

Slides That are used as screenshots in the github/statmike/vertex-ai-mlops repository

Notes:

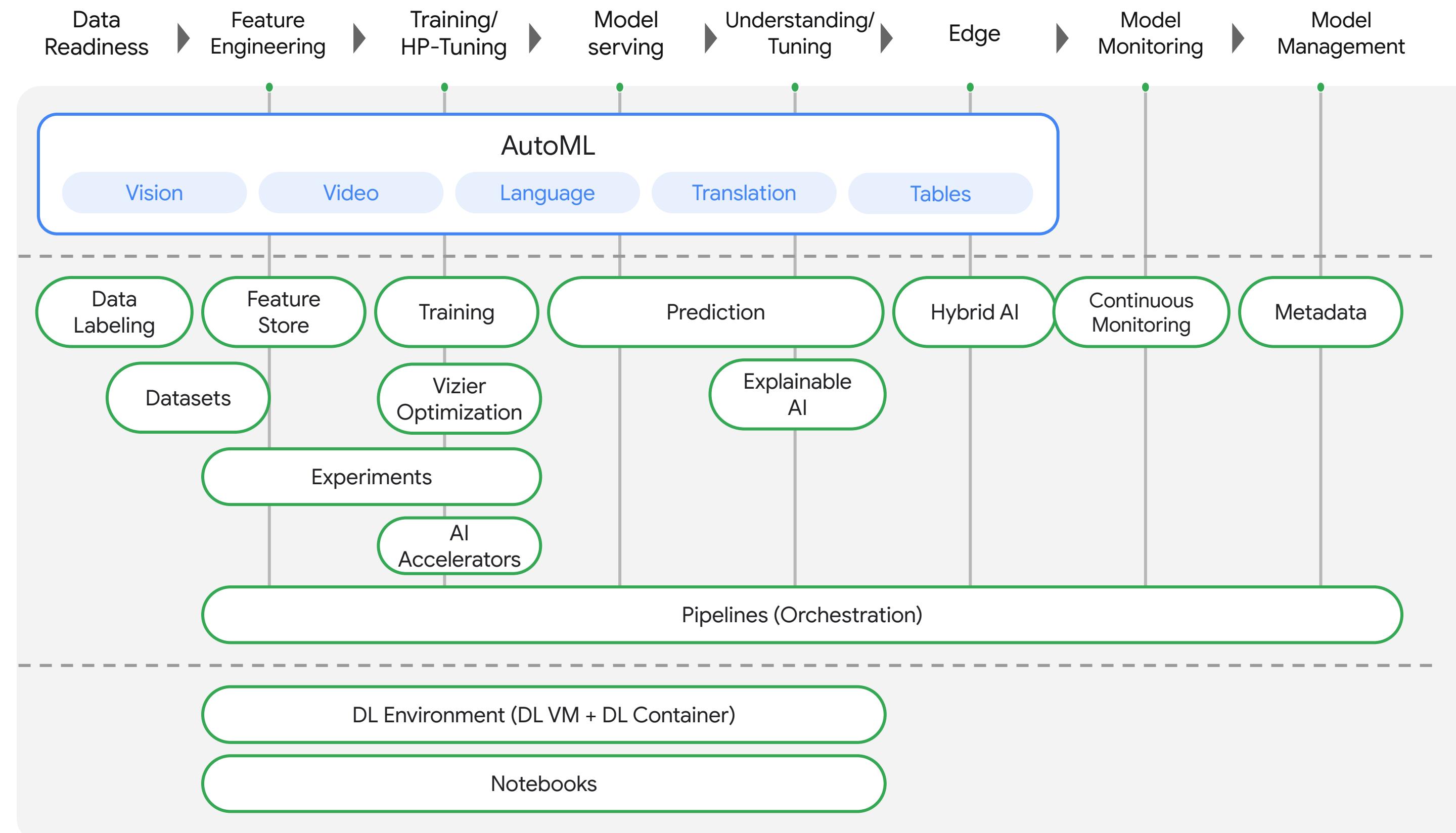
- Do not insert new slides or reorder without updating the notebooks. The slides are exported to numbered .png files that are referenced in the notebooks

Process

- Save as PDF
- Copy to `github/statmike/vertex-ai-mlops/architectures/slides`
- Convert PDF to PNG images
 - Use Notebook: `/architectures/Create Images.ipynb`
 - OUTPUT:
 - To `/architectures/slides`

README

Vertex AI Overview



Vertex AI

Dashboard

Dashboard

Datasets

Features

Labeling tasks

Notebooks

Pipelines

Training

Experiments

Models

Endpoints

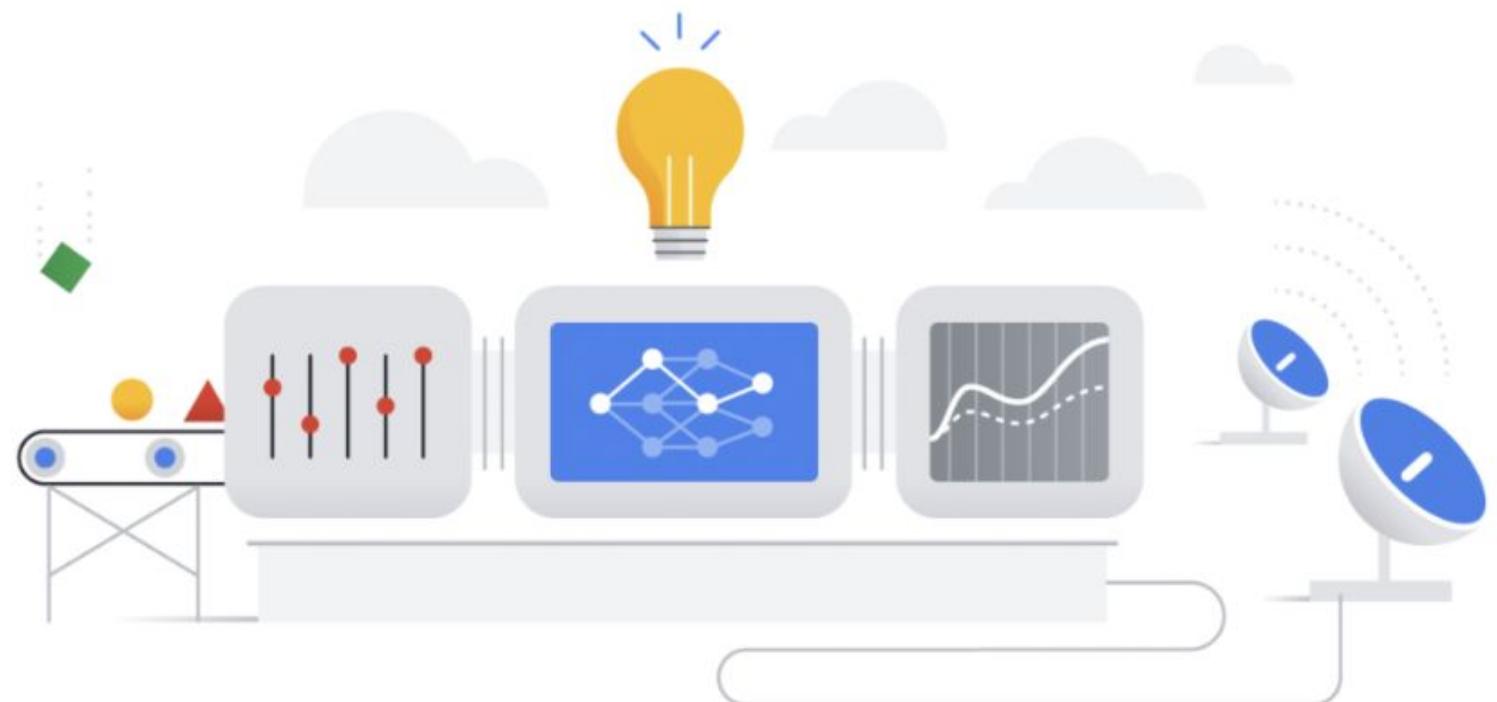
Batch predictions

Metadata

Marketplace

Get started with Vertex AI

Vertex AI empowers machine learning developers, data scientists, and data engineers to take their projects from ideation to deployment, quickly and cost-effectively. [Learn more](#)



Region

us-central1 (Iowa)



Recent datasets

- ✓ 02c_digits_20210919213805 16 hours ago
- ✓ 02b_digits_20210919205707 20 hours ago
- ✓ 02a 4 days ago
- ✓ 02b_digits_20210916141540 4 days ago
- ✓ 02c_digits_20210916004500 5 days ago

[+ CREATE DATASET](#)

Recent models

- ✓ 05f_digits_20210920145828 1 hour ago
- ✓ 05e_digits_20210920125450 3 hours ago
- ✓ 02c_digits_20210919213805 12 hours ago
Average precision: 1
- ✓ 02b_digits_20210919205707 19 hours ago
Average precision: 1
- ✓ 05c_digits_20210919214125-model 19 hours ago

[+ TRAIN NEW MODEL](#)

Get predictions

After you train a model, you can use it to get predictions, either online as an endpoint or through batch requests

[+ CREATE BATCH PREDICTION](#)[Show debug panel](#)



Vertex AI



readme.md

Vertex AI for Machine Learning Operations

I'm Mike

I want to share and enable Vertex AI from Google Cloud with you. The goal here is to share a comprehensive set of end-to-end workflows for machine learning that each cover the range of data to model to serving and managing - even automating the flow. Regardless of your data type, skill level or framework preferences you will find something helpful here.

Considerations

Data Type

- Tables: Tabular, structured data in rows and columns
- Language: Text for translation and/or understanding
- Vision: Images
- Video

Convenience Level

- Use Pre-Trained APIs
- Automate building Custom Models
- End-to-end Custom ML with core tools in the framework of your choice

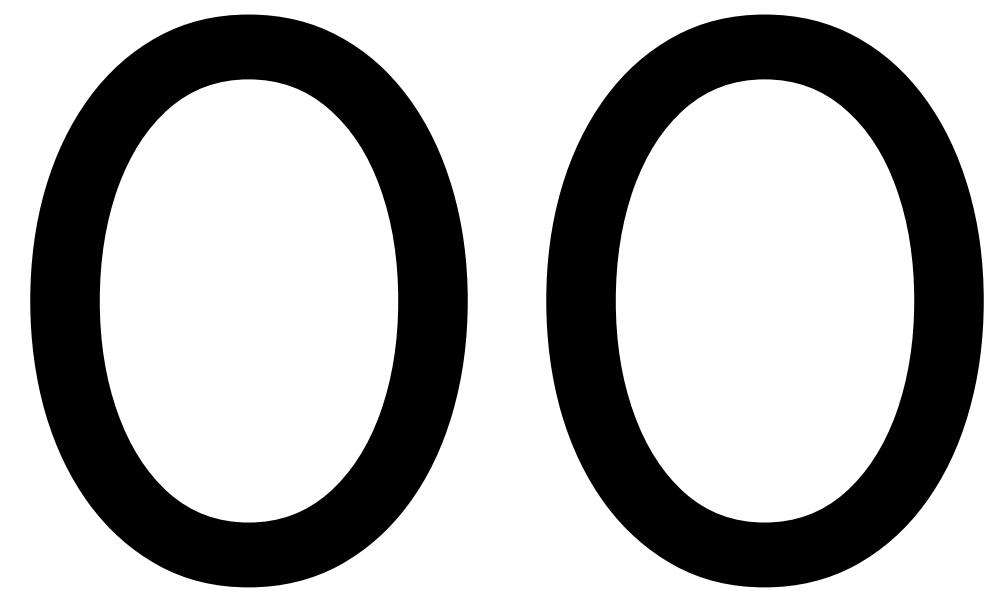
Framework Preferences

- Scikit-learn
- XGBoost
- Tensorflow
- Pytorch
- More!

Overview

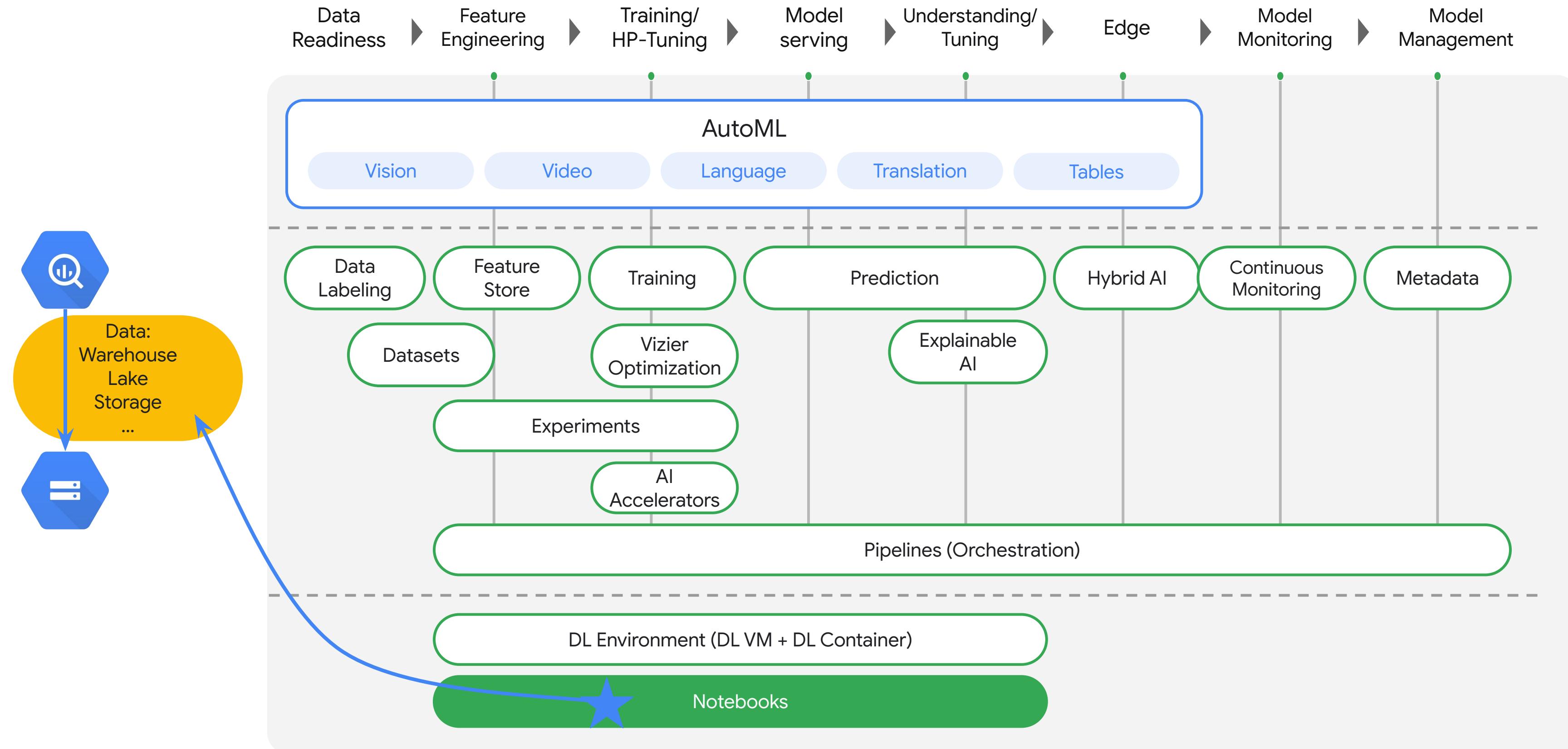
This is a series of workflow demonstrations that use the same data source to build and deploy the same machine

Introduction



Notebook: 00

Vertex AI Overview



Vertex AI

Notebooks

NEW INSTANCE

REFRESH

START

STOP

RESET

DELETE

SHOW INFO PANEL

Dashboard

MANAGED NOTEBOOKS PREVIEW

INSTANCES

EXECUTIONS PREVIEW

SCHEDULES PREVIEW

SCHEDULED RUNS

Datasets

Create and use Jupyter Notebooks with a notebook instance. Notebook instances have JupyterLab pre-installed and are configured with GPU-enabled machine learning frameworks. [Learn more](#)

Labeling tasks

Filter Enter property name or value

Notebooks

	Instance name	Zone	Auto-upgrade	Environment	Machine type	GPUs	Permission	Last modified
	mlops2	OPEN JUPYTERLAB	us-central1-a	TensorFlow:2.3	4 vCPUs, 15 GB RAM	None	Service account	Sep 15, 2021, 10:05:03 PM

