

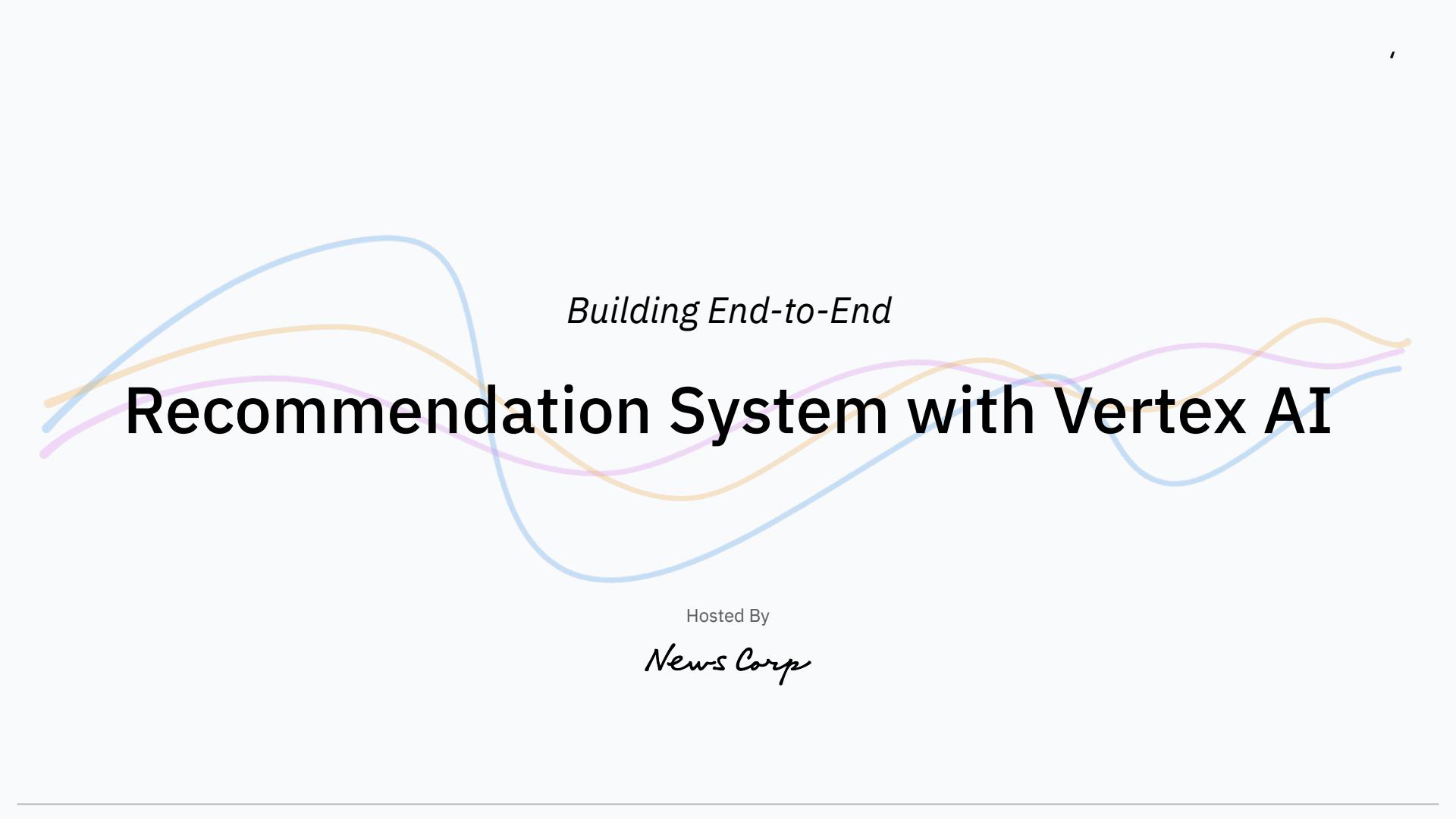


Building End-to-End
Recommendation System
with

Vertex AI

Hosted By

News Corp

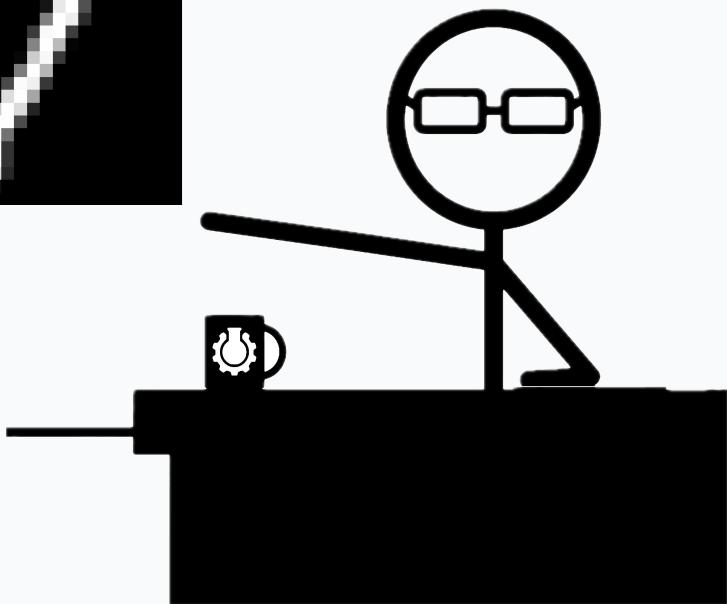
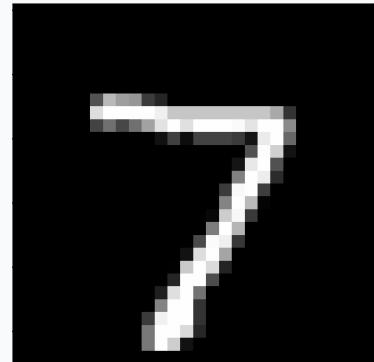
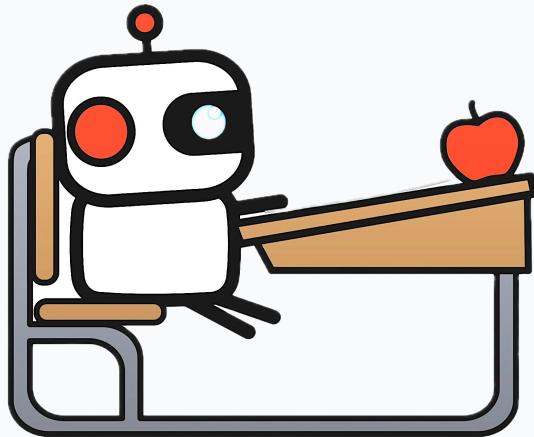


