From Paper to JSON

Automating Invoice Processing with Vertex AI

Andrei Terecoasa

Software Development Engineer at IBM

- Bucharest, Romania 🇷🇴
- tips & tricks on tere.ro
- some of the repos I maintain: andreitere
- Say hi at X andreiterecoasa

The AI Revolution ✨

Transforming industries with automation, customer support, and content generation.

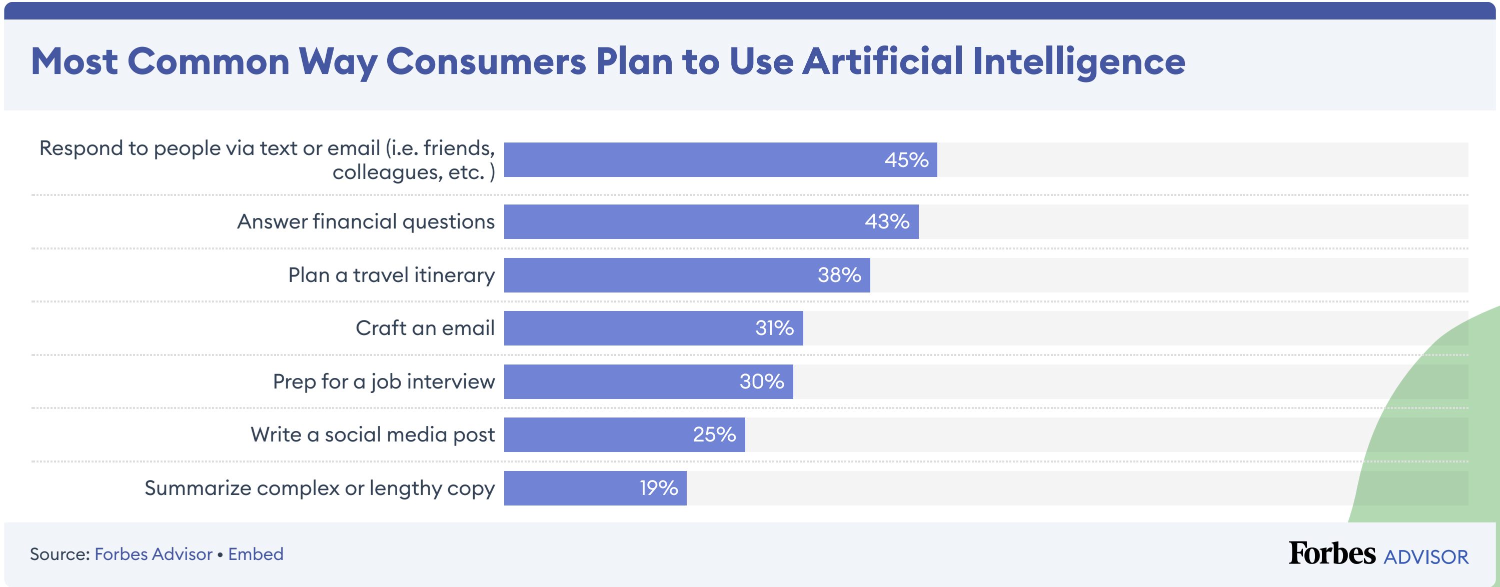
nefits:

- Understands natural language.
- Automates repetitive tasks.
- Scalable and adaptable to various industries

Opportunities:

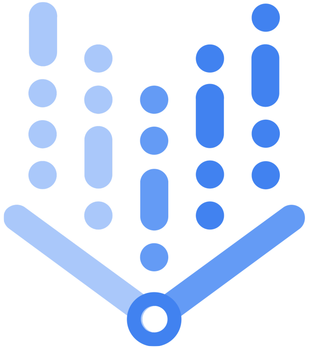
- Faster, more efficient workflows.
- Enhanced, personalized user experiences.
- Potential for new product innovations.

How is AI used?