Pipelines

Training

Experiments

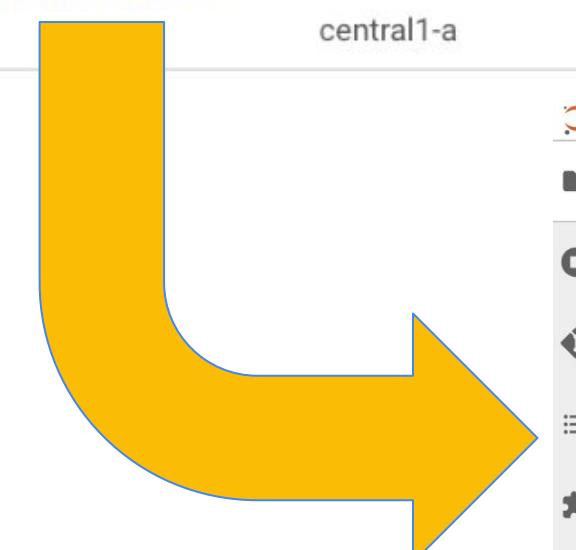
Models

Endpoints

Batch predictions

Metadata

Marketplace



```
File Edit View Run Kernel Git Tabs Settings Help
+ Filter files by name
/ vertex-ai-mlops/
Name Last Modified
architectures 9 minutes ago
Dev 9 minutes ago
00 - Environment Setu... 2 minutes ago
01 - BigQuery - Table ... 9 minutes ago
02a - Vertex AI - Auto... 9 minutes ago
02b - Vertex AI - Auto... 9 minutes ago
02c - Vertex AI > Pipel... 9 minutes ago
03a - BigQuery Machi... 9 minutes ago
03b - Vertex AI + BQ... 9 minutes ago
04a - Vertex AI > Note... 9 minutes ago
05 - Vertex AI > Exper... 9 minutes ago
05a - Vertex AI > Train... 9 minutes ago
05b - Vertex AI > Trai... 9 minutes ago
05c - Vertex AI > Train... 9 minutes ago
05d - Vertex AI > Trai... 9 minutes ago
05e - Vertex AI > Train... 9 minutes ago
05f - Vertex AI > Train... 9 minutes ago
06 - Vertex AI > Exper... 9 minutes ago
07 - Vertex AI > Featu... 9 minutes ago
LICENSE 9 minutes ago
readme.md 9 minutes ago
XX - Cleanup.ipynb 9 minutes ago

Launcher 00 - Environment Setup.ipynb
File Edit View Run Kernel Git Tabs Settings Help
+ Filter files by name
/ vertex-ai-mlops/
Name Last Modified
architectures 9 minutes ago
Dev 9 minutes ago
00 - Environment Setu... 2 minutes ago
01 - BigQuery - Table ... 9 minutes ago
02a - Vertex AI - Auto... 9 minutes ago
02b - Vertex AI - Auto... 9 minutes ago
02c - Vertex AI > Pipel... 9 minutes ago
03a - BigQuery Machi... 9 minutes ago
03b - Vertex AI + BQ... 9 minutes ago
04a - Vertex AI > Note... 9 minutes ago
05 - Vertex AI > Exper... 9 minutes ago
05a - Vertex AI > Train... 9 minutes ago
05b - Vertex AI > Trai... 9 minutes ago
05c - Vertex AI > Train... 9 minutes ago
05d - Vertex AI > Trai... 9 minutes ago
05e - Vertex AI > Train... 9 minutes ago
05f - Vertex AI > Train... 9 minutes ago
06 - Vertex AI > Exper... 9 minutes ago
07 - Vertex AI > Featu... 9 minutes ago
LICENSE 9 minutes ago
readme.md 9 minutes ago
XX - Cleanup.ipynb 9 minutes ago

Setup
inputs:
[1]: REGION = 'us-central1'
PROJECT_ID = 'statmike-mlops'
DATANAME = 'fraud'

# Data source for this series of notebooks: Described in notebook 01
BQ_SOURCE = 'bigquery-public-data.ml_datasets.ulb_fraud_detection'

packages:
[2]: from google.cloud import storage
from google.cloud import bigquery

import pandas as pd
from sklearn import datasets

parameters:
[3]: BUCKET = PROJECT_ID

Create Storage Bucket
[4]: gcs = storage.Client(project = PROJECT_ID)

[5]: if not gcs.lookup_bucket(BUCKET):
    bucketDef = gcs.bucket(BUCKET)
```



Vertex AI

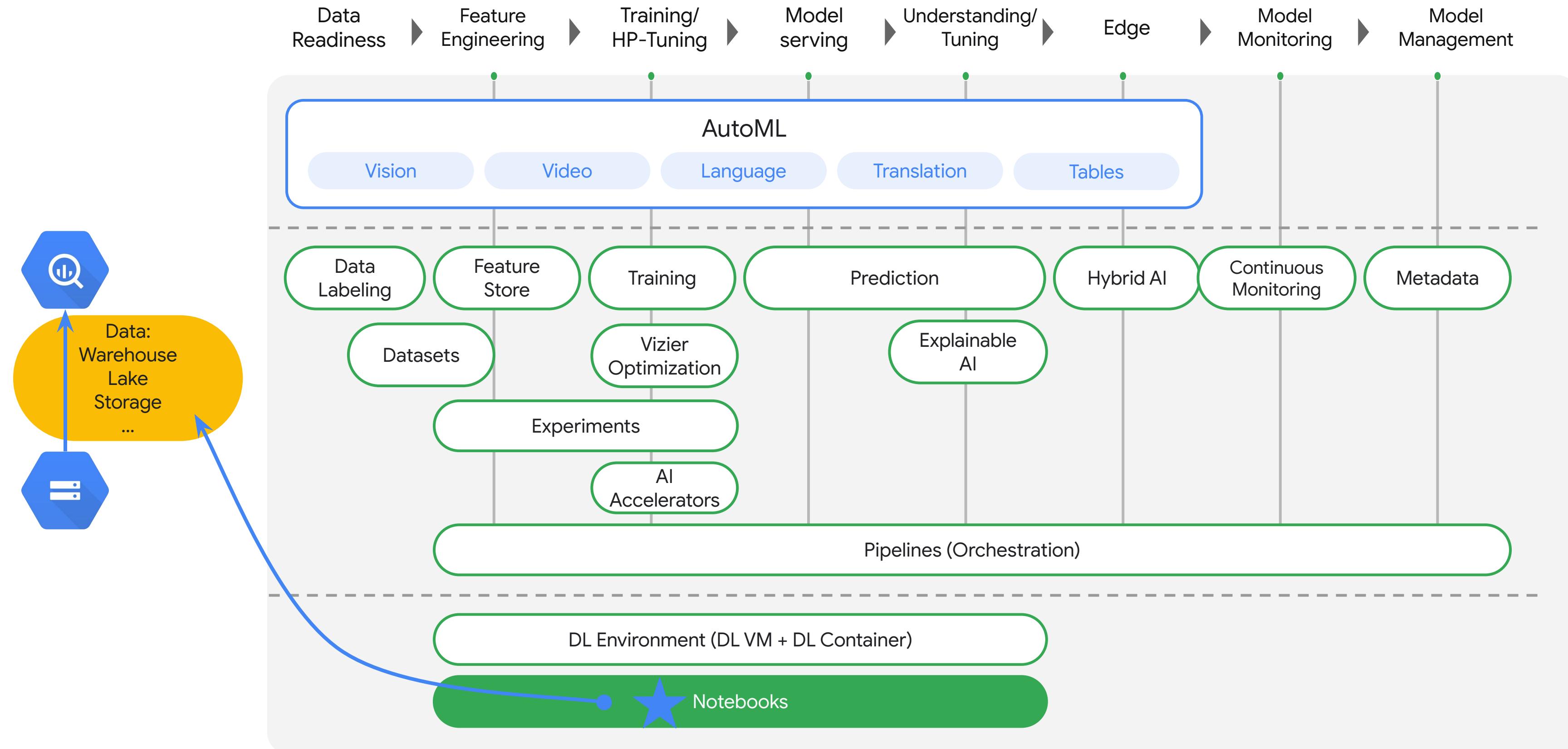


Environment Setup

01

Notebook: 01

Vertex AI Overview



Google Cloud Platform

Vertex AI

Dashboard

Datasets

Features

Labeling tasks

Notebooks

Pipelines

Training

Experiments

Models

Endpoints

Batch predictions

Metadata

Marketplace

The screenshot shows a Jupyter Notebook environment within Google Colab. The notebook title is "01 - BigQuery - Table Data Source". The content discusses using BigQuery to load and prepare data for machine learning, listing prerequisites (Environment Setup) and an overview of BigQuery setup, table creation, and data loading from GCS. A large yellow arrow points from the "Notebooks" menu in the sidebar to the "01 - BigQuery - Table Data Source" tab in the notebook interface.

01 - BigQuery - Table Data Source

Use BigQuery to load and prepare data for machine learning:

Prerequisites:

- 00 - Environment Setup

Overview:

- Setup BigQuery
 - Create a Dataset
 - Use BigQuery Python Client
 - Create Tables
 - Copy from another Project:Dataset
 - SQL with BigQuery
 - Load data from GCS
 - BigQuery Python
 - Prepare Data For Analysis
 - Run SQL Queries to pi

Resources:

- Python Client For Google BigQ
- Download BigQuery Data to Pa
- Query Template Notebooks

0 19 Git: idle Python 3 | Idle

File Edit View Run Kernel Git Tabs Settings Help

Launcher 01 - BigQuery - Table Data git Python 3

00 - Environme... 3 days ago

01 - BigQuery - Table Data Source 3 days ago

02a - Vertex AI ... 4 days ago

02b - Vertex AI ... 8 hours ago

02c - Vertex AI ... 8 hours ago

03a - BigQuery ... 4 days ago

03b - Vertex AI ... 8 hours ago

04a - Vertex AI ... 8 hours ago

05 - Vertex AI >... 20 hours ago

05a - Vertex AI ... 8 hours ago

05b - Vertex AI ... 4 hours ago

05c - Vertex AI ... 4 hours ago

05d - Vertex AI ... 4 hours ago

05e - Vertex AI ... 3 hours ago

05f - Vertex AI ... an hour ago

06 - Vertex AI >... 4 days ago

07 - Vertex AI >... 4 days ago

readme.md 5 days ago

XX - Cleanup.ip... 5 days ago

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

Explorer + ADD DATA

Type to search

Viewing pinned projects.

statmike-mlops

digits

Models (1)

digits

digits_featurestore_import

digits_fs_training

digits_prepred

Row	p0	p1	p2	p3	p4	p5	p6	p7	p8	p
1	0.0	5.0	16.0	15.0	5.0	0.0	0.0	0.0	0.0	1
2	0.0	5.0	16.0	12.0	1.0	0.0	0.0	0.0	0.0	1
3	0.0	5.0	15.0	16.0	6.0	0.0	0.0	0.0	0.0	1
4	0.0	4.0	15.0	15.0	8.0	0.0	0.0	0.0	0.0	0.0
5	0.0	6.0	16.0	16.0	15.0	10.0	0.0	0.0	0.0	0.0
6	0.0	8.0	16.0	12.0	15.0	16.0	7.0	0.0	0.0	1
7	0.0	8.0	13.0	15.0	16.0	16.0	8.0	0.0	0.0	0.0
8	0.0	7.0	12.0	14.0	16.0	8.0	0.0	0.0	0.0	0.0



Vertex AI

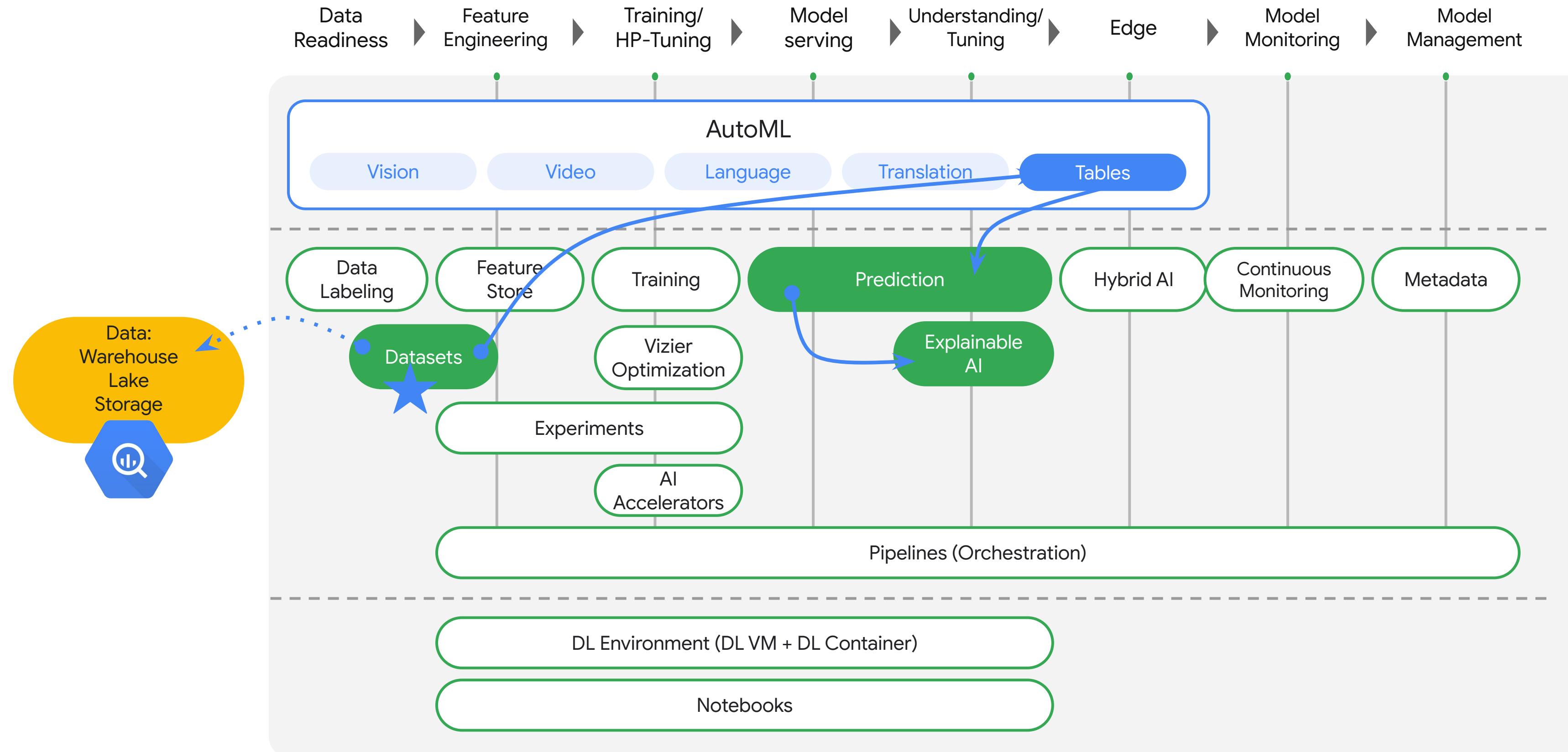


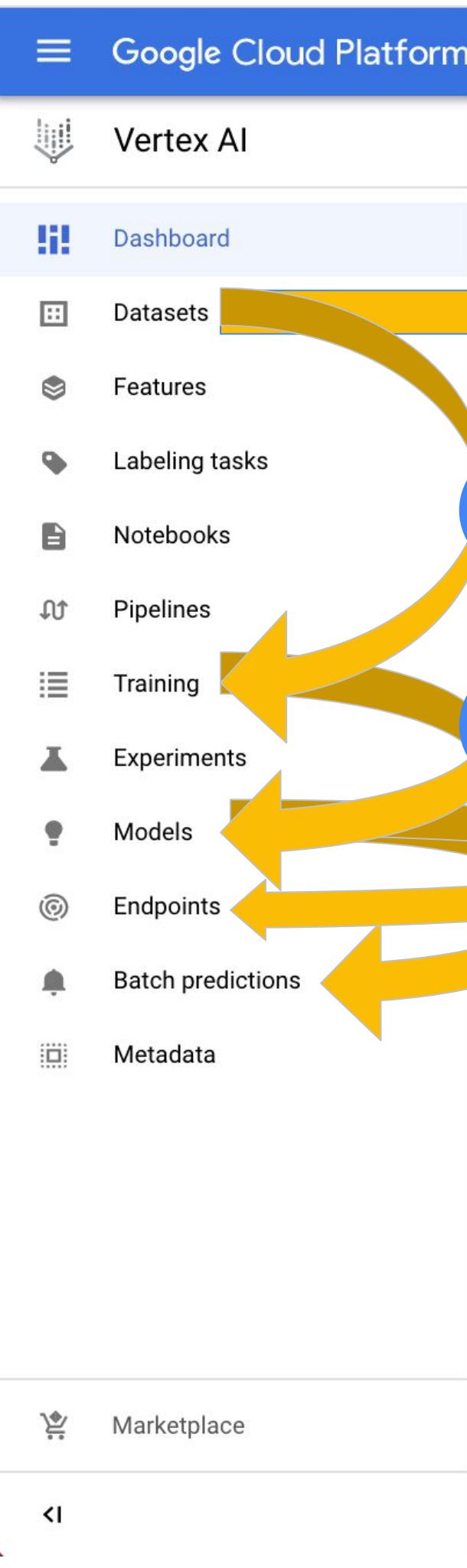
Data Source

02a

Notebook: 02a

Vertex AI Overview





The screenshot shows the Google Cloud Platform Explorer interface. At the top, the project is set to 'statmike-mlops'. The main area displays the 'digits' dataset under the 'DIGITS' tab. The 'PREVIEW' tab is selected, showing a table with 8 rows and 10 columns. The columns are labeled p0, p1, p2, p3, p4, p5, p6, p7, p8, and p. The data rows are as follows:

Row	p0	p1	p2	p3	p4	p5	p6	p7	p8	p
1	0.0	5.0	16.0	15.0	5.0	0.0	0.0	0.0	0.0	
2	0.0	5.0	16.0	12.0	1.0	0.0	0.0	0.0	0.0	
3	0.0	5.0	15.0	16.0	6.0	0.0	0.0	0.0	0.0	1
4	0.0	4.0	15.0	15.0	8.0	0.0	0.0	0.0	0.0	
5	0.0	6.0	16.0	16.0	16.0	15.0	10.0	0.0	0.0	
6	0.0	8.0	16.0	12.0	15.0	16.0	7.0	0.0	0.0	1
7	0.0	8.0	13.0	15.0	16.0	16.0	8.0	0.0	0.0	
8	0.0	7.0	12.0	14.0	16.0	8.0	0.0	0.0	0.0	