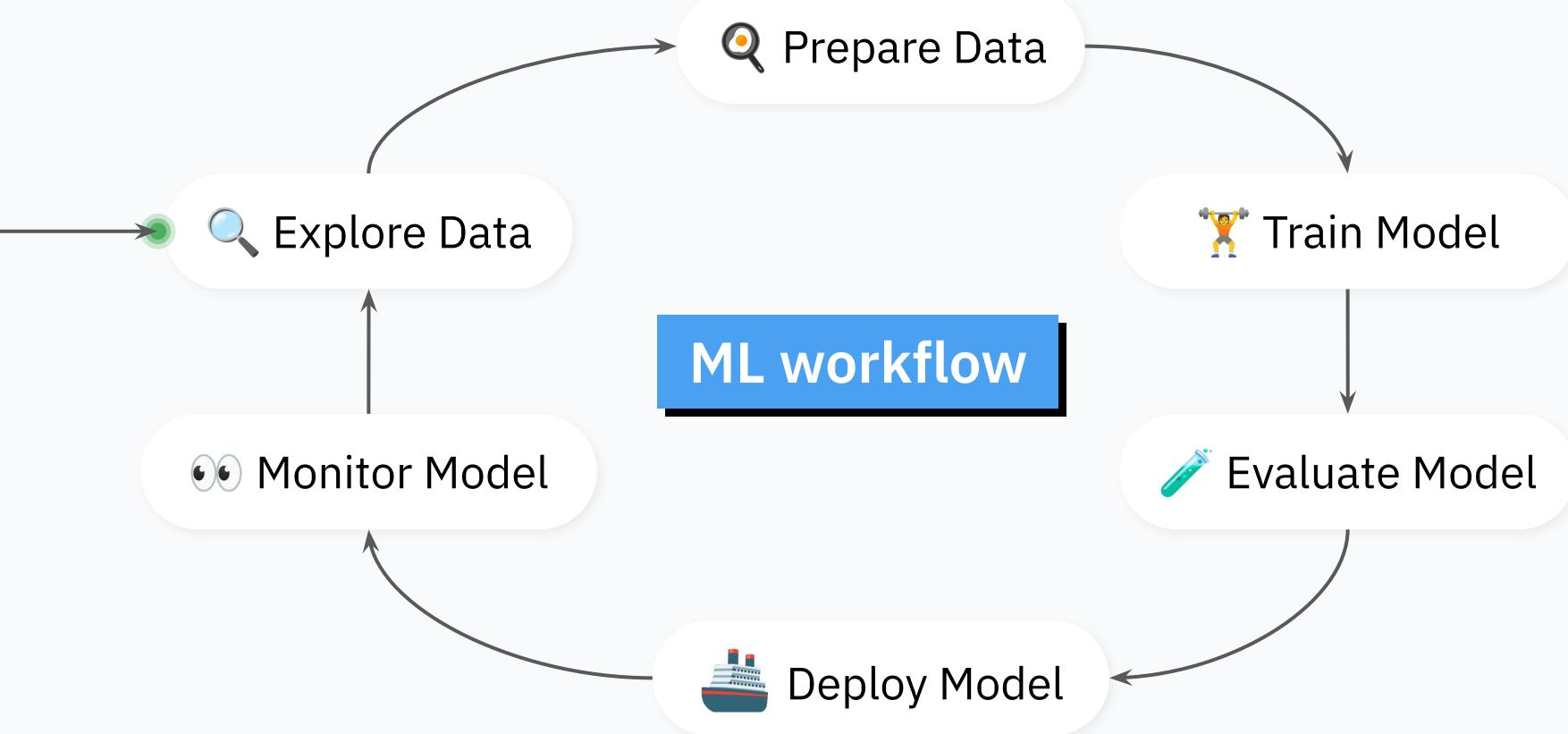
Recommendation System with Vertex AI

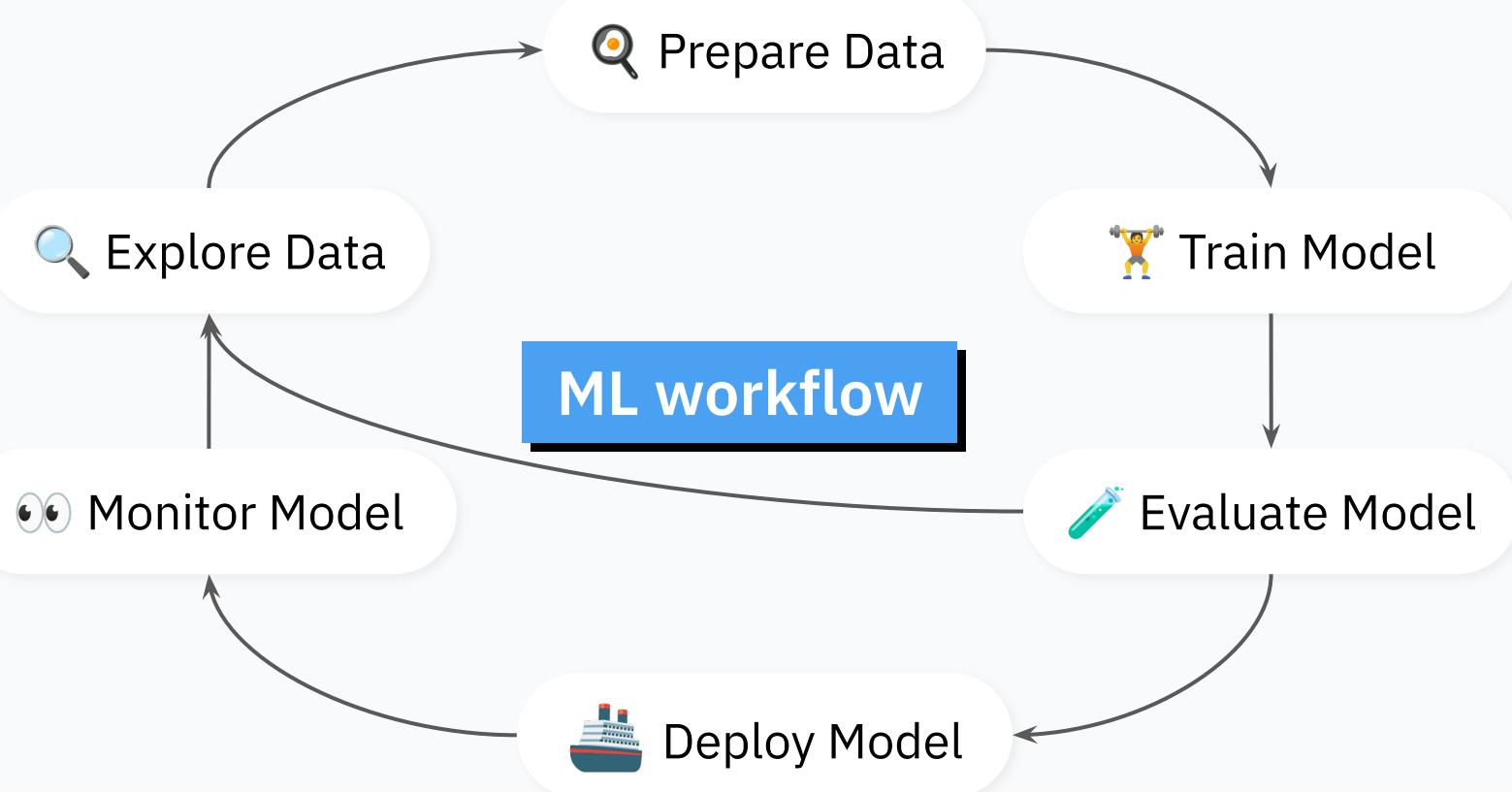
Building End-to-End