Most Common Way Consumers Plan to Use Artificial Intelligence

Source: Forbes Advisor • Embed



Gemini

Vertex AI Model Garden

[redacted]

Vertex AI Studio

[redacted]

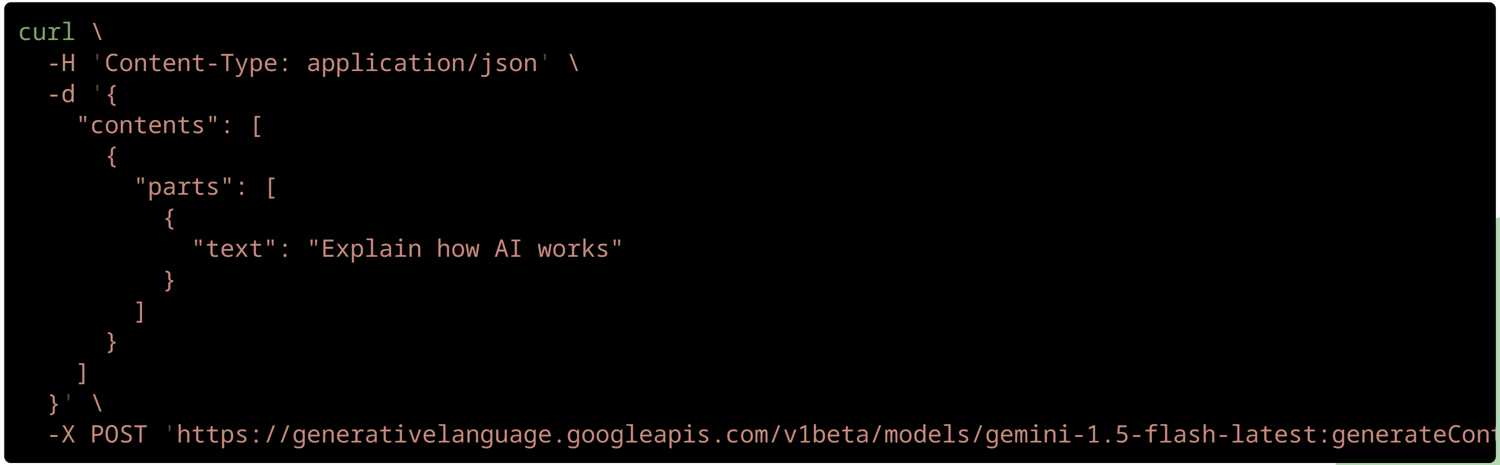
s

Google AI Studio

[redacted]

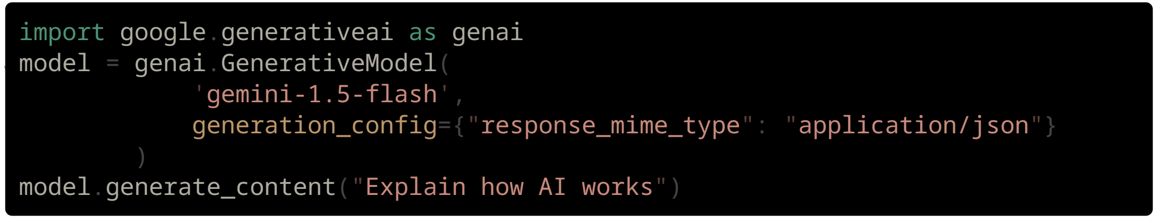
Calling Gemini Models through VertexAI?

VIA **REST** CALLS



```
curl \
  -H 'Content-Type: application/json' \
  -d '{
    "contents": [
      {
        "parts": [
          {
            "text": "Explain how AI works"
          }
        ]
      }
    ]
  }' \
  -X POST 'https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-latest:generateCont
```

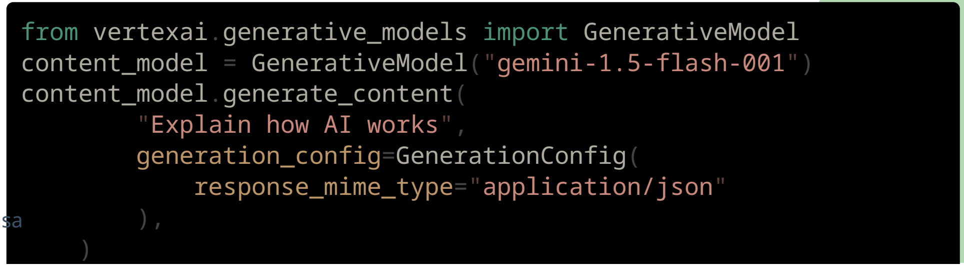
Calling Gemini Models through VertexAI?



```python
import google.generativeai as genai
model = genai.GenerativeModel(
            'gemini-1.5-flash',
            generation_config={"response_mime_type": "application/json"}
        )
model.generate_content("Explain how AI works")
```