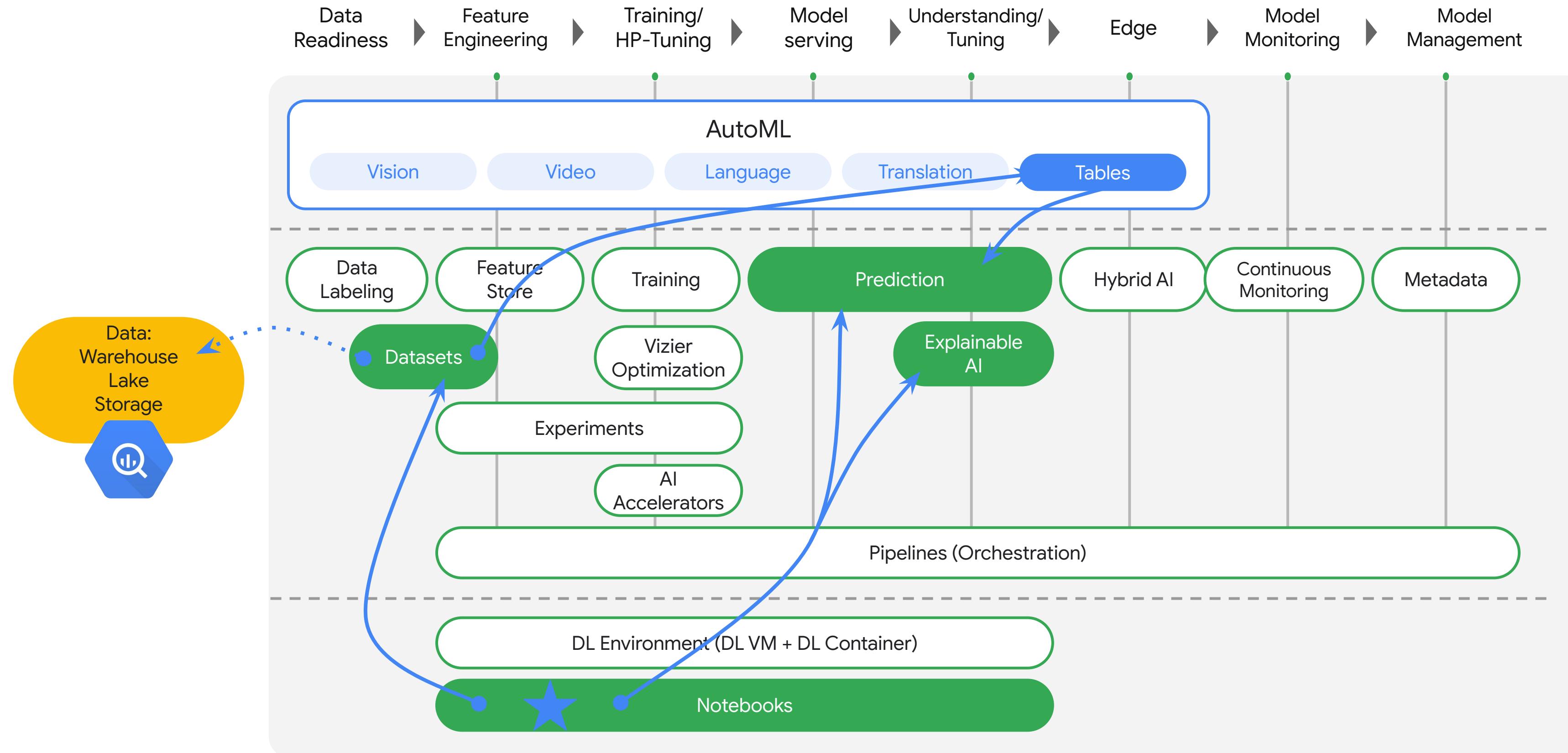


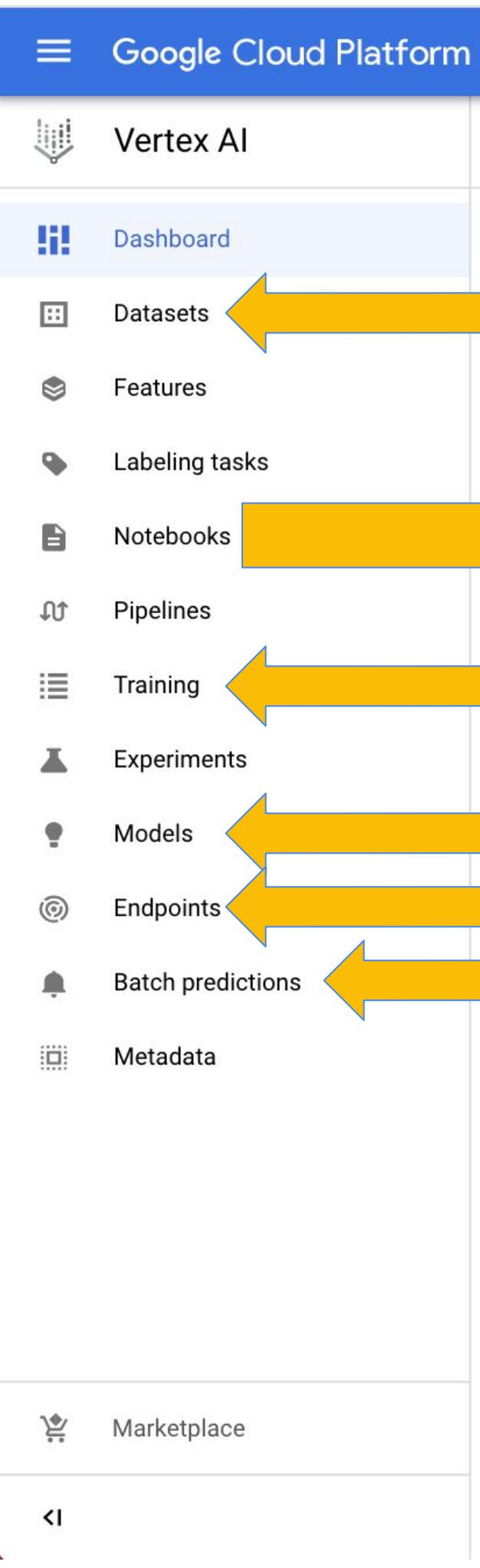
Vertex AI



End-To-End: No Code

02b





File Edit View Run Kernel Git Tabs Settings Help

Launcher **02b - Vertex AI - AutoML v**

Python 3

02b - Vertex AI - AutoML with clients (code)

Use the Vertex AI Python Client to recreate the no-code approach of (02a) with code (Python). This builds a custom model with AutoML and deploys it to an Endpoint for predictions and explanations.

Prerequisites:

- 01 - BigQuery - Table Data Source

Overview:

- Use Python client `google.cloud.aiplatform` for Vertex AI
 - Create a dataset
 - `aiplatform.TabularDataset`
 - Link BigQuery table
 - Train Model with AutoML
 - `aiplatform.AutoMLTabularTrainingJob`
 - Evaluate
 - Review the model in GCP Console > Vertex AI > Models
 - Deploy to Endpoint
 - `Endpoint = aiplatform.Endpoint`
 - `Endpoint.deploy`
 - Online Predictions
 - `Endpoint.predict`

Mode: Command **Ln 1, Col 1** 02b - Vertex AI - AutoML with clients (code).ipynb

Name	Last Modified
architectures	2 hours ago
Dev	2 days ago
temp	4 hours ago
01 - BigQuery - ...	3 days ago
02a - Vertex AI ...	4 days ago
02b - Vertex AI ...	9 hours ago
02c - Vertex AI ...	9 hours ago
03a - BigQuery ...	4 days ago
03b - Vertex AI ...	9 hours ago
04a - Vertex AI ...	9 hours ago
05 - Vertex AI >...	a day ago
05a - Vertex AI ...	9 hours ago
05c - Vertex AI ...	5 hours ago
05d - Vertex AI ...	5 hours ago
05e - Vertex AI ...	4 hours ago
06 - Vertex AI >...	2 hours ago
readme.md	4 days ago
requirements.ip...	4 days ago
M README	5 days ago
run.ip...	5 days ago

0 \$ 19 Git: idle Python 3 | Idle

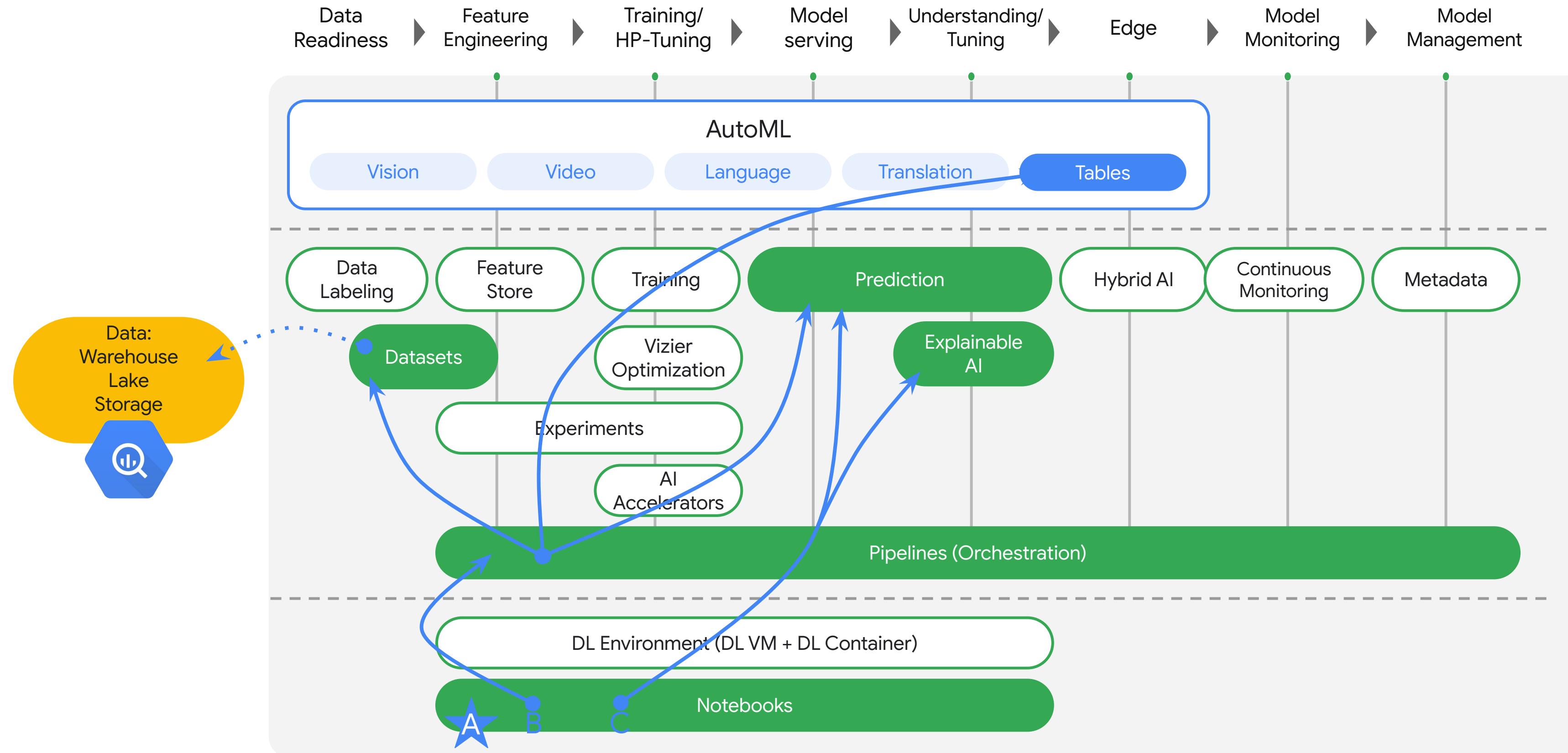


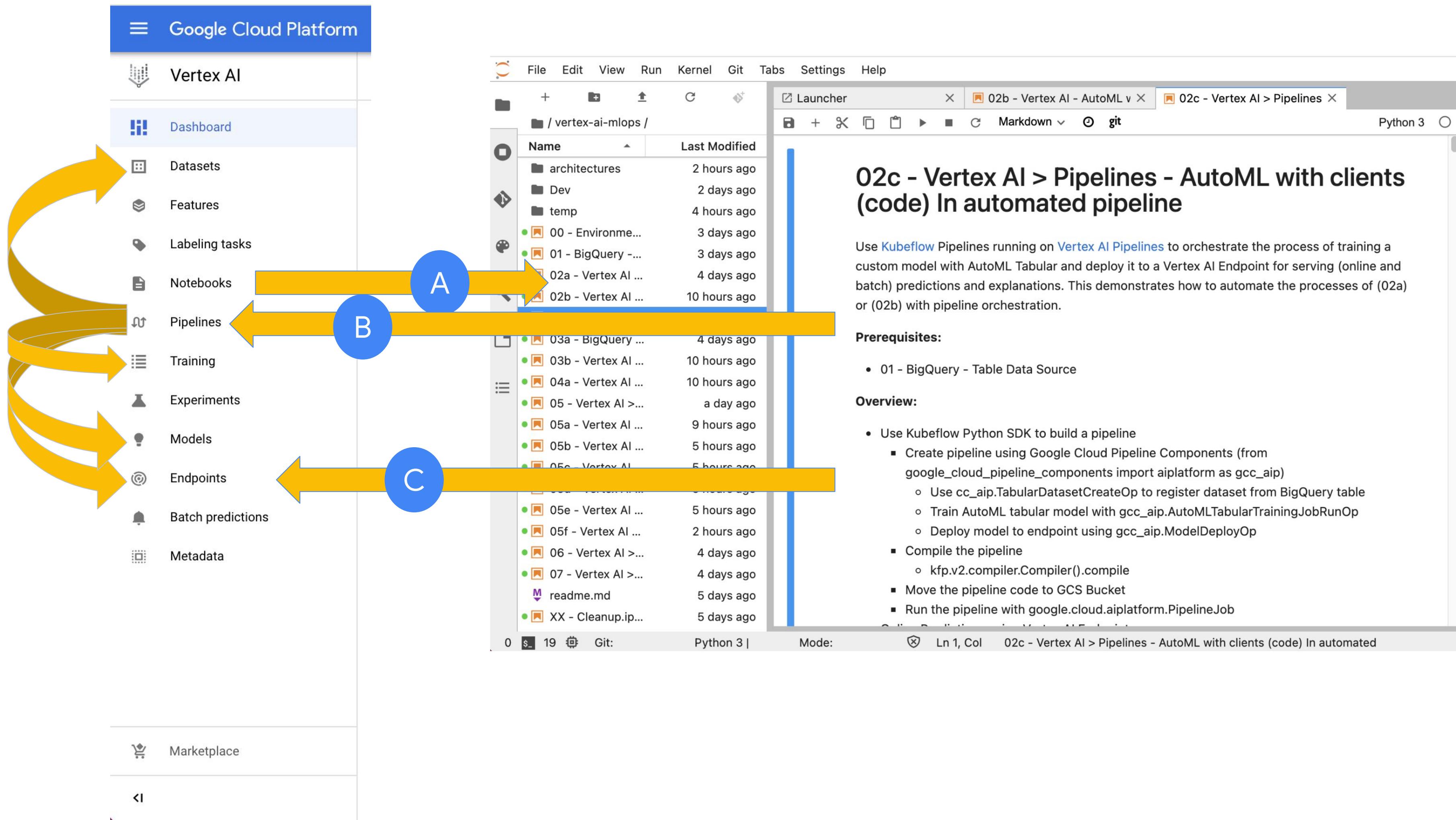
Vertex AI



End-To-End: Interactive Code

02c







Vertex AI

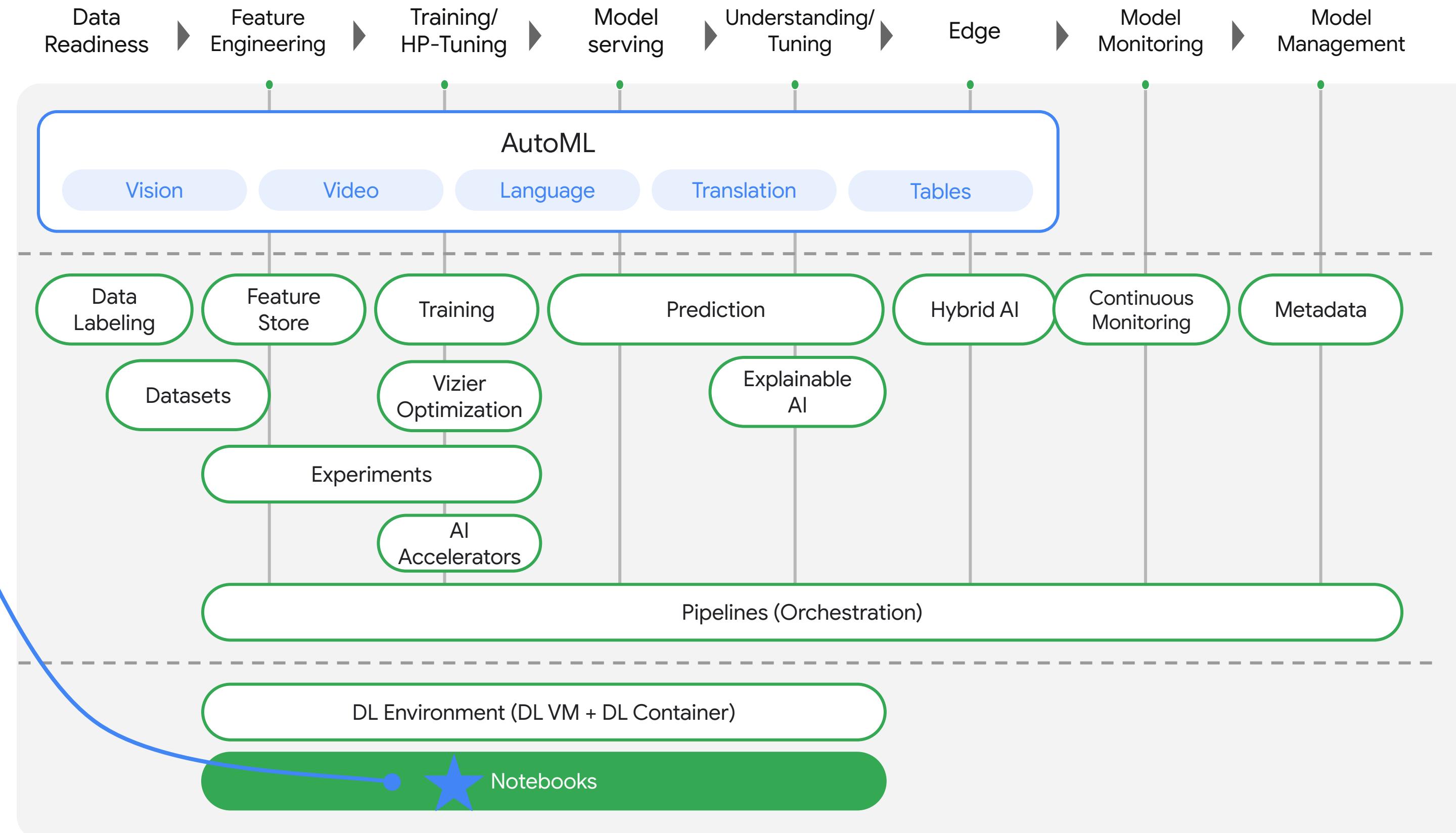


End-To-End: Pipeline Orchestration

03a

Notebook: 03a

Vertex AI Overview



Google Cloud Platform

Vertex AI

- Dashboard
- Datasets
- Features
- Labeling tasks
- Notebooks
- Pipelines
- Training
- Experiments
- Models
- Endpoints
- Batch predictions
- Metadata

Marketplace

Google Cloud Platform statmike-mlops Search products and resources

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

Explorer + ADD DATA EDITOR DIGITS DIGITS_LR

Type to search

Viewing pinned projects.

statmike-mlops digits Models (1) digits_lr digits digits_featurestore_import digits_fs_training digits_preped

DETAILS TRAINING EVALUATION SCHEMA

Loss Duration (sec)

10 Training loss: 0.011 Evaluation loss: 0.014

Duration (seconds)

File Edit View Run Kernel Git Tabs Settings Help

/ vertex-ai-mlops /

Name	Last Modified
architectures	3 hours ago
Dev	2 days ago
temp	5 hours ago
00 - Environment...	3 days ago
01 - BigQuery -...	3 days ago
02a - Vertex AI ...	4 days ago
02b - Vertex AI ...	10 hours ago
02c - Vertex AI ...	10 hours ago
03a - BigQuery Machine Learning (BQML) - Machine Learning with SQL	4 days ago

03a - BigQuery Machine Learning (BQML) - Machine Learning with SQL

BigQuery has a number of machine learning algorithms callable directly from SQL. This gives the convenience of using the common language of SQL to "CREATE MODEL ..."). The library of available models is constantly growing and covers supervised, unsupervised, and time series methods as well as functions for evaluation - even anomaly detection from results, explainability and hyperparameter tuning. A great starting point for seeing the scope of available methods is [your journey for models](#).