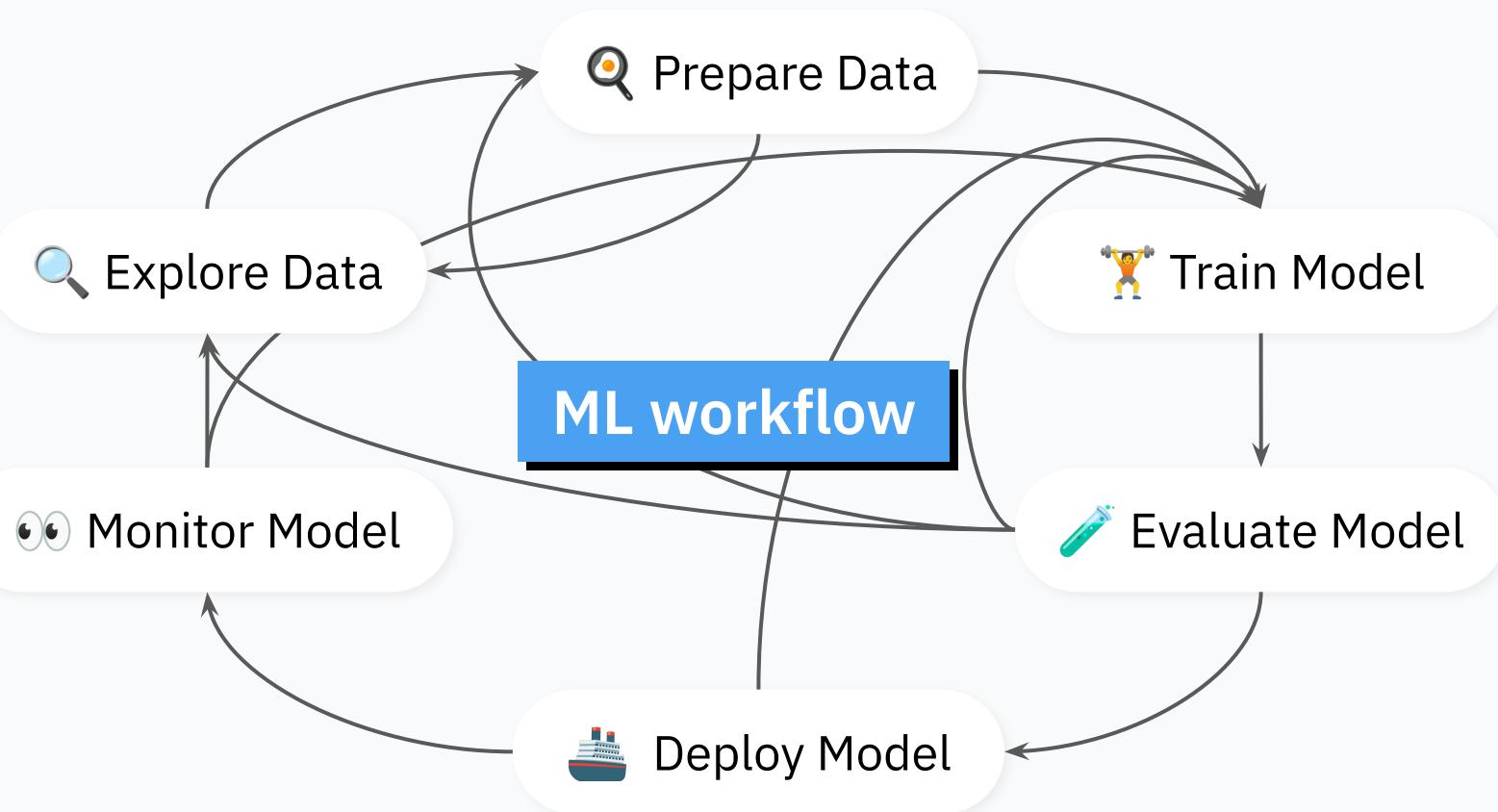
Hosted By

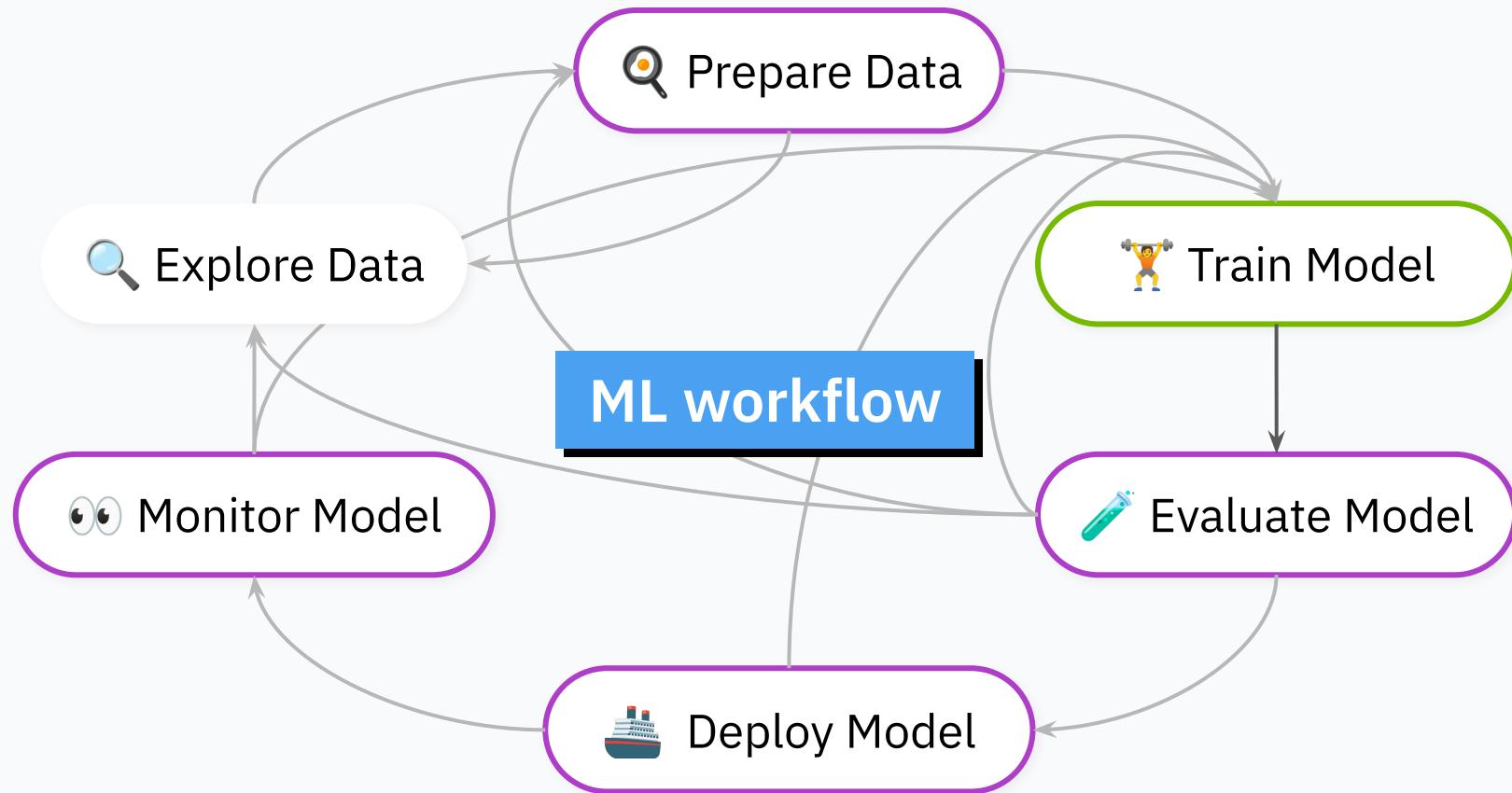
News Corp













Vertex AI



