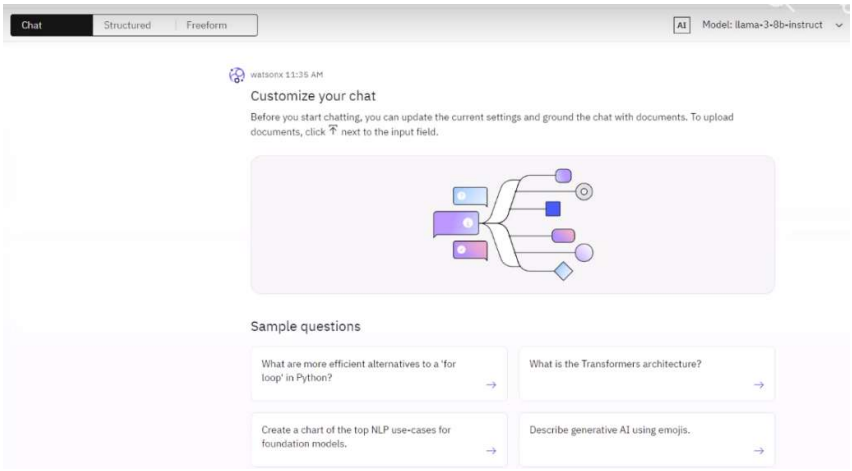
Challenge May 2025
Risky Stocks

A solution with IBM watsonx ai by Alex Fleischer
afleischer@fr.ibm.com

watsonx ai offers all kind of AI tools but for this challenge generative AI makes sense. So let's try the prompt lab within watsonx ai and choose the Structured option



You need to create a decision service that decides whether to buy certain stocks or not. Here are examples of the guiding rules:

- Rule 1: Stock in debt is considered risky.
- Rule 2: Stocks in fusion with other stocks may be risky.
- Rule 3: Stock in fusion with a strong stock is not risky.
- Rule 4: Do not buy risky stocks unless they have a good price.

Keep in mind that more rules that could be in conflict with some of these rules can be added later. Here are examples of a few stocks with the expected recommendations:

Stock Is In Debt	Stock Is In Fusion With Any Stock	Stock Is In Fusion With Strong Stock	Stock Has Good Price	Stock Is Risky	Buy Stock Shares
No	Yes	Yes	Yes	No	Yes
Yes	Yes	Yes	No	Yes	No
No	Yes	No	No	Yes	No
No	Yes	No	Yes	Yes	Yes
Yes	No	No	No	Yes	No

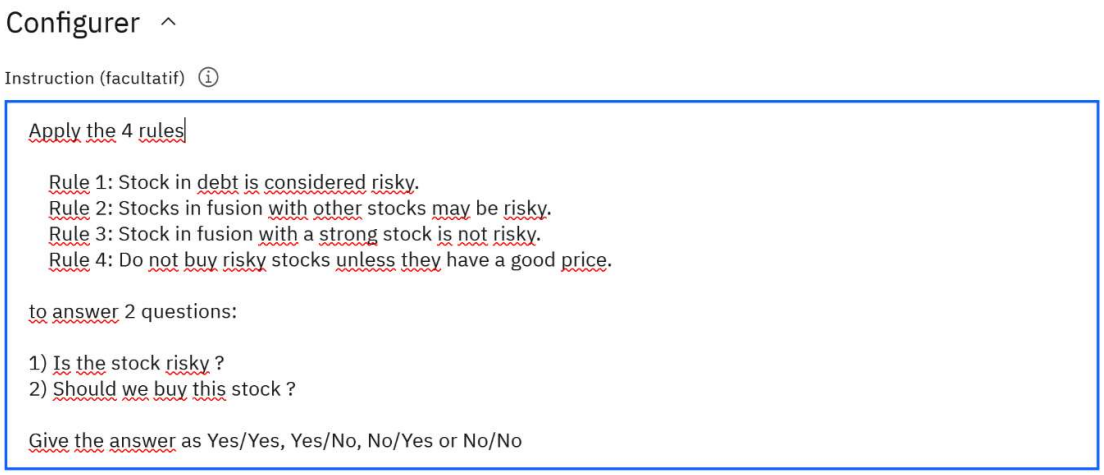
Let’s use the following prompt.