BigQuery Machine Learning

Vertex AI

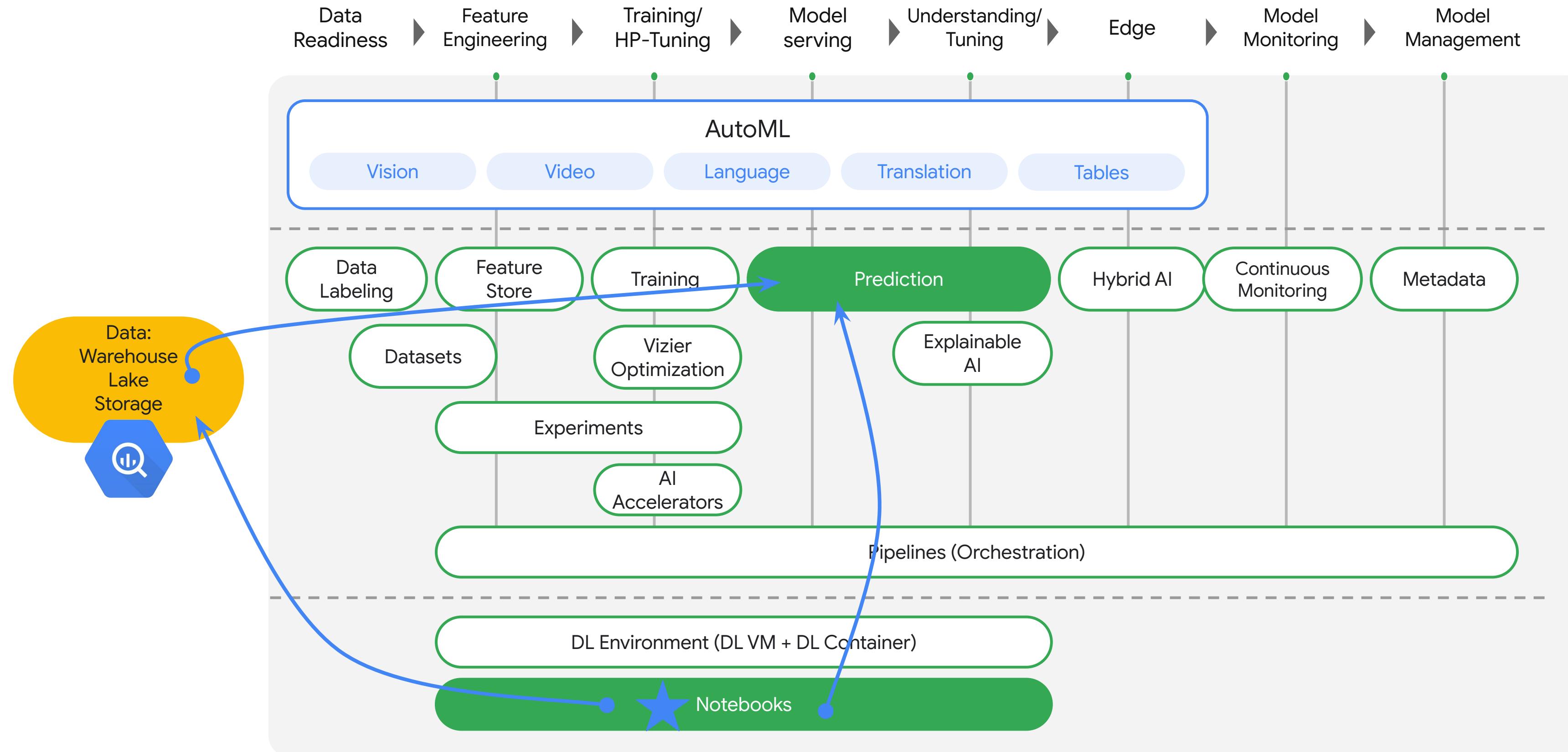


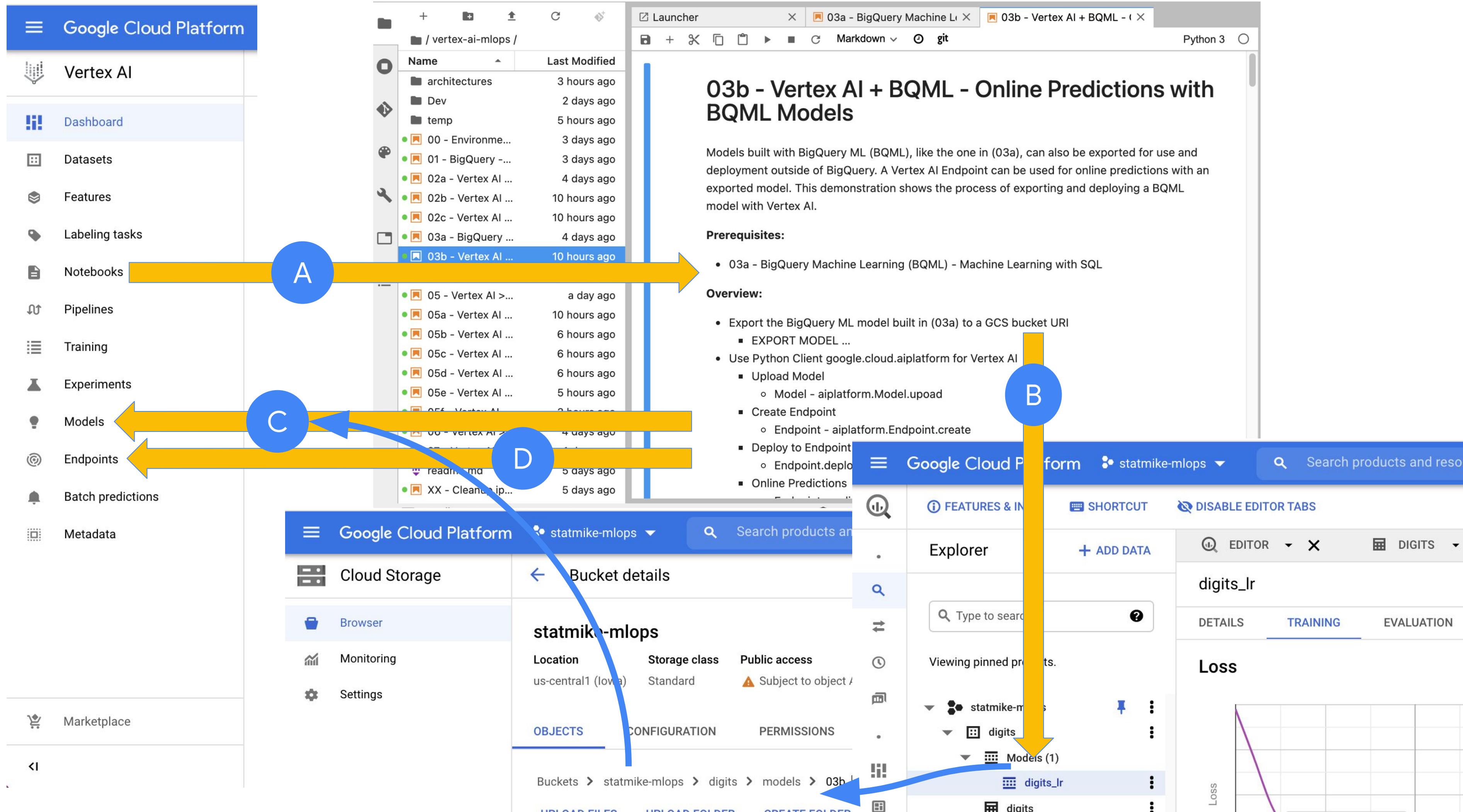
End-To-End with SQL

03b

Notebook: 03b

Vertex AI Overview

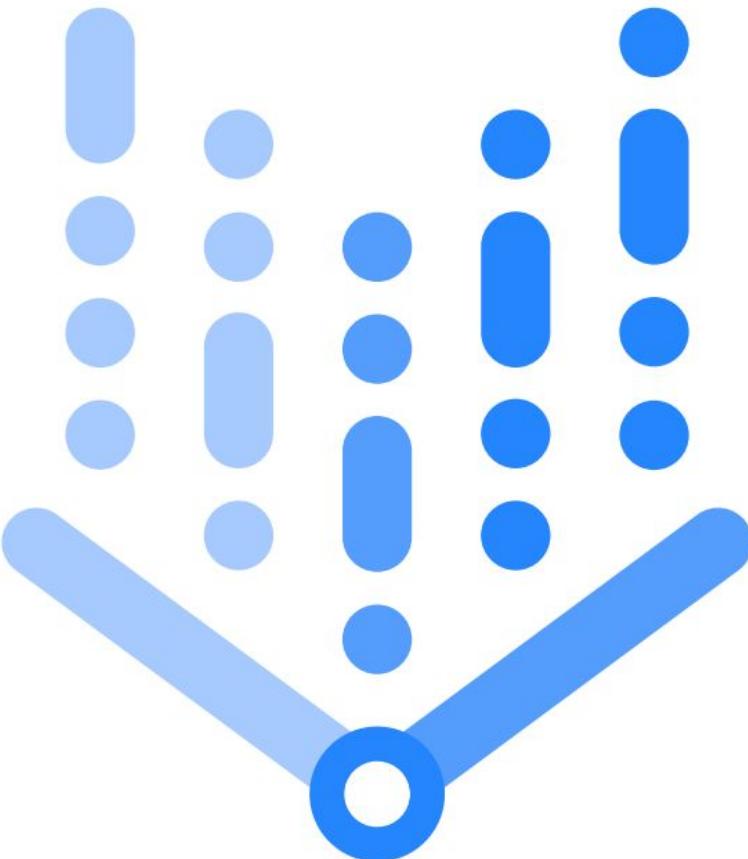






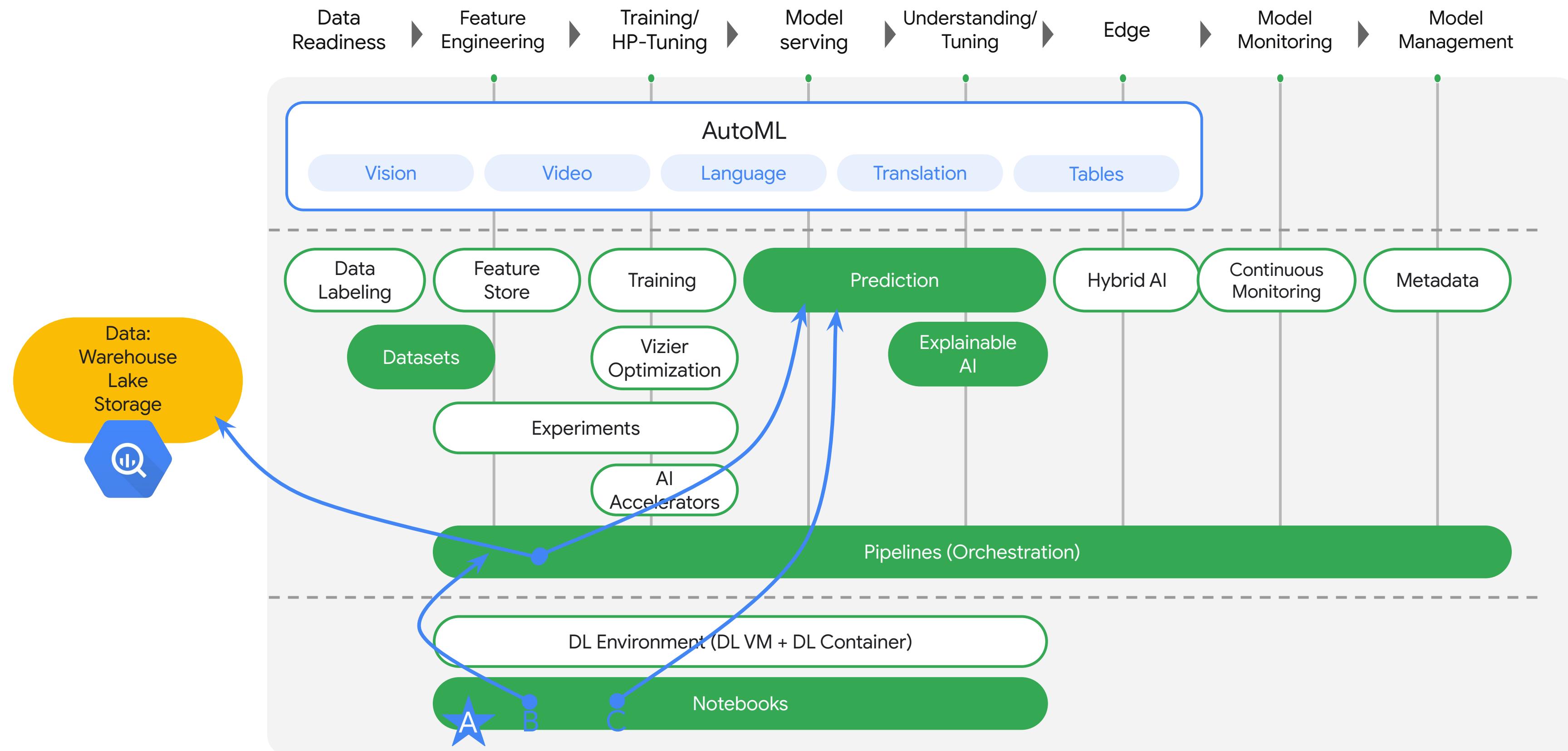
BigQuery Machine Learning

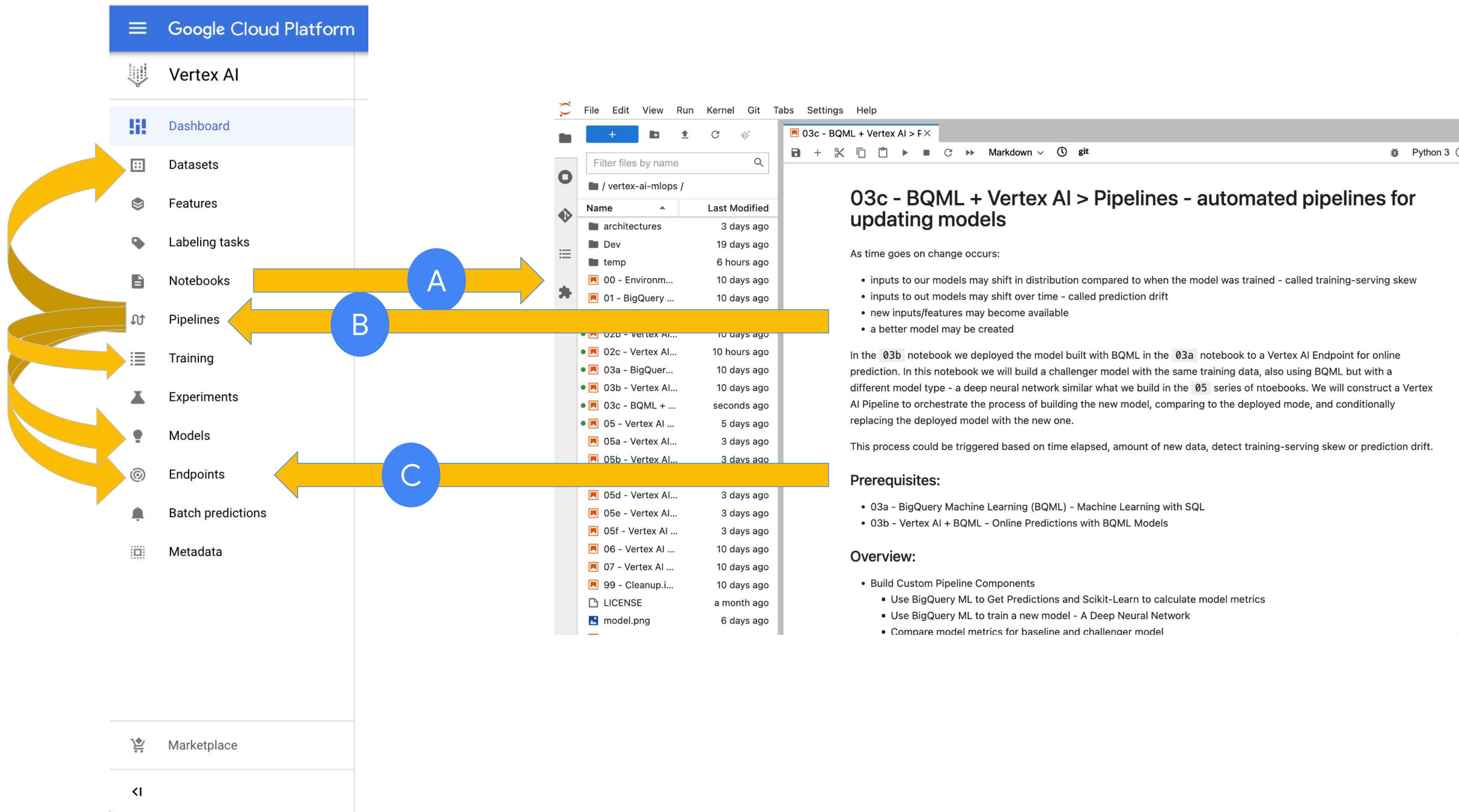
Vertex AI



BQML to Online Predictions

03c







BigQuery Machine Learning

Vertex AI

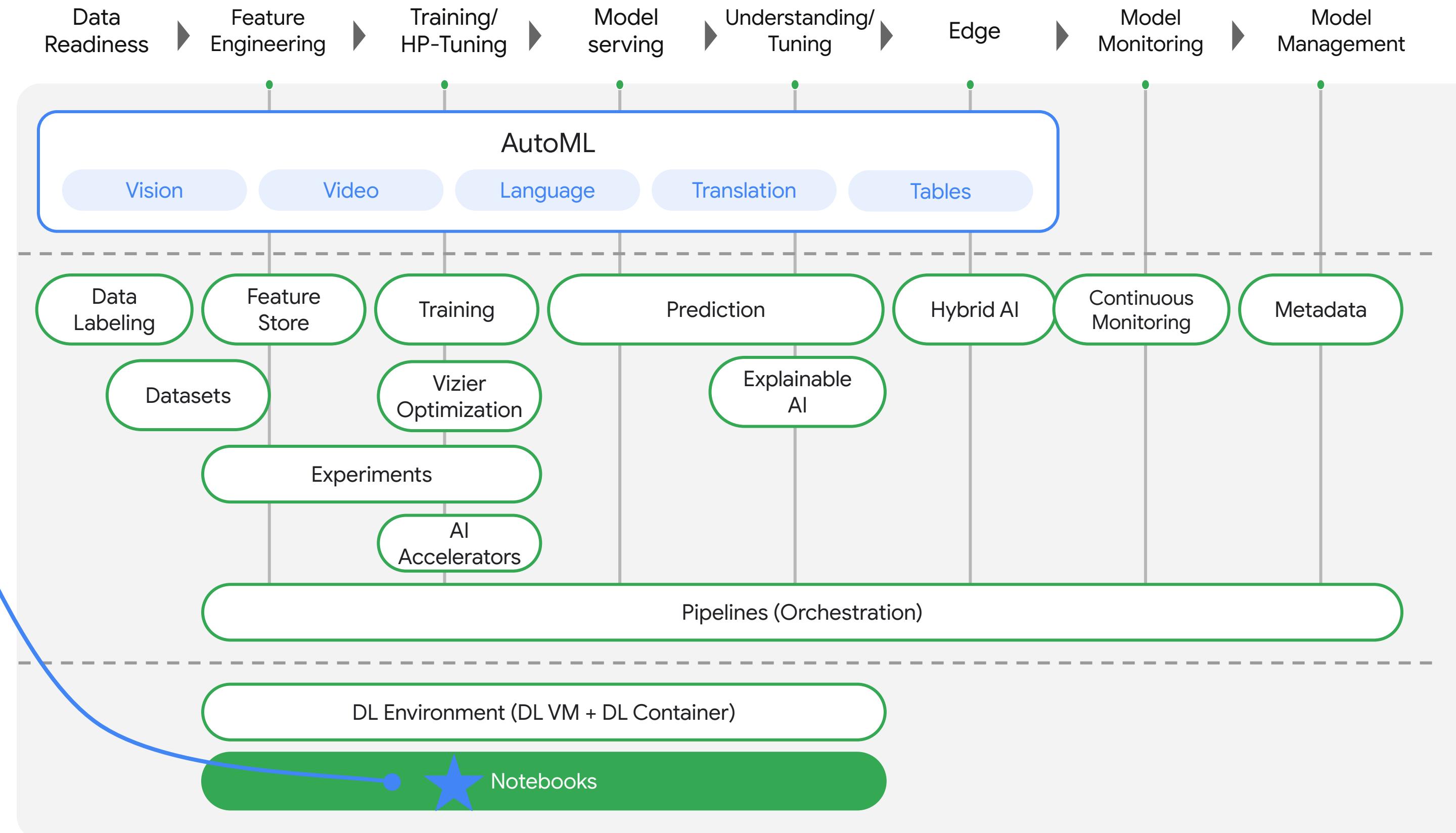


End-To-End Pipeline Orchestration
Conditionally Update Endpoints

04

Notebook: 04

Vertex AI Overview



Google Cloud Platform

Vertex AI

Dashboard

Datasets

Features

Labeling tasks

Notebooks

Pipelines

Training

Experiments

Models

Endpoints

Batch predictions

Metadata

Marketplace

File Edit View Run Kernel Git Tabs Settings Help

Launcher 04 - Time Seri 04a - BigQuer 04b - Vertex A 04c - Vertex A 04d - Vertex A 04e - Vertex A 04f - Vertex A 04g - Vertex A

Markdown git Python 3

04 - Time Series Forecasting - Data Review in BigQuery

This series will explore forecasting with Vertex AI and BigQuery. Forecasting is following a measurement over time and exploring trends, the impact of seasonality (years, months, days, etc), holidays, and special events with the hope of using these insights to forecast into the near future. Before we look into methods of forecasting, this notebook will first look at a data source with a time element.

This series will use bike Citibike rentals in New York city. The bike stations near central park will be selected and the daily number bike trips that originate from these stations will be followed over time. This will be complicated as new stations are introduced over time and some stations only have the most recent few months, or just weeks of data.

In this notebook we look at the data available and visualize the trend over time.

Prerequisites:

- 00 - Environment Setup

Overview:

- Setup
- Data Setup
- Propose Train/Validation/Test Splits by Date Range
- Create Source BigQuery Table for Forecasting
- Review Forecasting Data
- Validation Check: Extract dates for splits from source
- Validation Check: Duplicate dates at stations?
- Time Series Review

Resources:

- [How to do time series forecasting in BigQuery](#)

Google Cloud Platform statmike-demo3 Search Products, resources, docs (/)

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

Explorer + ADD DATA

CITIBIKE... X CITIBIKE... X CITIBIKE... X

citibikes_prep... QUERY ASK QUESTION SHARE COPY SNAPSHOT DELETE

Type to search

Viewing pinned projects.

statmike-demo3 citibikes Models (1) 04b_automl 04c_automl 04d_automl_best 04d_automl_cw0 04d_automl_cw1 04d_automl_cw2 04d_automl_cw4 04d_automl_cw8 04e_prophet 04f_prophet

SCHEMA DETAILS PREVIEW TABLE EXPLORER

Row	start_station_name	date	num_trips	splits
1	Central Park S & 6 Ave	2015-09-11	256	TRAIN
2	Central Park S & 6 Ave	2016-08-29	256	TRAIN
3	Central Park S & 6 Ave	2013-09-01	256	TRAIN
4	Central Park S & 6 Ave	2016-02-28	256	TRAIN
5	Central Park S & 6 Ave	2016-06-07	256	TRAIN