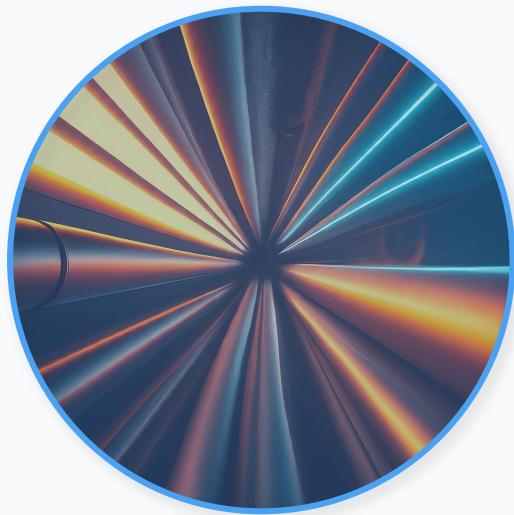
Amazon
SageMaker



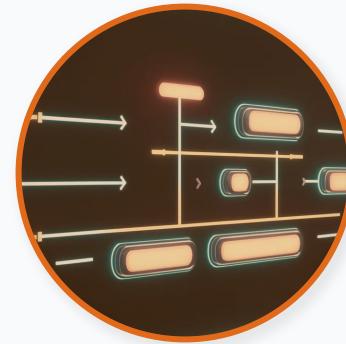
Azure Machine Learning



What is the talk about?