Apply the 4 rules

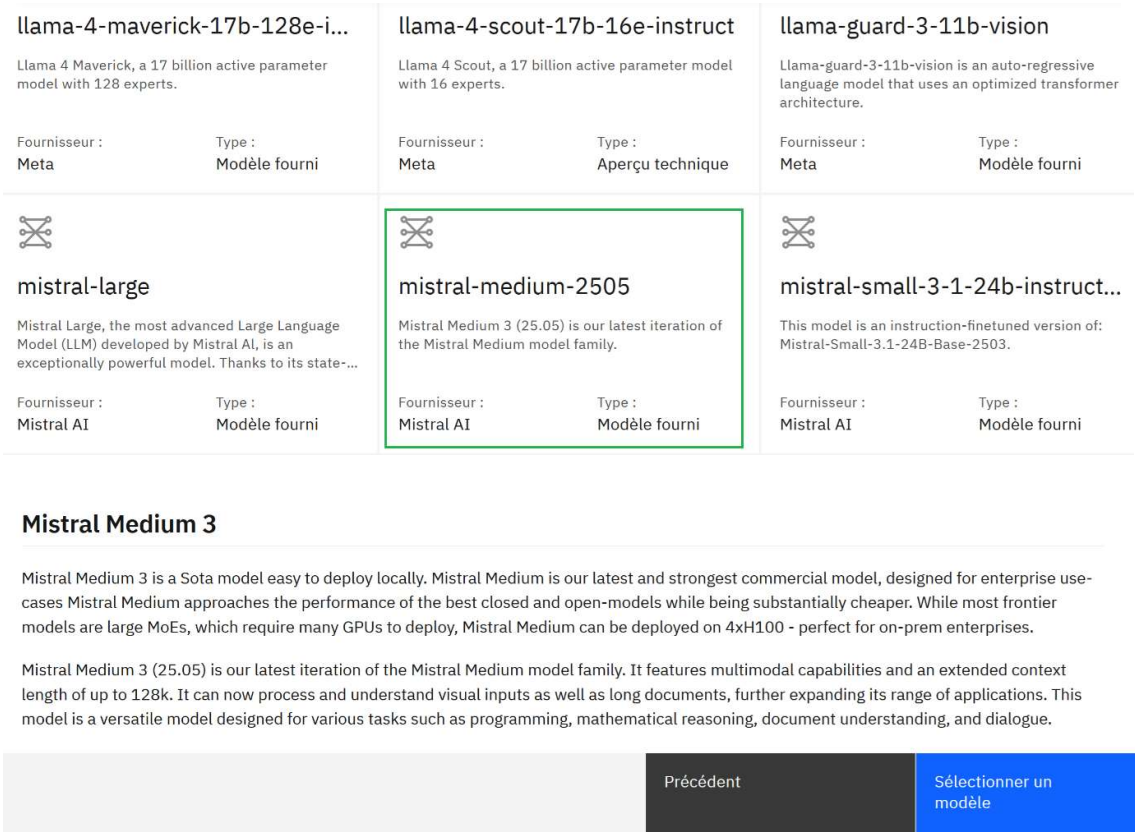
Rule 1: Stock in debt is considered risky.
Rule 2: Stocks in fusion with other stocks may be risky.
Rule 3: Stock in fusion with a strong stock is not risky.
Rule 4: Do not buy risky stocks unless they have a good price. to answer 2 questions:

1) Is the stock risky ?
2) Should we buy this stock ?

Give the answer as Yes/Yes, Yes/No, No/Yes or No/No



And then let’s choose a medium size model after checking the properties of that model : mistral medium



Now we can enter the 5 tests and run.

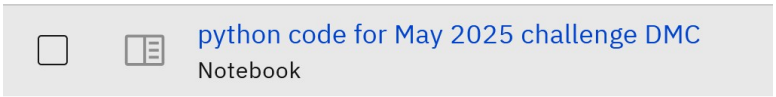
Entrée :	Sortie : AI
Stock in debt : No Stock in fusion with other stock : Yes Stock in fusion with a strong stock : Yes...	No/Yes
Stock in debt : Yes Stock in fusion with other stock : Yes Stock in fusion with a strong stock : Yes...	Yes/No
Stock in debt : No Stock in fusion with other stock : Yes Stock in fusion with a strong stock : No...	Yes/No
Stock in debt : No Stock in fusion with other stock : Yes Stock in fusion with a strong stock : No...	Yes/Yes
Stock in debt : Yes Stock in fusion with other stock : No Stock in fusion with a strong stock : No...	Yes/No

That is ok with

Stock Is In Debt	Stock Is In Fusion With Any Stock	Stock Is In Fusion With Strong Stock	Stock Has Good Price	Stock Is Risky	Buy Stock Shares
No	Yes	Yes	Yes	No	Yes
Yes	Yes	Yes	No	Yes	No
No	Yes	No	No	Yes	No
No	Yes	No	Yes	Yes	Yes
Yes	No	No	No	Yes	No

As can be checked in deeper details with

Within the prompt lab, we can add any new question by adding a cell. We can also save this in a notebook that will have everything to share this with any python user.