6	Central Park S & 6 Ave	2015-06-21	256	TRAIN
7	Central Park S & 6 Ave	2015-03-28	1	TRAIN
8	Central Park S & 6 Ave	2014-02-04	1	TRAIN
9	Central Park S & 6 Ave	2015-01-18	1	TRAIN
10	Central Park S & 6 Ave	2014-01-22	1	TRAIN
11	Central Park S & 6 Ave	2015-02-16	1	TRAIN
12	Central Park S & 6 Ave	2014-01-07	1	TRAIN



Time Series Forecasting

Vertex AI

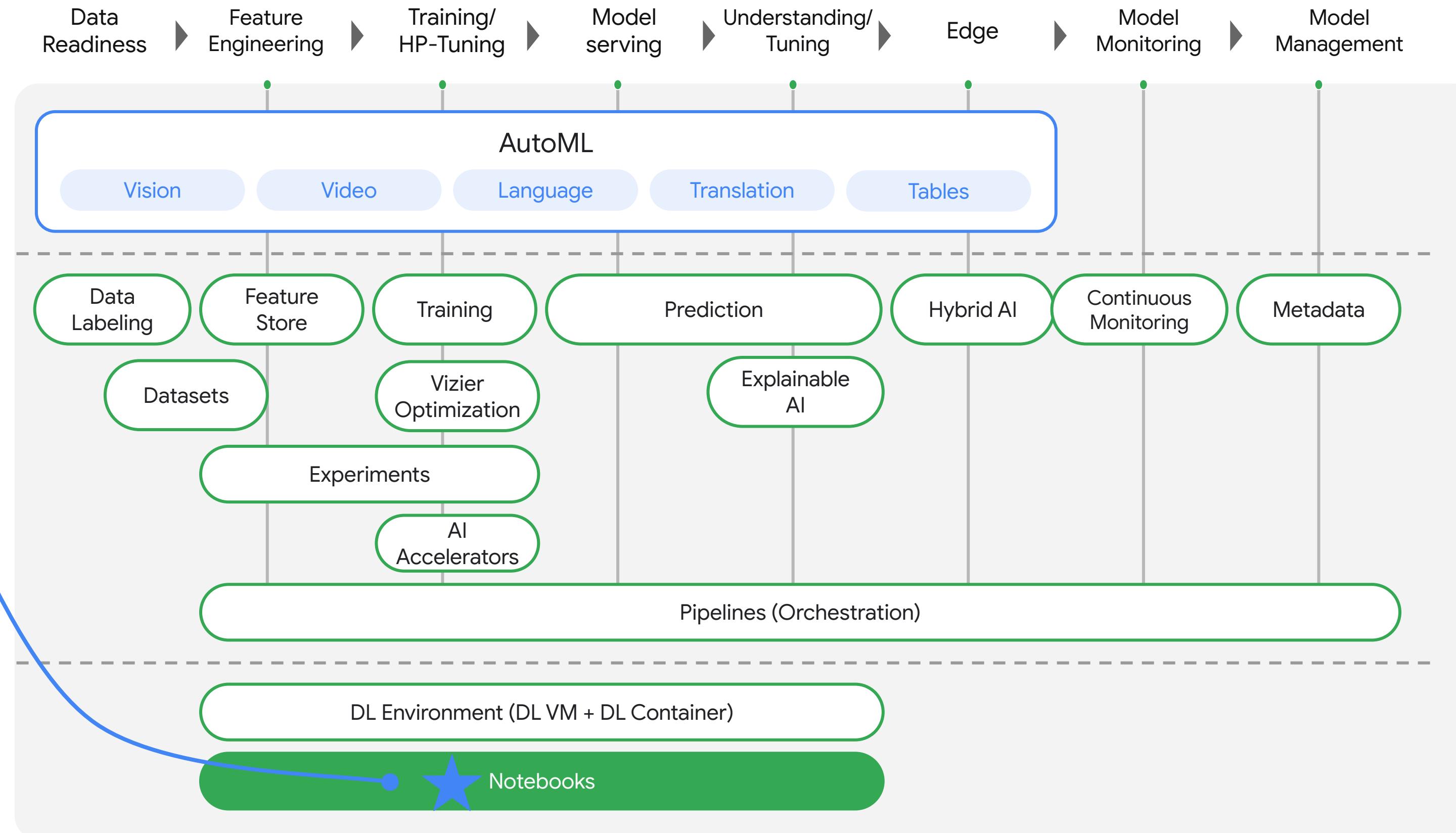


Data Review with BigQuery

04a

Notebook: 04a

Vertex AI Overview



Google Cloud Platform

Vertex AI

- Dashboard
- Datasets
- Features
- Labeling tasks
- Notebooks
- Pipelines
- Training
- Experiments
- Models
- Endpoints
- Batch predictions
- Metadata

Marketplace

File Edit View Run Kernel Git Tabs Settings Help

Launcher X 04 - Time Seri X 04a - BigQuer X 04b - Vertex A X 04c - Vertex A X 04d - Vertex A X 04e - Vertex A X 04f - Vertex A X 04g - Vertex A ● Python 3

04a - BigQuery Machine Learning (BQML) - Univariate Forecasting with ARIMA+

In this notebook we will fit a forecasting model to the data reviewed and prepared in the 04 notebook - citibike bike rentals daily volume by station near Central Park in New York City. For this attempt we will keep the forecasting activity within the data warehouse by using BigQuery's built-in machine learning - often called BQML. One of the model types available is ARIMA+ (`model_type = 'ARIMA_PLUS'`). This fits univariate forecasts by using auto-regressive integrated moving average (ARIMA) models while also automatically handling many of the data pre and post processing steps required. Each station will be forecasted separately by simply providing the station names as `start_station_name` to the input `time_series_id_col`. The forecast will be evaluated with various provided metrics and we will use SQL to calculate customized metrics. It is even possible to identify anomalies using BigQuery ML!

Prerequisites:

- 04 - Time Series Forecasting - Data Review in BigQuery

Overview:

- Setup
- Review Time Series
- Create Forecast Model
 - Review the ARIMA coefficients
 - Review the input features
 - Review the training information
- Forecast Evaluation
 - Forecast metrics
 - Evaluate time series metrics
- Forecast Time Series
- Forecast + Explanation for each station
 - Visualize Time Series
 - Forecast
- Using SQL to Calculate Metrics
- Anomaly Detection

Google Cloud Platform statmike-demo3 Search Products, resources, docs (/)

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

Explorer + ADD DATA

CITIBIKE... X CITIBIKE... X CITIBIKE... X CITIBIKE... X

citibikes_arimaplus

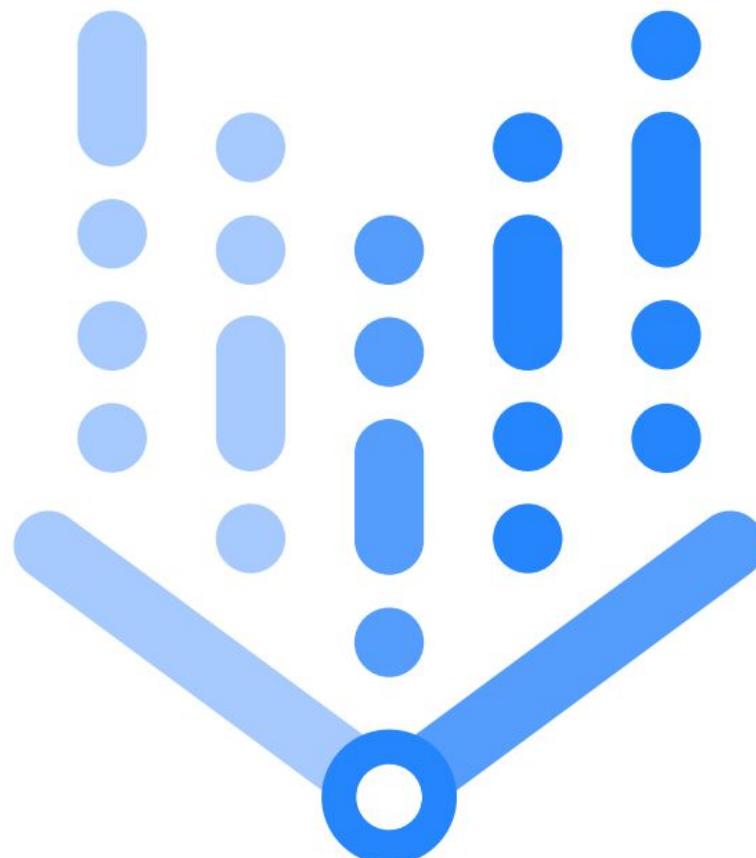
DETAILS TRAINING EVALUATION SCHEMA

Time Series ID	Non Seasonal P	Non Seasonal D	Non Seasonal Q	Has Drift	Has Spik
Central Park	0	0	1	False	False
North &					
Adam					
Clayton					
Powell Blvd					
Central Park	0	1	5	False	True
S & 6 Ave					
Central Park	1	0	0	False	False
W & W 96 St					
Central Park	0	0	0	False	False
West & W 100 St					
Central Park	0	0	4	False	False
West & W 102 St					
Central Park	0	1	5	False	True
West & W 68 St					
Central Park	0	1	5	False	True
West & W 70					