Vertex AI



Recommendation Systems



Pipelines



Machine Learning



Vector Search



Embeddings

Trending

Similar

Collaborative Filtering

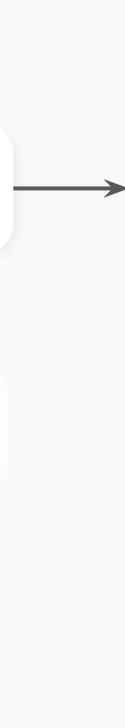
Vertex AI Pipelines

A fully managed machine learning pipeline service that allows users to orchestrate their ML workflows in a serverless manner.

Vertex AI Pipelines



Kubeflow Pipelines



Tensorflow Extended

Vertex AI Pipelines

Kubeflow Pipelines

Kubernetes

Components

Building blocks of pipelines

Takes inputs and optionally produces output

```
@component
def add(a: int, b: int) -> int:
    return a + b
```



add
python:3.7



Components

There are three kinds of Components



Lightweight Python Components



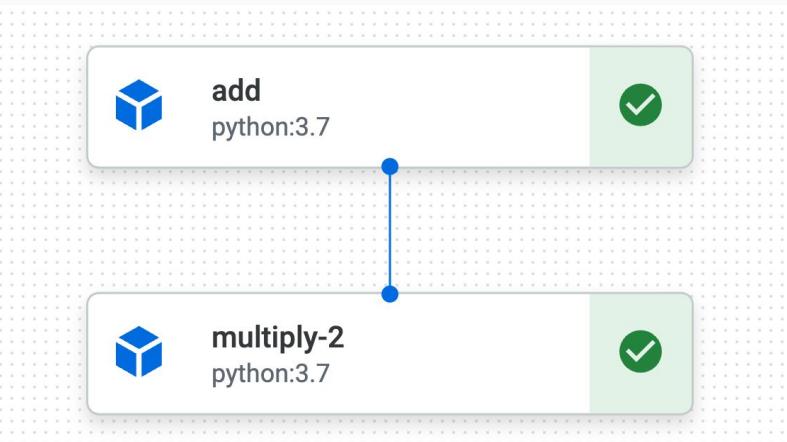
Containerized Python Components



Container Components

Pipelines

Connect components by passing the output of one component to another



```
@component
def add(a: int, b: int) -> int:
    return a + b

@component()
def multiply_2(a: int) -> int:
    return a * 2
```

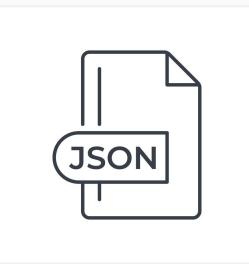
```
@pipeline
def simple_pipeline(a: int = 1, b: int = 2):
    add_task = add(a=a, b=b)
    multiply_2_task = multiply_2(a=add_task.output)
```

Compiler

Compiles the pipeline into YAML format.

The YAML file includes all the information for executing your pipeline on Vertex AI Pipelines

```
compiler.Compiler().compile(  
    pipeline_func=simple_pipeline,  
    package_path="pipelines/simple_pipeline.json"  
)
```

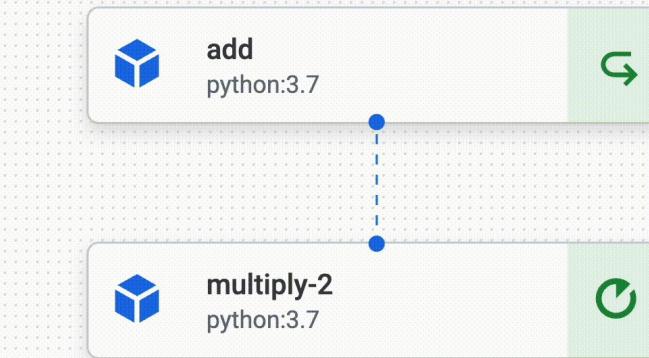


Job

Submits your pipeline run

Pass parameter values

```
job = pipeline_jobs.PipelineJob(  
    display_name="simple-pipeline",  
    template_path="pipelines/simple_pipeline.json",  
    parameter_values={"a": 43, "b": 24},  
)  
job.run()
```



Schedule Job

Runs pipeline at schedule

Create multiple schedule from
same pipeline

```
pipeline_job_schedule = aiplatform.PipelineJobSchedule(  
    pipeline_job=pipeline_job,  
    display_name="test_schedule",  
)  
  
pipeline_job_schedule.create(  
    cron="0 * * * *", # Run every hour  
)
```

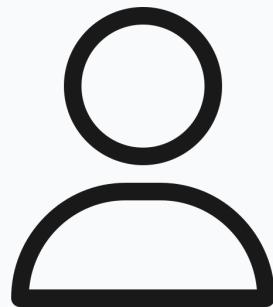


Demo

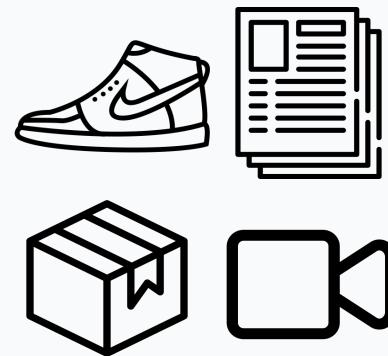


Vertex AI Pipelines

How does recommendation work ?

