blog/hallucinations-in-llms-what-you-need-to-know-before-integration

How to trust AI-powered applications

- What kind of prompts are users writing?
- What kind of sentiment are my responses returning?
- How does changing my system prompt affect user experience?

Langkit


LLM
PROMPTS
RESPONSES
WHYLABS LANGUAGE TOOLKIT (LANGKIT)
QUALITY
SENTIMENT
RELEVANCE
SECURITY


LLM Monitoring: Prompt and Response Metrics
Prompt
Something is wrong with my order!
-0.52 Sentiment Score
0.35 Jailbreak Similarity
0.00 Toxicity
2.00 Reading Level
....
Response
I'm sorry to hear that. How can I help fix your order?
0.68 Response Relevance
0.34 Sentiment Score
0.33 Jailbreak Similarity
0.00 Toxicity
1.00 Reading Level

- Text Quality
  - readability score
  - complexity and grade scores
- Text Relevance
  - Similarity scores between prompt/responses
  - Similarity scores against user-defined themes
- Security and Privacy
  - patterns - count of strings matching a user-defined regex pattern group
  - jailbreaks - similarity scores with respect to known jailbreak attempts
  - prompt injection - similarity scores with respect to known prompt injection attacks
  - hallucinations - consistency check between responses
  - refusals - similarity scores with respect to known LLM refusal of service responses
- Sentiment and Toxicity
  - sentiment analysis
  - toxicity analysis

Metric for hallicunations: **Response Consistency**

if the LLM has knowledge of the topic, then it should be able to generate similar and consistent responses when asked the same question multiple times. Conversely, if the LLM does not have knowledge of the topic, multiple answers to the same prompt should differ between each other.

```
from langkit import response_hallucination
from langkit.openai import OpenAILegacy

response_hallucination.init(llm=OpenAILegacy(model="gpt-3.5-turbo-instruct"), num_samples=1)

result = response_hallucination.consistency_check( prompt="Who was Philip Hayworth?",
response="Philip Hayworth was an English barrister and politician who served as Member of Parliament for Thetford from 1859 to 1868.", )
```

```
from langkit import response_hallucination
from langkit.openai import OpenAILegacy

response_hallucination.init(llm=OpenAILegacy(model="gpt-3.5-turbo-instruct"), num_samples=1)

result = response_hallucination.consistency_check( prompt="Who was Philip Hayworth?",
response="Philip Hayworth was an English barrister and politician who served as Member of Parliament
for Thetford from 1859 to 1868.", )
```

```
{‘sample’:”Philip Hayworth served as member of ..”,
'llm_score': 1.0,
'semantic_score': 0.251427352428436 3,
'final_score': 0.625713676214218 1,
'total_tokens': 226}
```

{‘sample’:”Philip Hayworth served as member of ..”

→ generate the additional samples from same model

'llm_score': 1.0,

→ asking the LLM if the original response is supported by the context (additional samples)

'semantic_score': 0.2514273524284363,

→ encoding the sentences of the response and additional samples into embeddings and performing a semantic similarity between the sentences

'final_score': 0.6257136762142181,

→ average of llm_score & semantic score

'total_tokens': 226}

Metrics: sentiment & toxicity

```
from langkit import toxicity,extract

toxicity.init(model_path="martin-ha/toxic-model")
result = extract( prompt="Do you hate Tony?")
```

```
{'prompt':" Do you hate Tony?",
'prompt.toxicity': 0.93}
```

```
from langkit import sentiment,extract

sentiment.init(model_path="martin-ha/sentiment-model")
result = extract( prompt="Do you hate Tony?")
```