Time Series Forecasting

Vertex AI

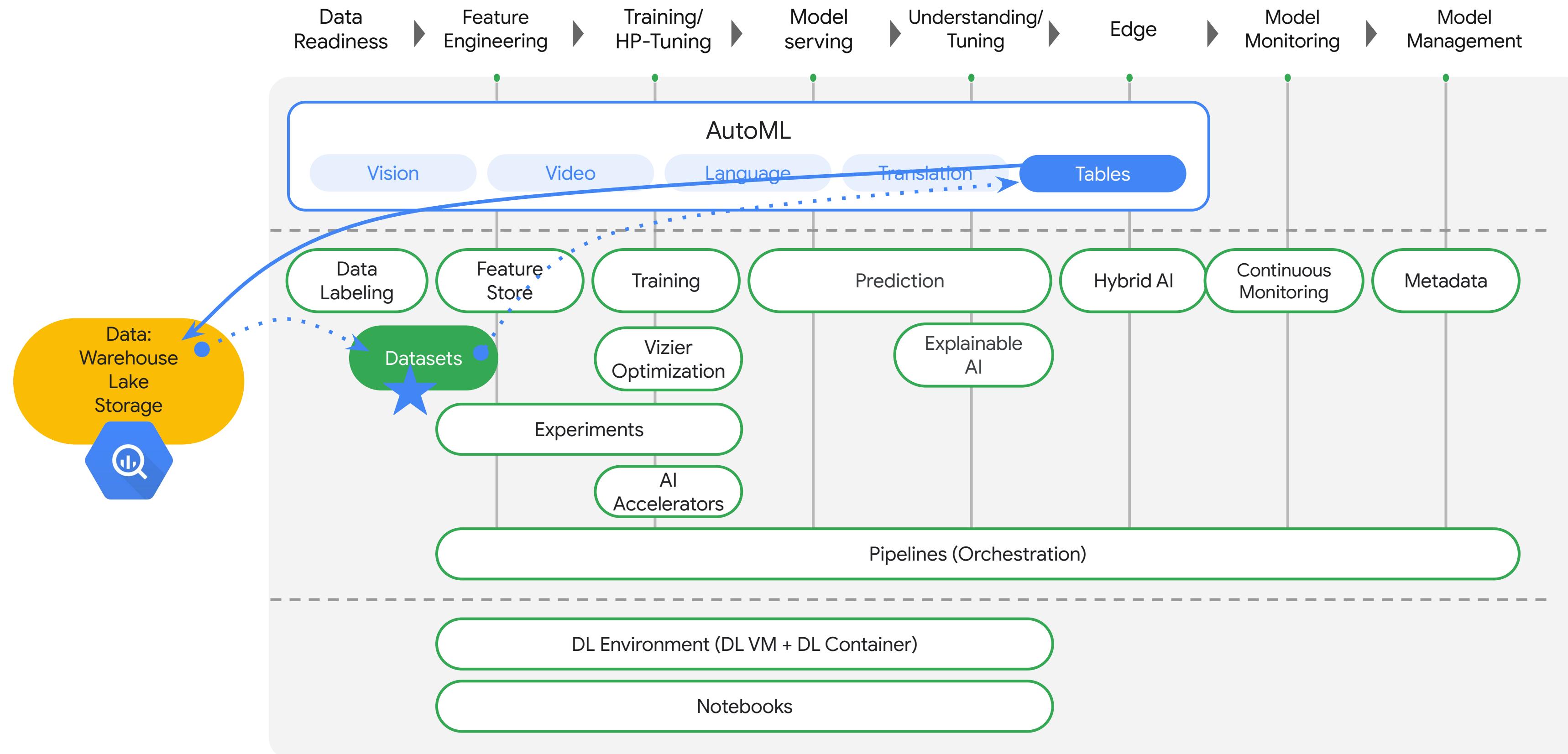


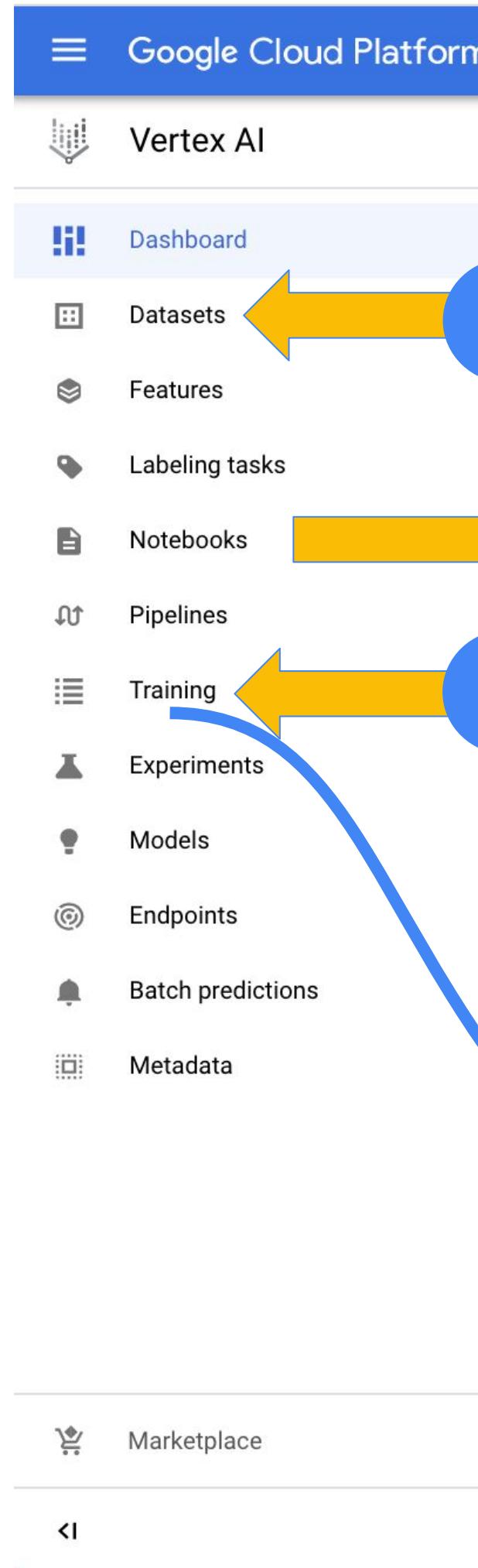
BigQuery Machine Learning
Univariate Forecasting with ARIMA+

04b

Notebook: 04b

Vertex AI Overview





04b - Vertex AI - AutoML for Global Forecasting with Deep Learning - In GCP Console (no code)

A newer way to fit forecast models to time series data is by using a deep learning approach with neural networks. This approach can have added success because it not only sees individual time series but also sees across many time series and can learn together from correlation. This approach can also use covariate features that may only be known in the past or may also be available at forecast time. Vertex AI AutoML forecasting will do automatic feature engineering, model architecture search and hyperparameter tuning specific to time series data. This method uses a set of data between the training data and the test data validation data to time series specific approaches using deep learning architectures.

In this notebook, screenshots are shared that illustrate a no-code walk-through all using the Vertex AI interface in the GCP Console. The best part, is a test data will be forecasted and written back to BigQuery where we will review it, visualize it, and calculate custom metrics to make fair comparison to other methods explored in this notebook series (04).

- 04 - Time Series Forecasting - Data Review in BigQuery

Overview:

- Create Dataset
- Train Forecasting Model with AutoML
 - Training Method
 - Model Details
 - Training Options
 - Compute and Pricing
 - Review Training
- Forecasting Model
- Test Data in BigQuery
- Setup

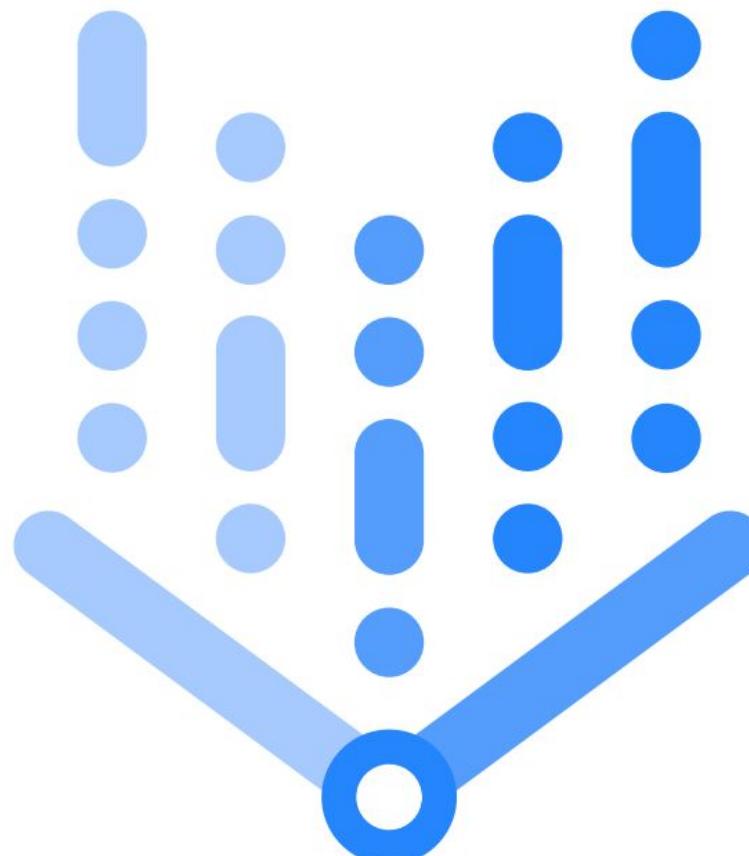
BigQuery Preview Data:

Row	date	num_trips	predicted_on_date	splits	start_station_name	predicted_num_trips
1	2016-09-15	392	2016-09-15	TEST	Central Park S & 6 Ave	267.69180297851562
2	2016-09-15	75	2016-09-15	TEST	Central Park W & W 96 St	...
3	2016-09-15	170	2016-09-15	TEST	Central Park West & W 68 St	...
4	2016-09-15	180	2016-09-15	TEST	Central Park West & W 72 St	...
5	2016-09-15	101	2016-09-15	TEST	Central Park West & W 76 St	...
6	2016-09-15	102	2016-09-15	TEST	Central Park West & W 85 St	...
7	2016-09-15	61	2016-09-15	TEST	W 82 St & Central Park West	...
8	2016-09-15	53	2016-09-15	TEST	Central Park West & W 100 St	...



Time Series Forecasting

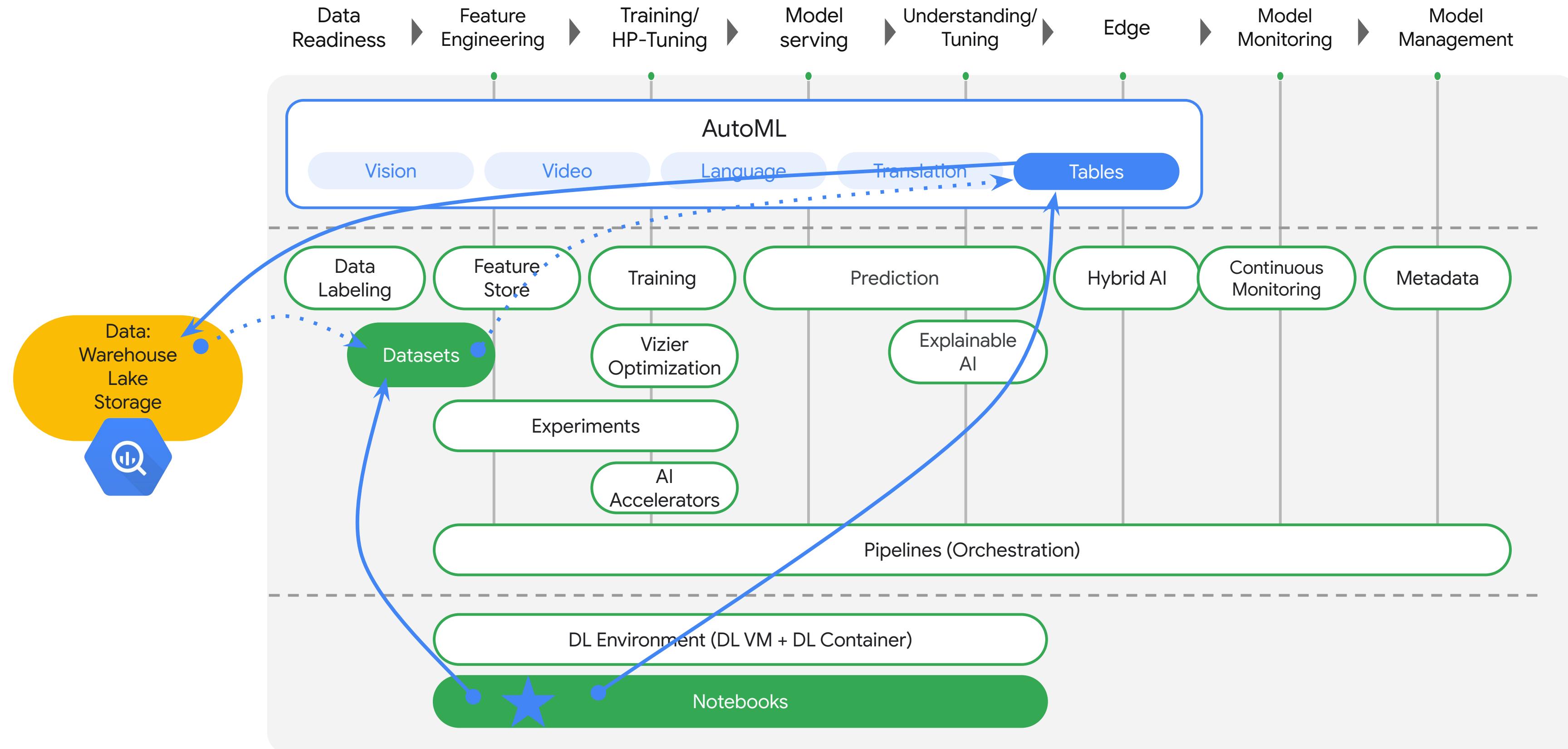
Vertex AI

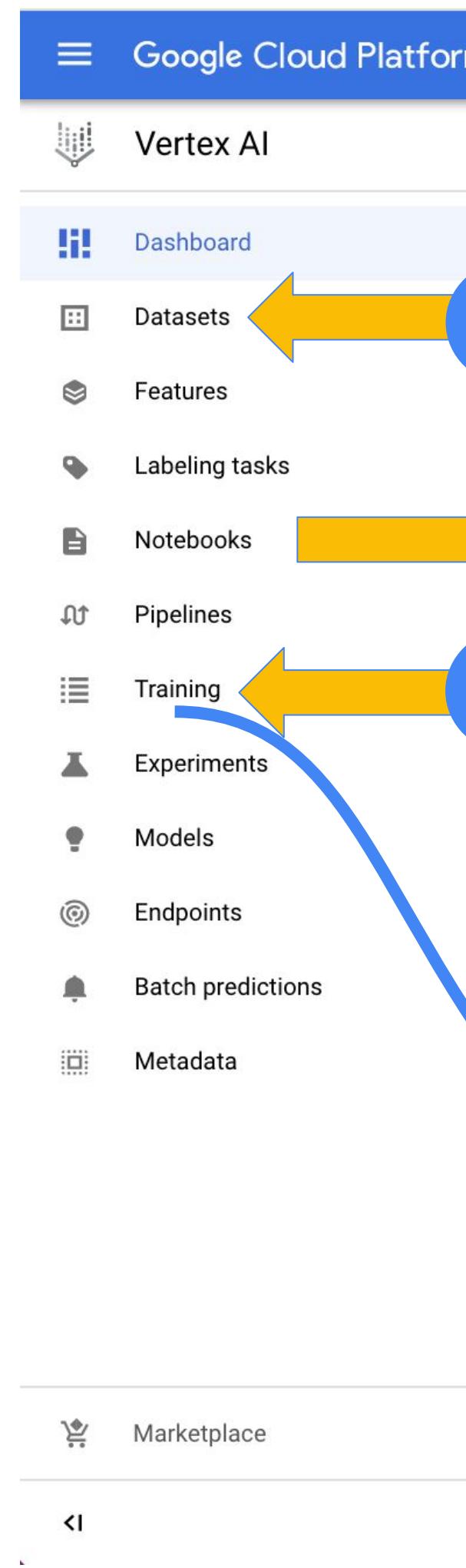


AutoML With The Console (No Code)

Global Forecasting With Deep Learning

04c





04c - Vertex AI - AutoML for Global Forecasting with Deep Learning - with Python client (code)

This notebook series explores forecasting time series data using deep learning. Unlike traditional time series forecasting, this approach can handle multiple time series simultaneously, learning correlations between them. It includes automatic feature engineering, model architecture search, and hyperparameter tuning specific to time series data. The process involves training a model on historical data and testing it on future data to predict outcomes.

In this notebook, the Vertex AI Python Client is used to automate the process of using Vertex AI AutoML Forecasting with interactive code. The best part is that all the test data will be forecasted and written back to BigQuery where we will review it, visualize it, and calculate custom metrics to make fair comparison to other methods explored in this notebook series (04).

Prerequisites:

- Setup
- Create Dataset
- Train Forecasting Model with AutoML
 - Create AutoML Forecasting Job
 - Run AutoML Forecasting Job
- Results
 - Retrieve Test Data
 - Review Custom Metrics with SQL
 - Retrieve the Forecasting
 - Visualize the Time Series

Overview:

The screenshot shows a Jupyter Notebook environment with tabs for 'Launcher', '04 - Time Seri...', '04a - BigQuer...', '04b - Vertex A...', '04c - Vertex A...', '04d - Vertex A...', '04e - Vertex A...', '04f - Vertex A...', and '04g - Vertex A...'. The current tab is '04c - Vertex AI - AutoML for Global Forecasting with Deep Learning - with Python client (code)'. The notebook content includes the text above, a 'Prerequisites:' section, and an 'Overview:' section. At the bottom, the BigQuery interface is visible, showing a table named '04c_automl' with columns 'date', 'num_trips', 'predicted_on_date', 'splits', 'start_station_name', and 'predicted_num_trips'. The table contains 8 rows of data for September 15, 2016, from Central Park S & 6 Ave to W 82 St & Central Park West.

Row	date	num_trips	predicted_on_date	splits	start_station_name	predicted_num_trips
1	2016-09-15	392	2016-09-15	TEST	Central Park S & 6 Ave	299.864013671875
2	2016-09-15	75	2016-09-15	TEST	Central Park W & W 96 St	...
3	2016-09-15	170	2016-09-15	TEST	Central Park West & W 68 St	...
4	2016-09-15	180	2016-09-15	TEST	Central Park West & W 72 St	...
5	2016-09-15	101	2016-09-15	TEST	Central Park West & W 76 St	...
6	2016-09-15	102	2016-09-15	TEST	Central Park West & W 85 St	...
7	2016-09-15	61	2016-09-15	TEST	W 82 St & Central Park West	...
8	2016-09-15	53	2016-09-15	TEST	Central Park West & W 100 St	...