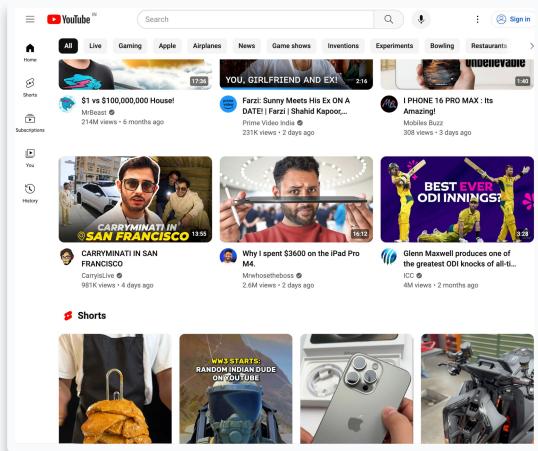
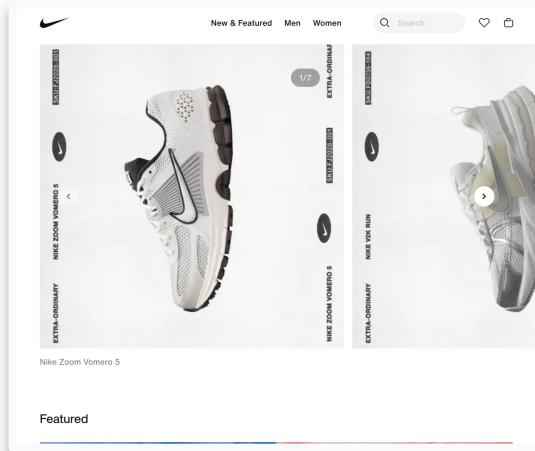


Users



Items

Where do we need recommendations ?



Signals

Indicators of user behavior and preferences



Purchase



Click



View



Impression



Time Spent



History



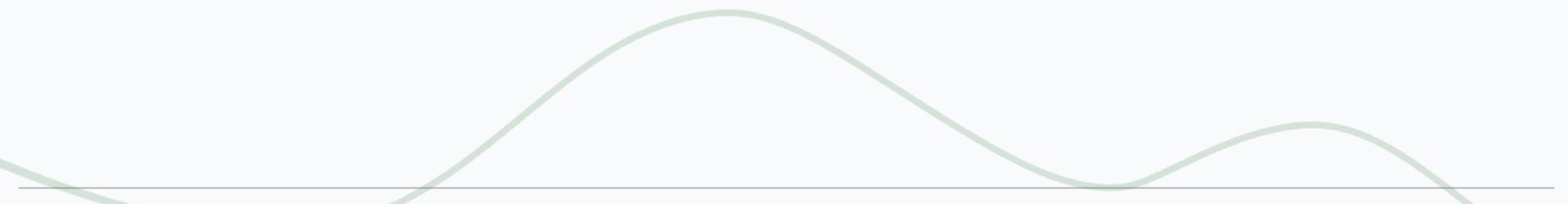
Share



Like



Comment



Signals

Indicators of user behavior and preferences



Purchase



Time Spent



History



Share



Like



Comment



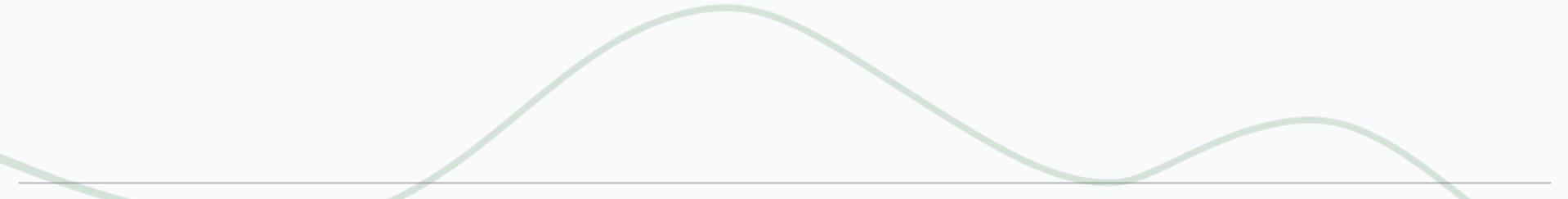
Impression



Click



View



Data

User Events

Data

user_id	item_id	event
...	...	click
...	...	view
...



Item Information

item_id	descrip...	created_dt	...
...
...



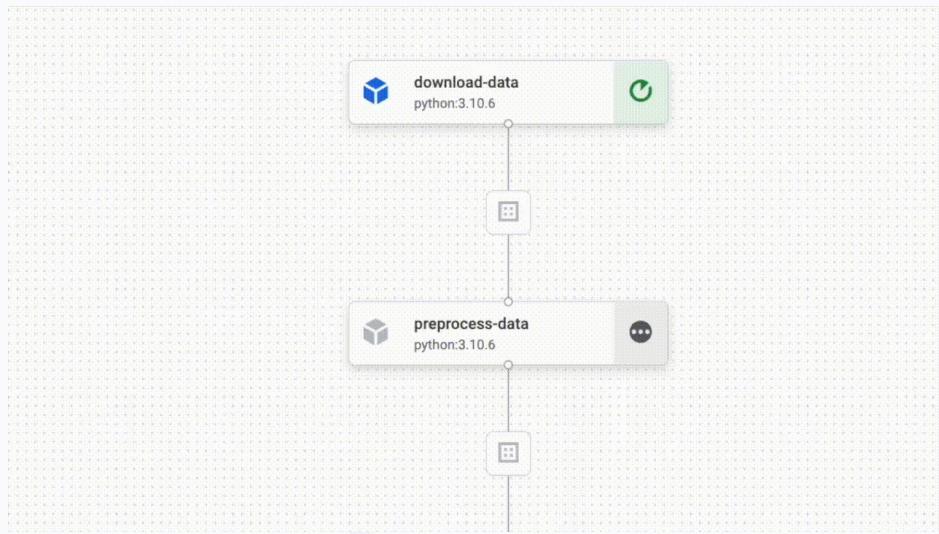
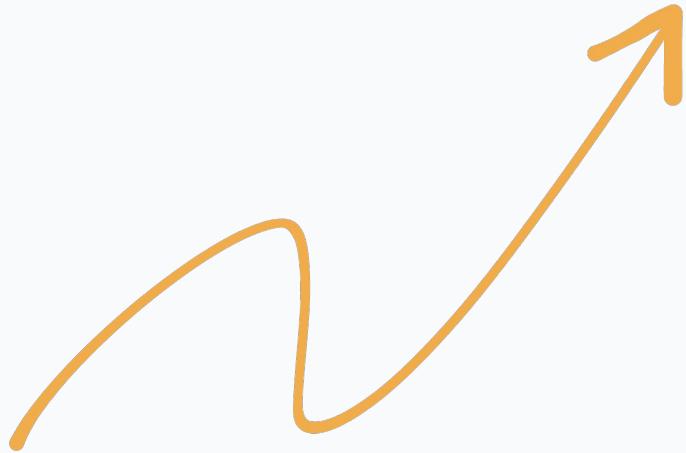
User Information

user_id	location	ads cate...	...
...
...