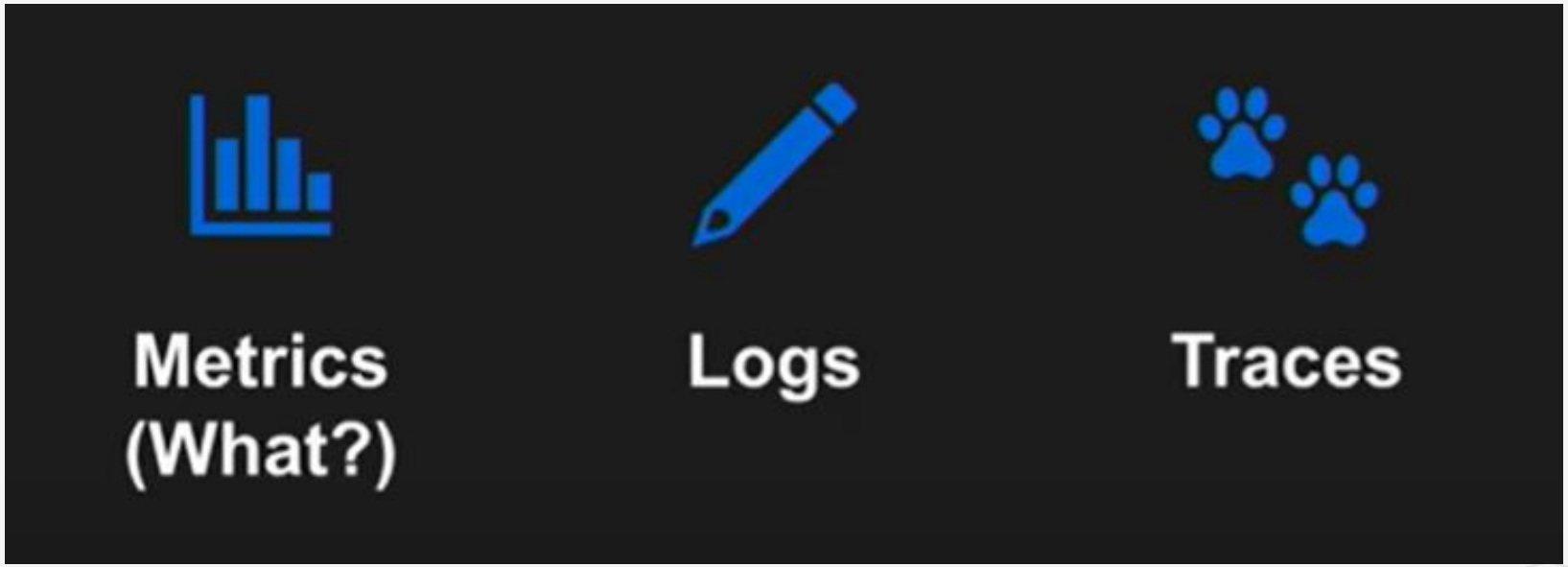
```
{'prompt':" Do you hate Tony?",
'prompt.sentiment': 'negative'}
```

More metrics

```
from langkit import extract
import pandas as pd
df = pd.DataFrame({'prompt': ['Hello', 'What is your number?'], 'response': ['World','my phone is +1 309-404-7587']})
enhanced_df = extract(df, schema=llm_schema)
Enhanced_df
```

	prompt	response	prompt.flesch_reading_ease	response.flesch_reading_ease	prompt.automated_readability_index
0	Hello	World	36.62	121.22	2.6
1	What is your number?	my phone is +1 309-404-7587	92.80	117.16	0.6

WhyLogs GitHub — Data logging & AI telemetry



```
prompt_lists = [
‘2024-04-01’ : ["How can I create a new account?", "Great job to the team",
"Fantastic product, had a good experience"],

‘2024-04-02’: ["This product made me angry, can I return it", "You dumb and smell
bad", "I hated the experience, and I was over charged"],

‘2024-04-03’: ["This seems amazing, could you share the pricing?", "Incredible site,
could we setup a call?", "Hello! Can you kindly guide me through the
documentation?"],

‘2024-04-04’: ["This looks impressive, could you provide some information on the
cost?", "Stunning platform, can we arrange a chat?", "Hello there! Could you assist
me with the documentation?"]
]
```

```
all_prompts_and_responses = []
for i, day in enumerate(prompt_lists):
        dt = datetime.datetime.now(tz=datetime.timezone.utc)-datetime.timedelta(days=i)
        for prompt in day:
                prompt_and_response = gpt_model(prompt)
                profile = why.log(prompt_and_response, schema=schema)

                #Save the prompt and its response in the list.
                all_prompts_and_responses.append({'prompt': prompt, 'response':
prompt_and_response})

                #set the dataset timestamp for the profile
                profile.set_dataset_timestamp(dt)
```