Without writing a single line of code we get:

Defining the model id

We need to specify model id that will be used for inferencing:

In [ ]:
```
model_id = "mistralai/mistral-medium-2505"
```

Defining the model parameters

We need to provide a set of model parameters that will influence the result:

In [ ]:
```
parameters = {
    "decoding_method": "greedy",
    "max_new_tokens": 200,
    "min_new_tokens": 0,
    "stop_sequences": ["\n"],
    "repetition_penalty": 1
}
```

Defining the project id or space id

The API requires project id or space id that provides the context for the call. We will obtain the id from the project or space in which this notebook runs:

In [ ]:
```
project_id = os.getenv("PROJECT_ID")
space_id = os.getenv("SPACE_ID")
```

Defining the inferencing input

Foundation model inferencing API accepts a natural language input that it will use to provide the natural language response. The API is sensitive to formatting. Input structure, presence of training steps (one-shot, two-shot learning etc.), as well as phrasing all influence the final response and belongs to the emerging discipline of Prompt Engineering.

Let us provide the input we got from the Prompt Lab:

In [ ]:
```
prompt_input = """Apply the 4 rules

    Rule 1: Stock in debt is considered risky.
    Rule 2: Stocks in fusion with other stocks may be risky.
    Rule 3: Stock in fusion with a strong stock is not risky.
    Rule 4: Do not buy risky stocks unless they have a good price.

to answer 2 questions:

1) Is the stock risky ?
2) Should we buy this stock ?

Give the answer as Yes/Yes, Yes/No, No/Yes or No/No

Entrée : Stock in debt : No
Stock in fusion with other stock : Yes
Stock in fusion with a strong stock  : Yes
Stock has a good price : Yes
Sortie :"""
```

Execution

Let us now use the defined Model object and pair it with input and generate the response:

In [ ]:
```
print("Submitting generation request...")
generated_response = model.generate_text(prompt=prompt_input, guardrails=True)
print(generated_response)
```

This can be done even faster:

In the prompt lab in 1 click you can copy/paste this in your code

Curl Node.js **Python**

```
import requests

url = "https://us-
south.ml.cloud.ibm.com/ml/v1/
text/generation?
version=2023-05-29"

body = {
  "input": """Apply the 4 rules

    Rule 1: Stock in debt is
considered risky.
    Rule 2: Stocks in fusion
with other stocks may be risky.
    Rule 3: Stock in fusion
with a strong stock is not
risky.
    Rule 4: Do not buy risky
stocks unless they have a good
price.

to answer 2 questions:

1) Is the stock risky ?
2) Should we buy this stock ?

Give the answer as Yes/Yes,
Yes/No, No/Yes or No/No
```

You can easily change the rules and add new ones.
Rule 3 can be reformulated as “Stock *not in debt* and in fusion with a strong stock is not risky.”

We just edit the prompt in the prompt lab and we get the same result as what we had with previous Rule 3

Apply the 4 rules

Rule 1: Stock in debt is considered risky.
Rule 2: Stocks in fusion with other stocks may be risky.
Rule 3: Stock not in debt and in fusion with a strong stock is not risky
Rule 4: Do not buy risky stocks unless they have a good price.

to answer 2 questions:

1) Is the stock risky ?
2) Should we buy this stock ?

Give the answer as Yes/Yes, Yes/No, No/Yes or No/No

And we get

Stock in debt : No Stock in fusion with other stock : Yes Stock in fusion with a strong stock : Yes...	No/Yes
Stock in debt : Yes Stock in fusion with other stock : Yes Stock in fusion with a strong stock : Yes...	Yes/No
Stock in debt : No Stock in fusion with other stock : Yes Stock in fusion with a strong stock : No...	Yes/No
Stock in debt : No Stock in fusion with other stock : Yes Stock in fusion with a strong stock : No...	Yes/Yes
Stock in debt : Yes Stock in fusion with other stock : No Stock in fusion with a strong stock : No...	Yes/No

This watsonx ai platform is hybrid in many ways:

1) Many different AI techniques from Machine Learning to generative AI with many different Large Language Models from Meta, Mistral, Google , IBM and many other companies.
2) Many different options from no code, low code and code
3) Run in many clouds like IBM, Azure and AWS but also on prem