Trending

What is popular ?

in the next x minutes / hours



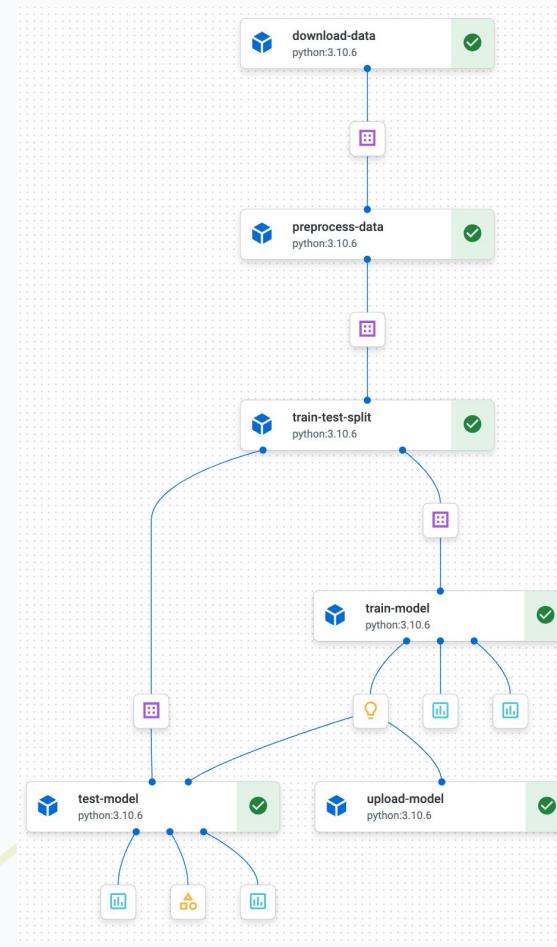
Trending

Create features

Split train-test by timestamp to avoid leakage

XGBRanker

Upload to model registry





Demo



Vertex AI Pipelines

Similar



Spatulicious: Engineered for Flipping Excellence Ultra...

¥ 650



The SlySpoon: The Spatula That's More Than Just Flip...

¥ 850



The SizzleScrape: From Sizzling Hot to Spotlessly i...

¥ 1150



The Ninja Flipper: Master of the Kitchen Flip, NOW WIT...

¥ 550

Similar



Embeddings



Spatulicious: Engineered for Flipping Excellence Ultra...

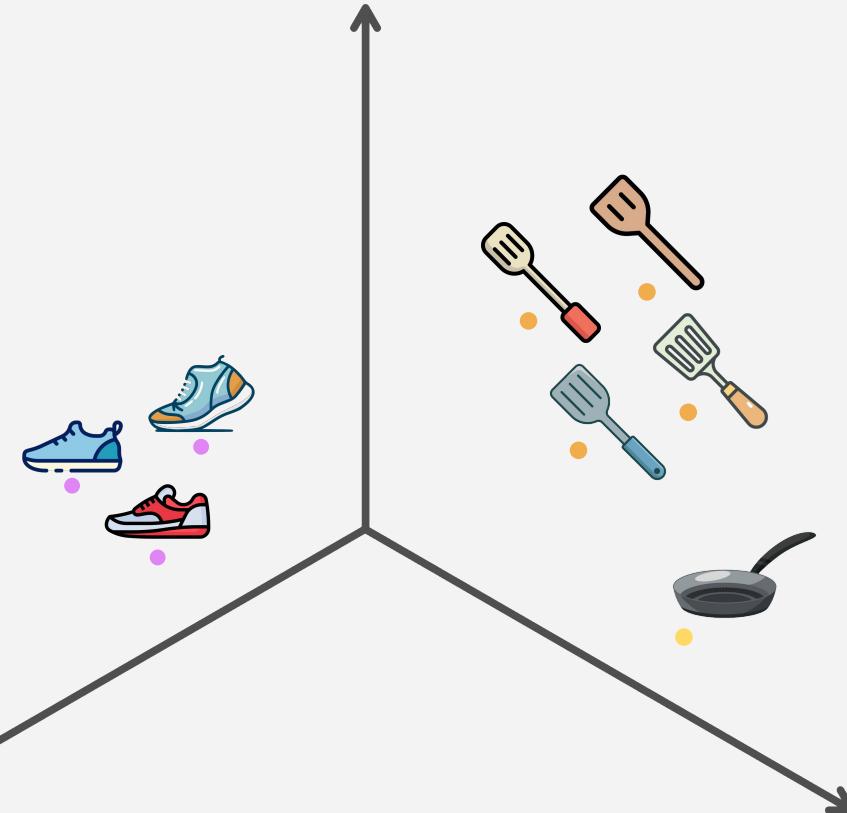
¥ 650

[0.03135875, 0.03640932,
-0.00031054, 0.04873694,
-0.03376802]

[0.00243857, -0.02919209,
-0.01841091, -0.03684188,
0.02765827]