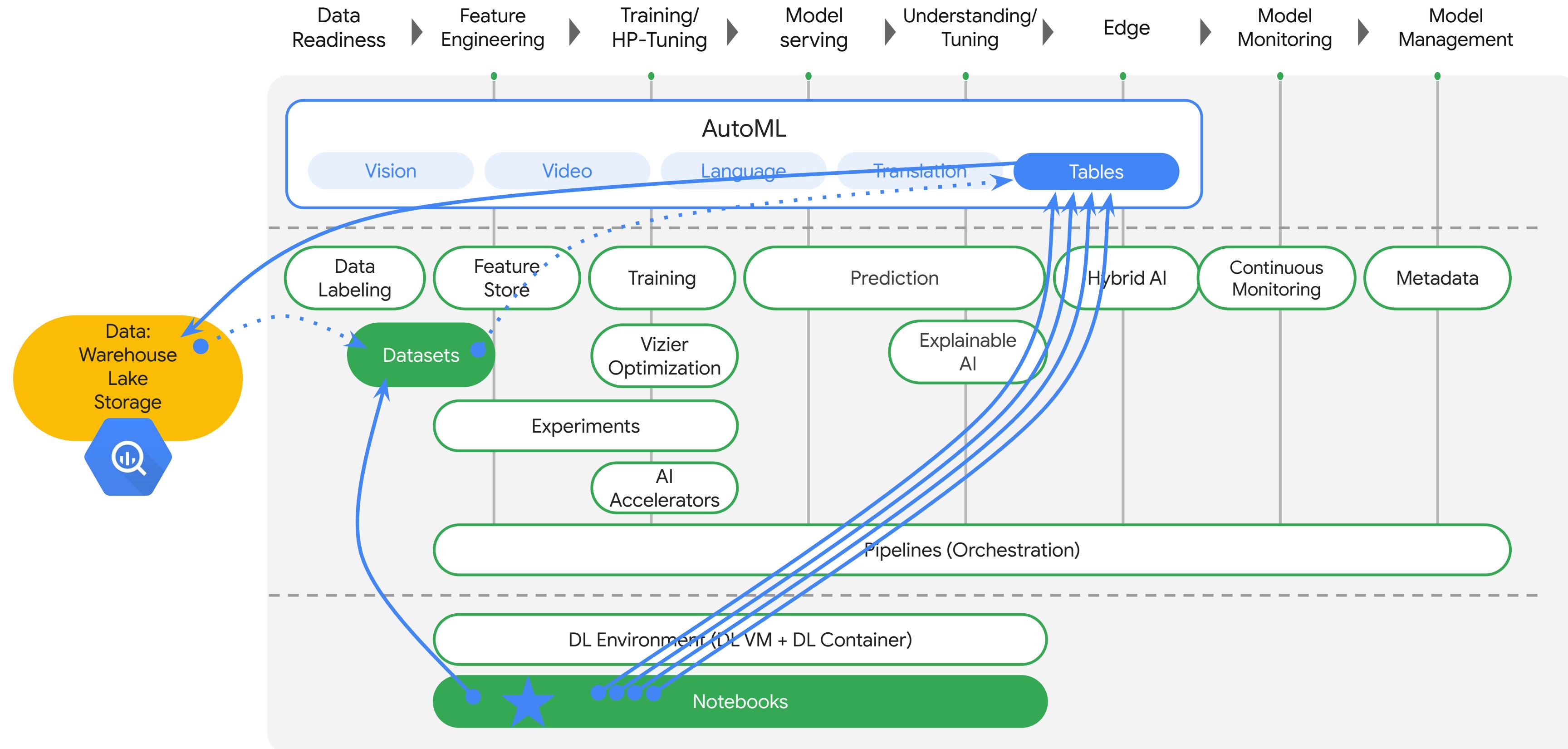
Time Series Forecasting

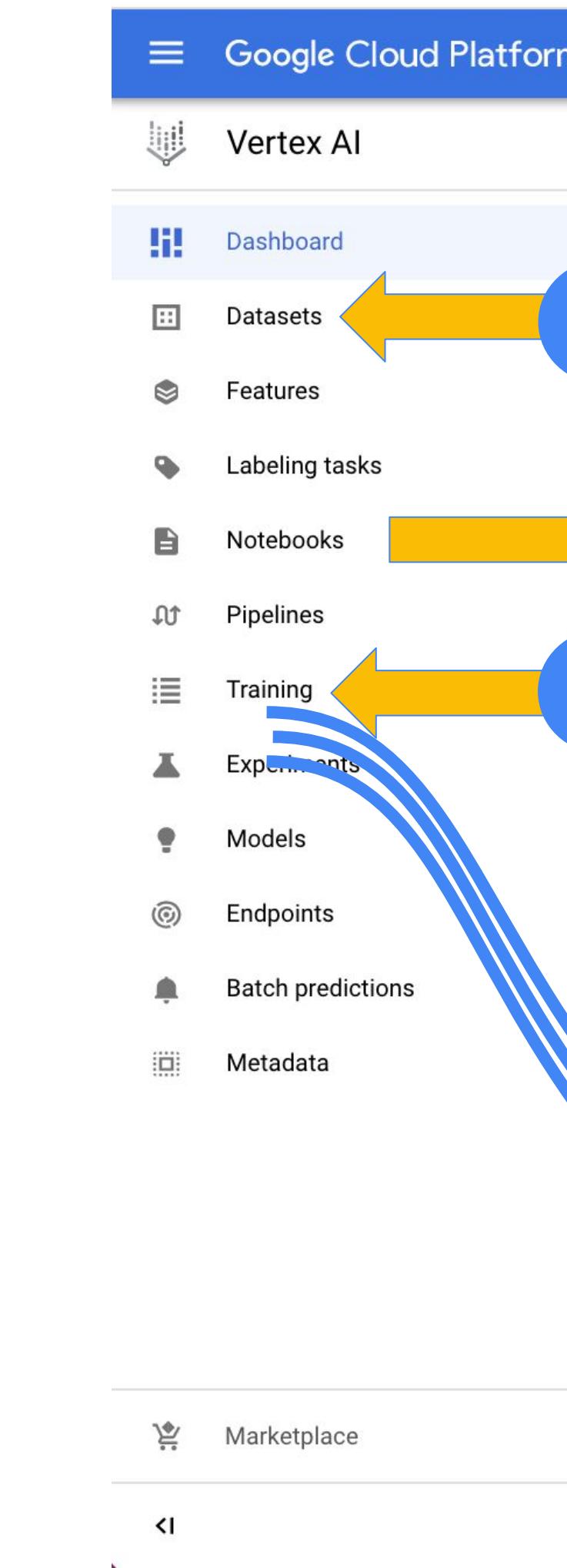
Vertex AI



AutoML With The Python Client
Global Forecasting With Deep Learning

04d





04d - Vertex AI - AutoML for Global Forecasting with Deep Learning - multiple parallel scenarios

A power way to fit forecast models to time series data is by using a deep learning approach with neural networks. This approach can have added success because it is a covariate features that may only be known in the past or may also be available at forecast time. Vertex AI AutoML forecasting will do automatic feature engineering, model architecture search and hyperparameter tuning specific to time series data. This method using a set of data between the training data and the test data called validation data to time series specific approaches using deep learning architectures.

In the `04c` notebook we used the Vertex AI Python Client to automate the process of using Vertex AI AutoML forecasting. One of the parameters for a forecast scenario is the context window - how far back the model looks in training and forecasting for predictive patterns. There are some tips for choosing an initial context window and iterating to an optimal value provided [here](#). To speed up this iteration we will conduct multiple scenarios, each with a different context window, all asynchronously - in parallel.

After all the scenarios complete we will use BigQuery SQL queries to combine the results of `04b`, `04c` and this notebooks scenarios and calculate custom metrics to compare and pick the best context window overall, and by series.

Prerequisites:

- 04 - Time Series Forecasting - Data Review in BigQuery
- 04b - Vertex AI - AutoML for Global Forecasting with Deep Learning - In GCP Console (no code)
- 04c - 04c - Vertex AI - AutoML for Global Forecasting with Deep Learning - with Python client (code)

Overview:

- Setup
- Create Dataset
- Train Forecasting Model with AutoML
 - Function to Create AutoML Model
- Review Scenarios

Google Cloud Platform - BigQuery Preview

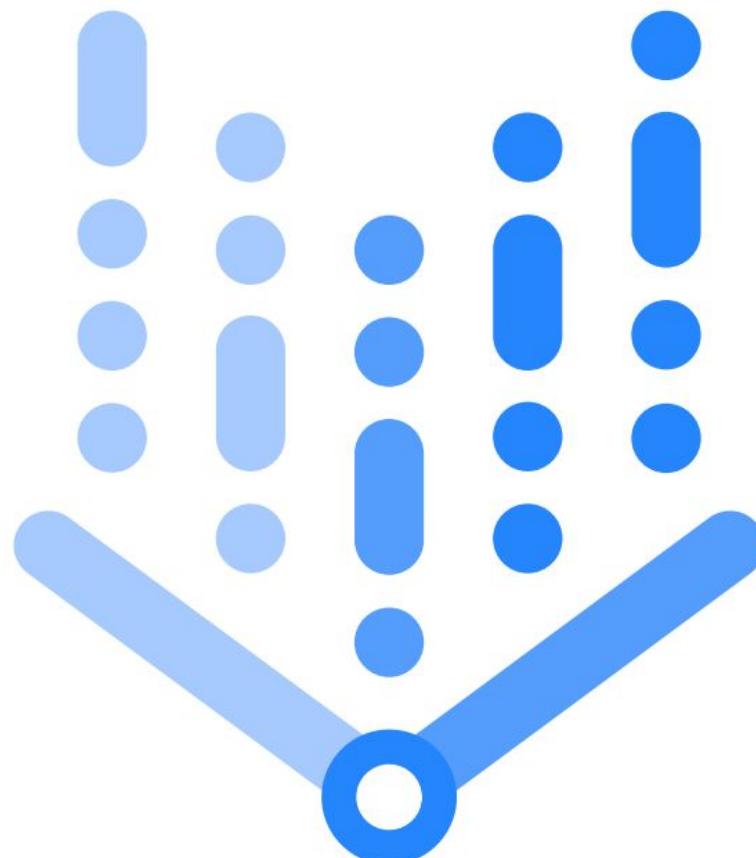
The screenshot shows the Google Cloud Platform BigQuery interface with the following schema and data preview:

Row	date	num_trips	predicted_on_date	splits	start_station_name	predicted_num_trips	CW
1	2016-09-15	132	2016-09-15	TEST	Central Park North & Adam Clayton Powell Blvd	1	
2	2016-09-16	175	2016-09-16	TEST	Central Park North & Adam Clayton Powell Blvd	1	
3	2016-09-16	175	2016-09-15	TEST	Central Park North & Adam Clayton Powell Blvd	1	
4	2016-09-17	269	2016-09-16	TEST	Central Park North & Adam Clayton Powell Blvd	1	
5	2016-09-17	269	2016-09-17	TEST	Central Park North & Adam Clayton Powell Blvd	1	
6	2016-09-17	269	2016-09-15	TEST	Central Park North & Adam Clayton Powell Blvd	1	
7	2016-09-18	272	2016-09-16	TEST	Central Park North & Adam Clayton Powell Blvd	1	
8	2016-09-18	272	2016-09-17	TEST	Central Park North & Adam Clayton Powell Blvd	1	
9	2016-09-18	272	2016-09-15	TEST	Central Park North & Adam Clayton Powell Blvd	1	
10	2016-09-18	272	2016-09-18	TEST	Central Park North & Adam Clayton Powell Blvd	1	
11	2016-09-19	39	2016-09-19	TEST	Central Park North & Adam Clayton Powell Blvd	1	
12	2016-09-19	39	2016-09-16	TEST	Central Park North & Adam Clayton Powell Blvd	1	



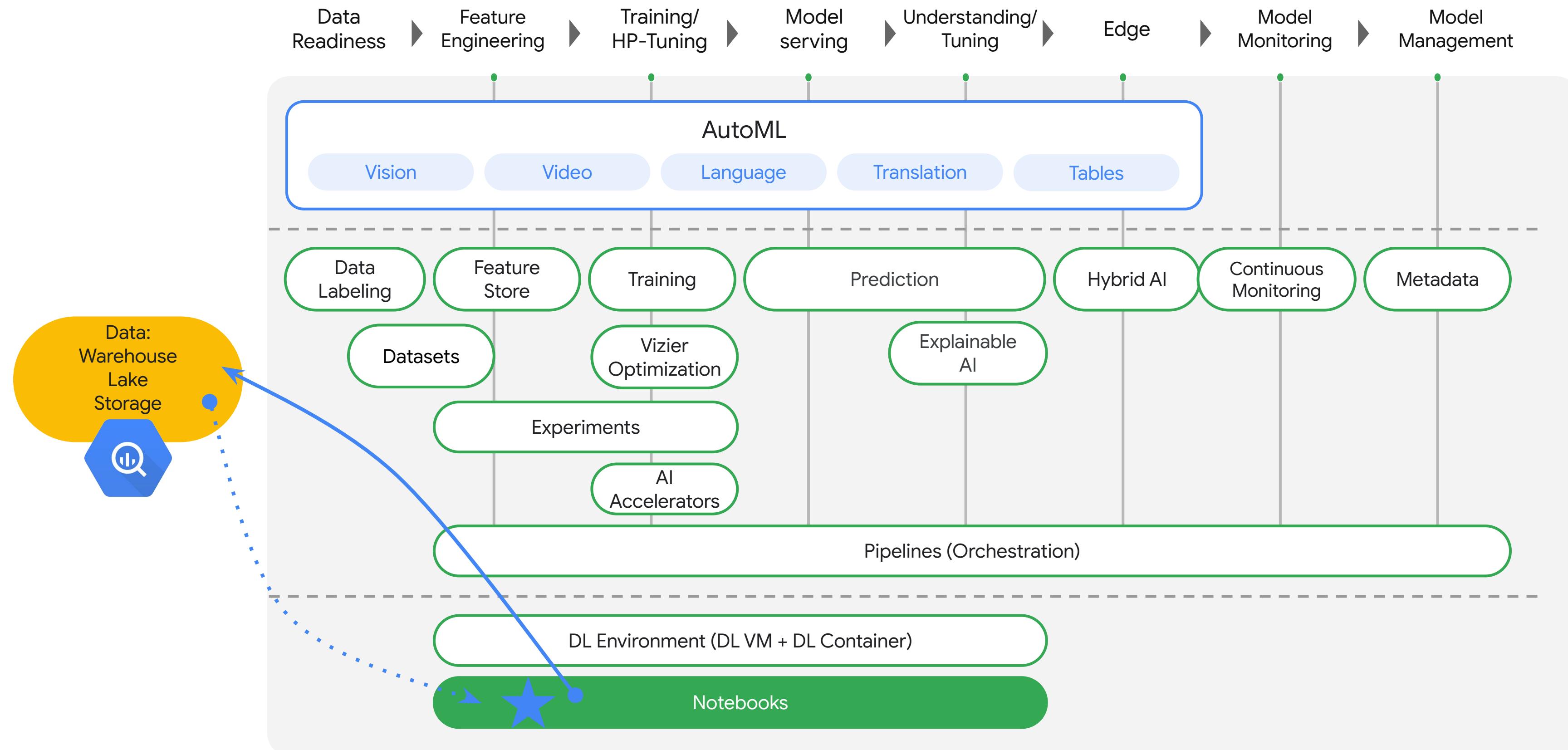
Time Series Forecasting

Vertex AI



AutoML With Multiple Scenarios
Global Forecasting With Deep Learning

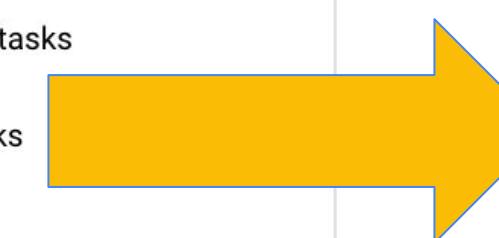
04e



Google Cloud Platform

Vertex AI

- Dashboard
- Datasets
- Features
- Labeling tasks
- Notebooks
- Pipelines
- Training
- Experiments
- Models
- Endpoints
- Batch predictions
- Metadata
- Marketplace



File Edit View Run Kernel Git Tabs Settings Help

Launcher 04 - Time Seri X 04a - BigQuer X 04b - Vertex A X 04c - Vertex A X 04d - Vertex A X 04e - Vertex A X 04f - Vertex A X 04g - Vertex A X

+ X Markdown git Python 3

04e - Vertex AI > Notebooks - Forecasting with Prophet (Python)

In the prior notebooks of this 04 series we have git forecast models using Vertex AI AutoML and BigQuery ML ARIMA+. In this notebook and open-source solution named [Prophet](#) is utilized for custom forecasting. Prophet is built on [Stan](#), a probabilistic programming language, and has been implemented in Python and R. Prophet is built to be robust to outliers and missing data while also being customizable where you can add domain knowledge about the forecasting. It also offers automated visualizations, including interactive plots using [Plotly](#).

The Prophet based forecast and fit within this notebook and the results are saved back to BigQuery using [Pandas gbq](#) for further processing in BigQuery. In the next notebook, 04f , the code from this notebook gets added to a custom container and used in a Vertex AI Training job as a managed service.