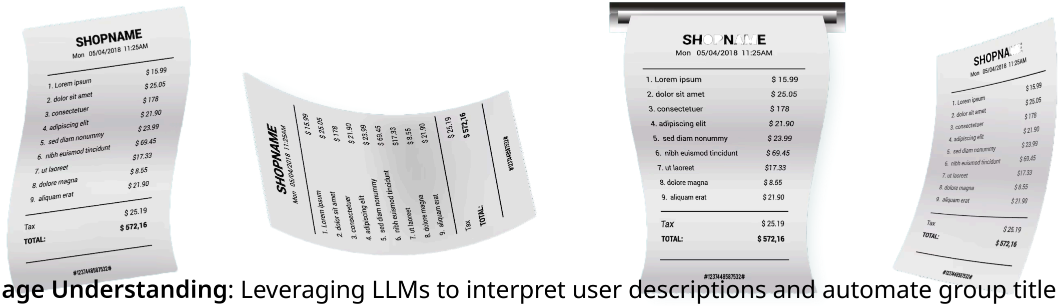
- https://aistudio.google.com/
- get started faster
- straightforward

- https://console.cloud.google.com/vertex-ai/studio/freeform
- For scaled deployments with **enterprise** ready data privacy and governance.



```python
from vertexai.generative_models import GenerativeModel
content_model = GenerativeModel("gemini-1.5-flash-001")
content_model.generate_content(
        "Explain how AI works",
        generation_config=GenerationConfig(
            response_mime_type="application/json"
        ),
    )
```

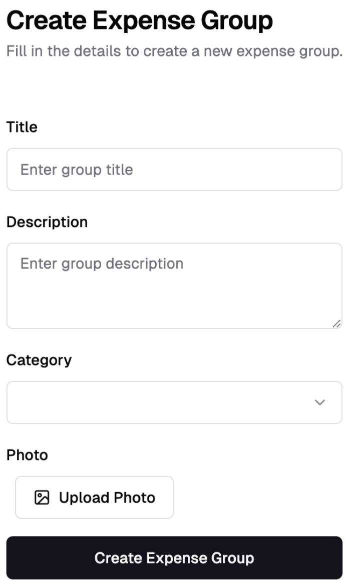
The Use Case: Enhancing User Experience



Focus Areas:

- **Natural Language Understanding**: Leveraging LLMs to interpret user descriptions and automate group title generation.
- **Text Extraction**: Using LLMs to scan and accurately itemize receipts.
- **Image Generation**: Automatically generating group-related visuals based on descriptions.
- **Output Formatting**: Structuring the extracted data into a usable format for the app.

Simplified forms (I)



Create Expense Group

Fill in the details to create a new expense group.

Title

Enter group title

Description

Enter group description

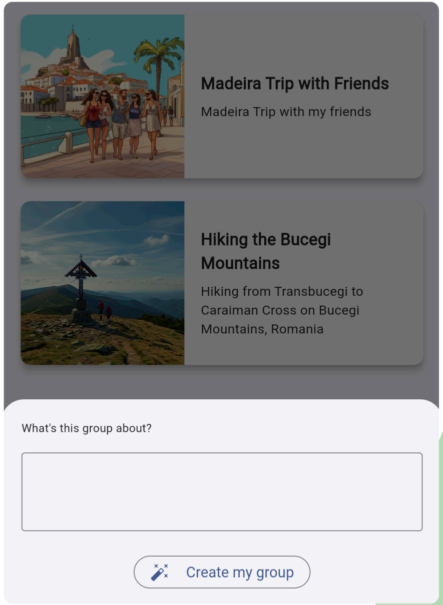
Category

Photo

Upload Photo

Create Expense Group

→



Madeira Trip with Friends

Madeira Trip with my friends

Hiking the Bucegi Mountains

Hiking from Transbucegi to Caraiman Cross on Bucegi Mountains, Romania

What's this group about?

Create my group

Simplified forms (I) - group details

```
group_details_prompt = """The following is an expenses group description: '{description}'.
            Extract a title, category and write a photo description."""
```

```
content_model.generate_content(
    group_details_prompt.format(description=description_from_form),
    generation_config=GenerationConfig()
)
```