Similar

⊕ Embeddings



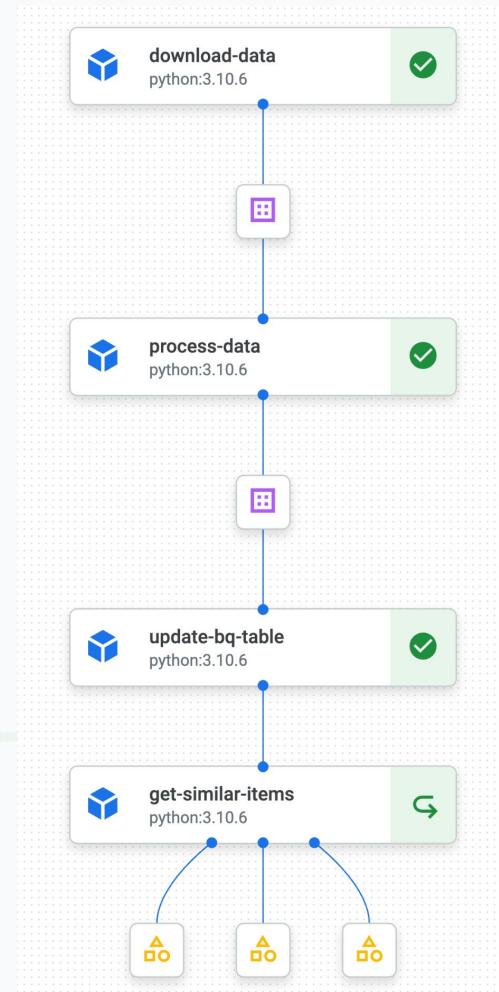
Similar

Download Image from Image URL

Create Embedding with VertexAI

Update Table

Get Similar Items





Demo



Vertex AI Embeddings



BigQuery Vector Search



Vertex AI Pipelines

Collaborative Filterings

A
B
C



Collaborative Filterings

A
B
C



Collaborative Filterings

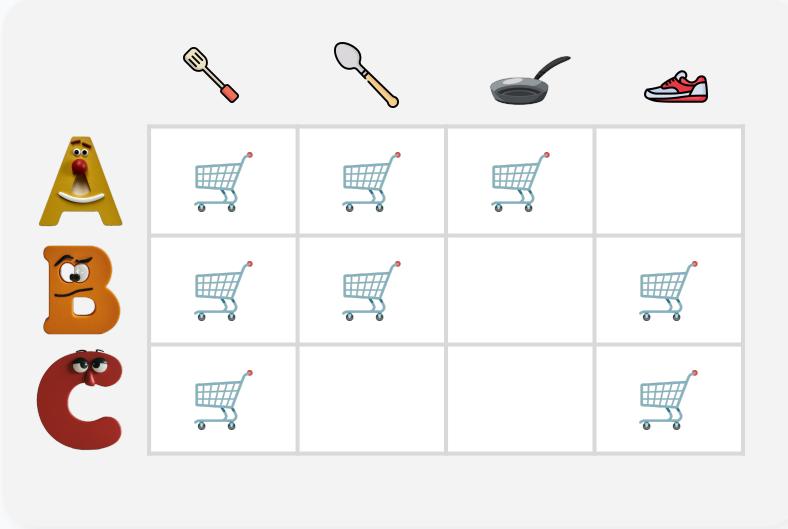
A
B
C

A				
B				
C				

Collaborative Filterings

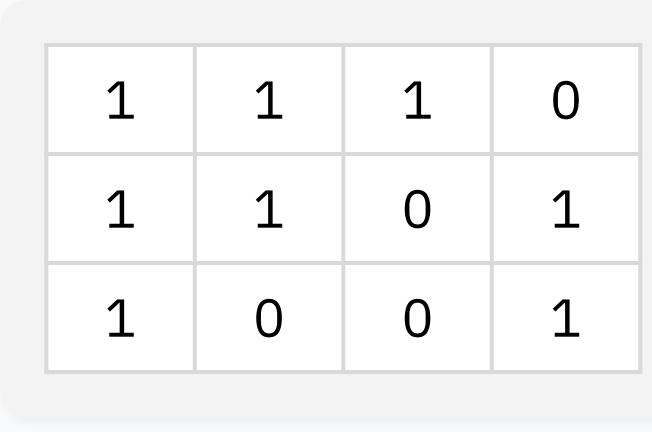


Interaction Matrix



A user-item interaction matrix diagram. On the left, three users are represented by cartoon letters: 'A' (yellow), 'B' (orange), and 'C' (red). Above them are four items: a fork, a spoon, a frying pan, and a red sneaker. The matrix itself is a 3x4 grid where each cell contains a shopping cart icon. The first two columns have icons in all cells, while the third and fourth columns have icons only in the first two rows. A large black arrow points from this matrix to the interaction matrix on the right.

	🍴	勺子	煎锅	👟
A	🛒	🛒	🛒	
B	🛒	🛒		🛒
C	🛒			🛒



An interaction matrix diagram showing user-item interactions as binary values (1 or 0). The matrix is 3x4, corresponding to the users A, B, C and items fork, spoon, pan, sneaker respectively. A value of 1 indicates an interaction (e.g., user A liked the pan), while 0 indicates no interaction (e.g., user C did not like the fork).

1	1	1	0
1	1	0	1
1	0	0	1

Collaborative Filterings



Matrix
Factorization

1	1	1	0
1	1	0	1
1	0	0	1



User Embeddings

0.8	0.2
0.6	0.4
0.7	0.1



Item Embeddings

0.6	0.5	0.4	0.2
0.1	0.3	0.2	0.7

Collaborative Filterings

0.8	0.2
0.6	0.4
0.7	0.1

0.6	0.5	0.4	0.2
0.1	0.3	0.2	0.7

 Approximate
Reconstruction

0.98	0.68	0.46	0.34
0.74	0.67	0.38	0.58
0.82	0.47	0.28	0.49

Collaborative Filterings

A				0.34
B			0.38	
C		0.47	0.28	



Demo



BigQuery ML



Vertex AI Pipelines

Santosh Rajan

Senior Data Scientist
News Corp

X @nottheralsanta

bit.ly/aicamp-newscorp