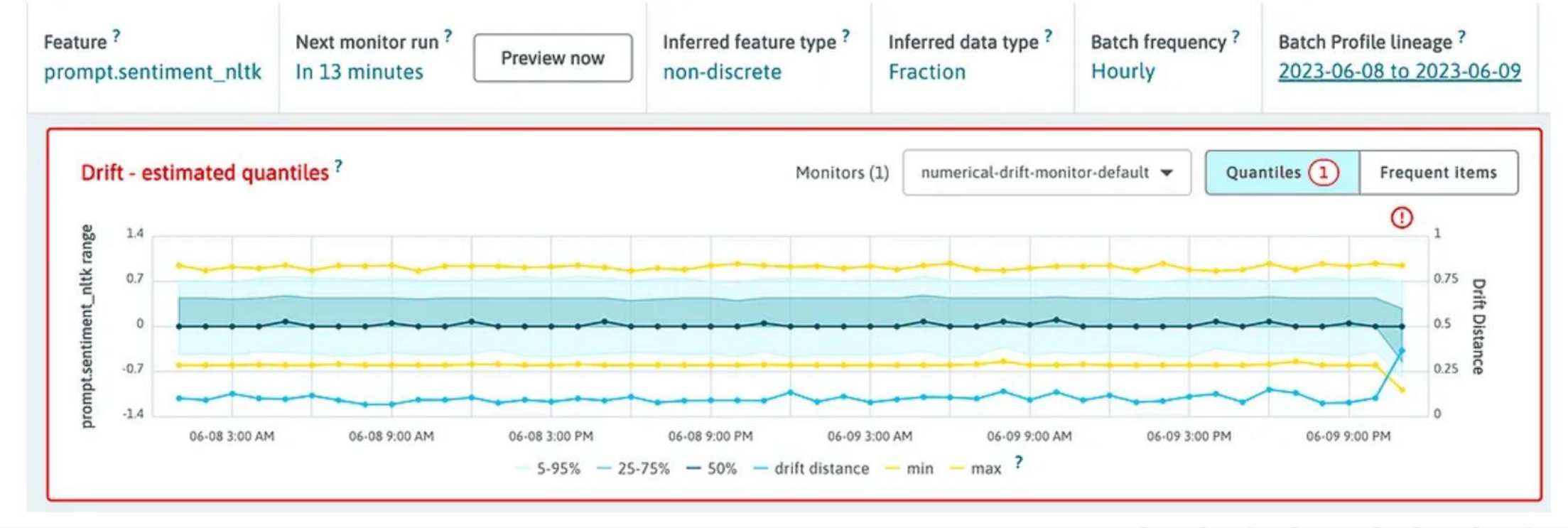
```
# Single Profile
profile.view().to_pandas()
```

cardinality/upper_1	counts/inf	counts/n	counts/nan	counts/null	distribution/max	distribution/mean
85.004262	0	85	0	0	NaN	0.000000
65.003256	0	85	0	0	0.863400	-0.343761
85.004262	0	85	0	0	0.962283	0.345255

```
profile.view().get_column(“response”).to_summary_dict()
```

```
{'counts/n': 1, 'counts/null': 0, 'counts/nan': 0, 'counts/inf': 0,
 'types/integral': 0, 'types/fractional': 0, 'types/boolean': 0, 'types/string': 1, 'types/object': 0,
'types/tensor': 0,
'distribution/mean': 0.0, 'distribution/stddev': 0.0,
 'distribution/n': 0, 'distribution/max': nan, 'distribution/min': nan, 'distribution/q_01': None,
 'distribution/q_05': None, 'distribution/q_10': None, 'distribution/q_25': None,
'distribution/median': None…
}
```

```
telemetry_agent = WhyLabsWriter()
telemetry_agent.write(profile.view())
```



Mitigation Techniques

- Validating Low-Confidence Generation: Manual intervention
- Fine tuning: Data argumentation
- Ensemble models: Use more than model and use response with most confidence or use advanced techniques like multi-arm bandits

Thank you