Prerequisites:

- 04 - Time Series Forecasting - Data Review in BigQuery

Overview:

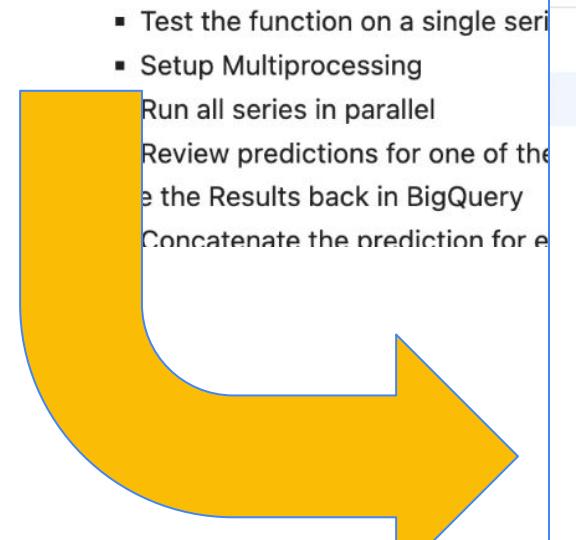
- Setup
- Time Series Data
 - Forecast Data From BigQuery to Dataframe
 - Prepate Data for Prophet
 - Review a Series Dataframe
- Create Forcast Model - Single Series
- Create Forecast Models - All Series .
 - Create a function to fit a series
 - Test the function on a single seri
 - Setup Multiprocessing

Run all series in parallel

Review predictions for one of the

Save the Results back in BigQuery

Concatenate the prediction for e



Google Cloud Platform statmike-demo3

Search Products, resources, docs (/)

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

Explorer + ADD DATA

04e_prophet QUERY ASK QUESTION SHARE COPY SNAPSHOT DELETE EXPORT

Type to search

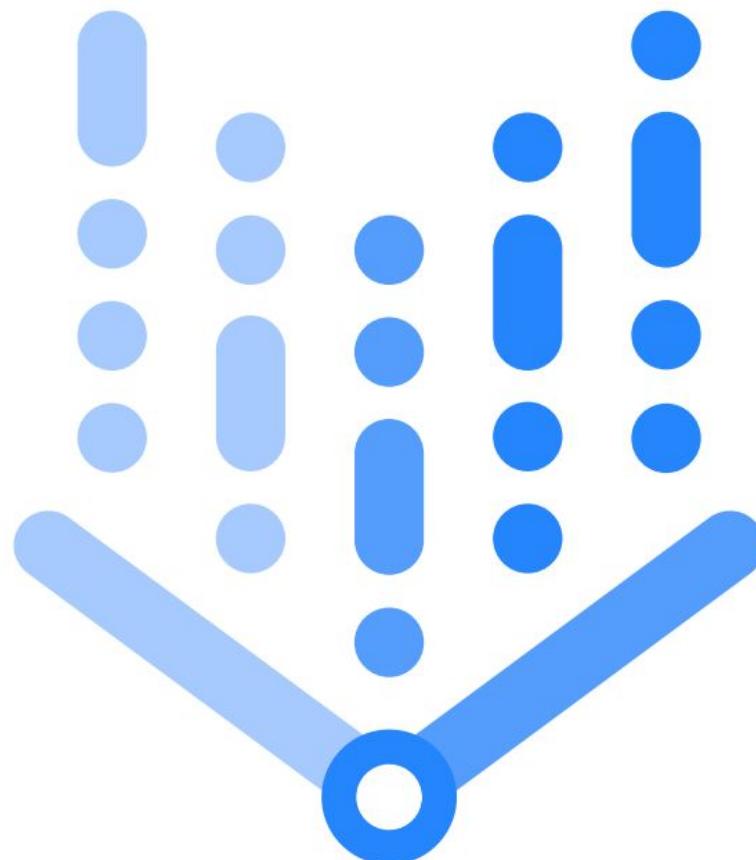
SCHEMA DETAILS PREVIEW TABLE EXPLORER

Row	start_station_name	date	num_trips	splits	yhat	yhat_lower	yhat_upper
1	Central Park North & Adam Clayton Powell Blvd	2016-09-01	14	VALIDATE	77.98183781260083	32.096824081085792	113.5208454459417
2	Central Park North & Adam Clayton Powell Blvd	2016-09-02	151	VALIDATE	139.98895202632892	97.120840528962617	175.03699166351916
3	Central Park North & Adam Clayton Powell Blvd	2016-09-03	255	VALIDATE	220.48902064285218	175.03699166351916	227.6776625348224
4	Central Park North & Adam Clayton Powell Blvd	2016-09-04	328	VALIDATE	274.49165769353232	227.6776625348224	320.86231465475782
5	Central Park North & Adam Clayton Powell Blvd	2016-09-05	259	VALIDATE	258.86231465475782	208.71078235045758	308.123508854459417
6	Central Park North & Adam Clayton Powell Blvd	2016-09-06	96	VALIDATE	123.52084854459417	80.677385615471422	164.86231465475782
7	Central Park North & Adam Clayton Powell Blvd	2016-09-07	117	VALIDATE	124.04159852540377	75.277203092482324	173.15390309116842
8	Central Park North & Adam Clayton Powell Blvd	2016-09-08	142	VALIDATE	77.978107482916442	34.660161587093668	112.04159852540377
9	Central Park North & Adam Clayton Powell Blvd	2016-09-09	129	VALIDATE	139.9852214724759	94.640588302342479	184.04159852540377
10	Central Park North & Adam Clayton Powell Blvd	2016-09-10	186	VALIDATE	220.48528990312093	173.15390309116842	267.04159852540377
11	Central Park North & Adam Clayton Powell Blvd	2016-09-11	221	VALIDATE	274.48792676519054	230.7252946035801	318.04159852540377
12	Central Park North & Adam Clayton Powell Blvd	2016-09-12	148	VALIDATE	148.151054528485	102.34062077162032	194.04159852540377
13	Central Park North & Adam Clayton Powell Blvd	2016-09-13	151	VALIDATE	123.51711746686846	78.799664045550358	171.04159852540377
14	Central Park North & Adam Clayton Powell Blvd	2016-09-14	131	VALIDATE	124.037867435348	75.892099391694217	175.03699166351916
15	Central Park North & Adam Clayton Powell Blvd	2016-09-15	132	TEST	77.9743764116525	33.52302216061836	113.5208454459417



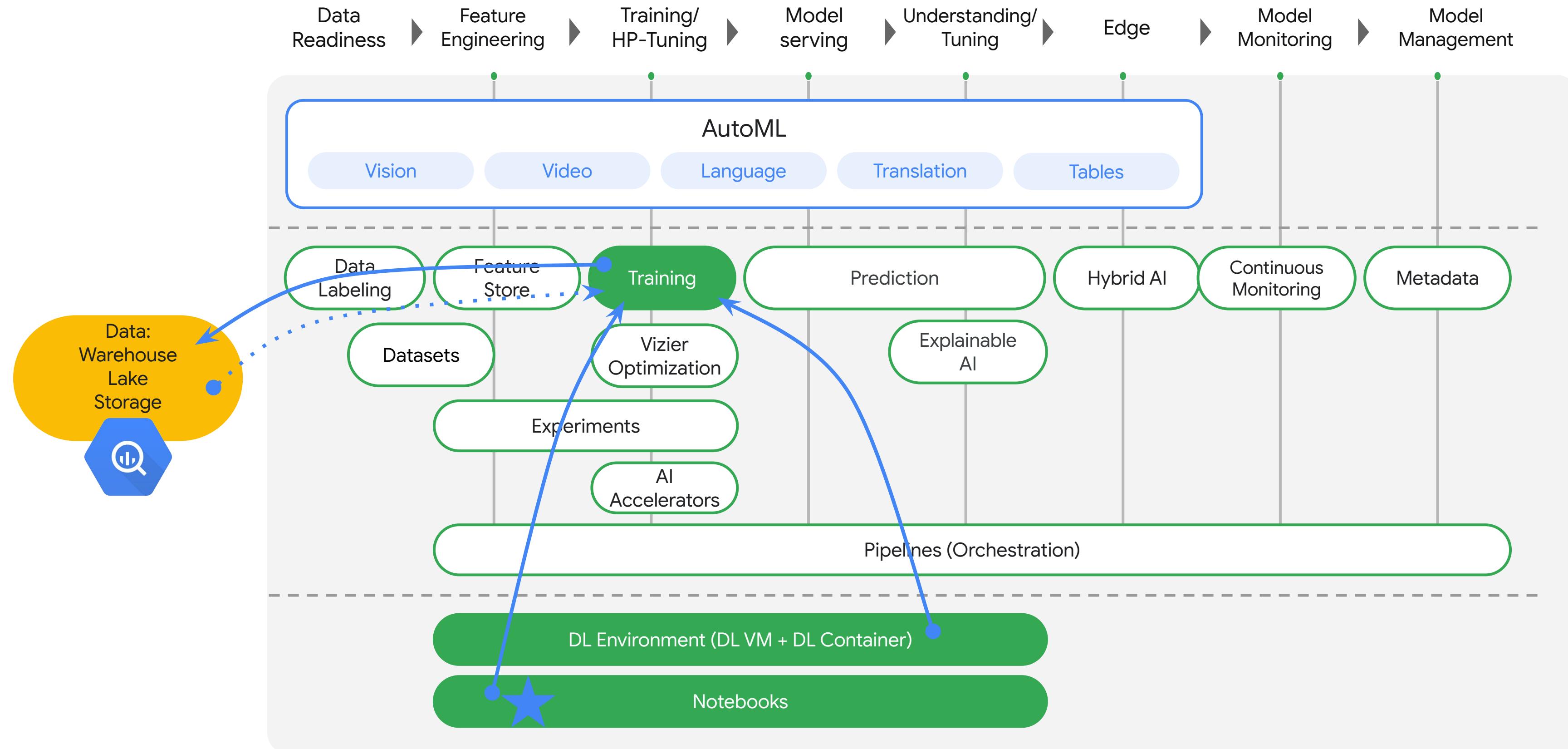
Time Series Forecasting

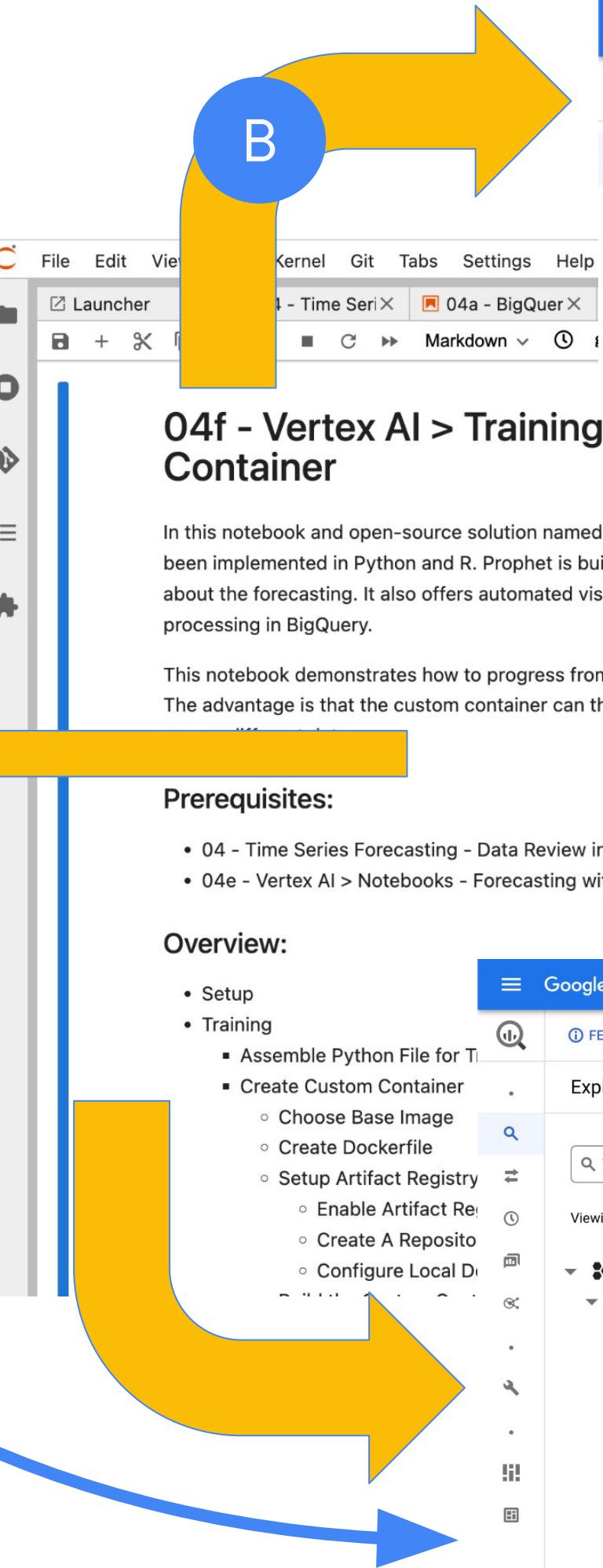
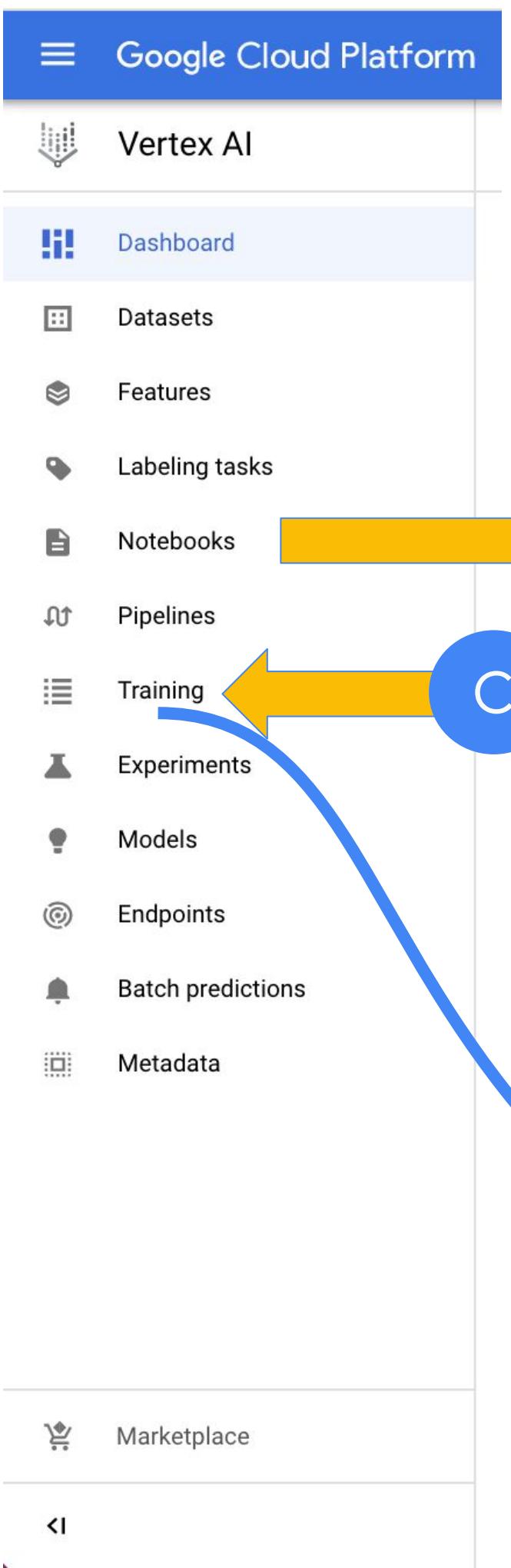
Vertex AI



**Custom Forecast In Notebook
Using Prophet in Vertex AI Notebooks**

04f





Google Cloud Platform statmike-demo3

Digests for 04f_citibikes

us-central1-docker.pkg.dev statmike-demo3 statmike-demo3 04f_citibikes

Filter Enter property name or value

Name	Description	Tags	Created	Updated
750832db4854	latest		13 hours ago	13 hours ago

Google Cloud Platform statmike-demo3

04e_prophet

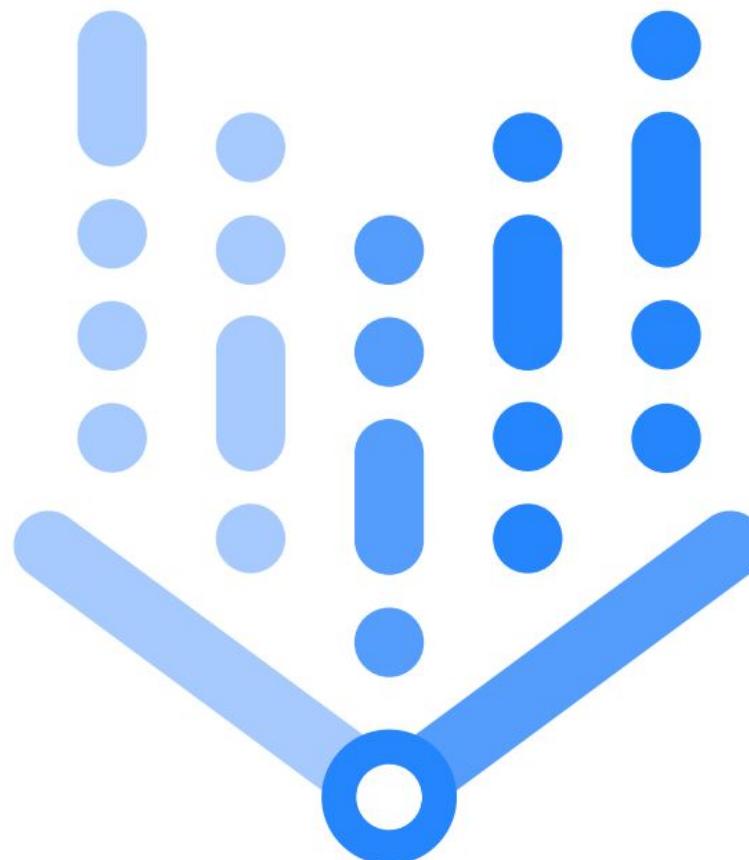
SCHEMA DETAILS PREVIEW TABLE EXPLORER

Row	start_station_name	date	num_trips	splits	yhat	yhat_lower	yhat_upper
1	Central Park North & Adam Clayton Powell Blvd	2016-09-01	14	VALIDATE	77.98183781260083	32.096824081085792	123.52084854459417
2	Central Park North & Adam Clayton Powell Blvd	2016-09-02	151	VALIDATE	139.98895202632892	97.120840528962617	227.67676625348224
3	Central Park North & Adam Clayton Powell Blvd	2016-09-03	255	VALIDATE	220.48902064285218	175.03699166351916	274.49165769353232
4	Central Park North & Adam Clayton Powell Blvd	2016-09-04	328	VALIDATE	274.49165769353232	227.67676625348224	328.52084854459417
5	Central Park North & Adam Clayton Powell Blvd	2016-09-05	259	VALIDATE	258.86231465475782	208.71078235045758	308.52084854459417
6	Central Park North & Adam Clayton Powell Blvd	2016-09-06	96	VALIDATE	123.52084854459417	80.677385615471422	164.52084854459417
7	Central Park North & Adam Clayton Powell Blvd	2016-09-07	117	VALIDATE	124.04159852540377	75.277203092482324	179.52084854459417
8	Central Park North & Adam Clayton Powell Blvd	2016-09-08	142	VALIDATE	77.978107482916442	34.660161587093668	122.52084854459417
9	Central Park North & Adam Clayton Powell Blvd	2016-09-09	129	VALIDATE	139.9852214724759	94.640588302342479	184.52084854459417
10	Central Park North & Adam Clayton Powell Blvd	2016-09-10	186	VALIDATE	220.48528990312093	173.15390309116842	267.52084854459417
11	Central Park North & Adam Clayton Powell Blvd	2016-09-11	221	VALIDATE	274.48792676519054	230.7252946035801	318.52084854459417
12	Central Park North & Adam Clayton Powell Blvd	2016-09-12	148	VALIDATE	148.151054528485	102.34062077162032	194.52084854459417
13	Central Park North & Adam Clayton Powell Blvd	2016-09-13	151	VALIDATE	123.51711746686846	78.799664045550358	168.52084854459417
14	Central Park North & Adam Clayton Powell Blvd	2016-09-14	131	VALIDATE	124.037967425249	75.892099201604217	179.52084854459417



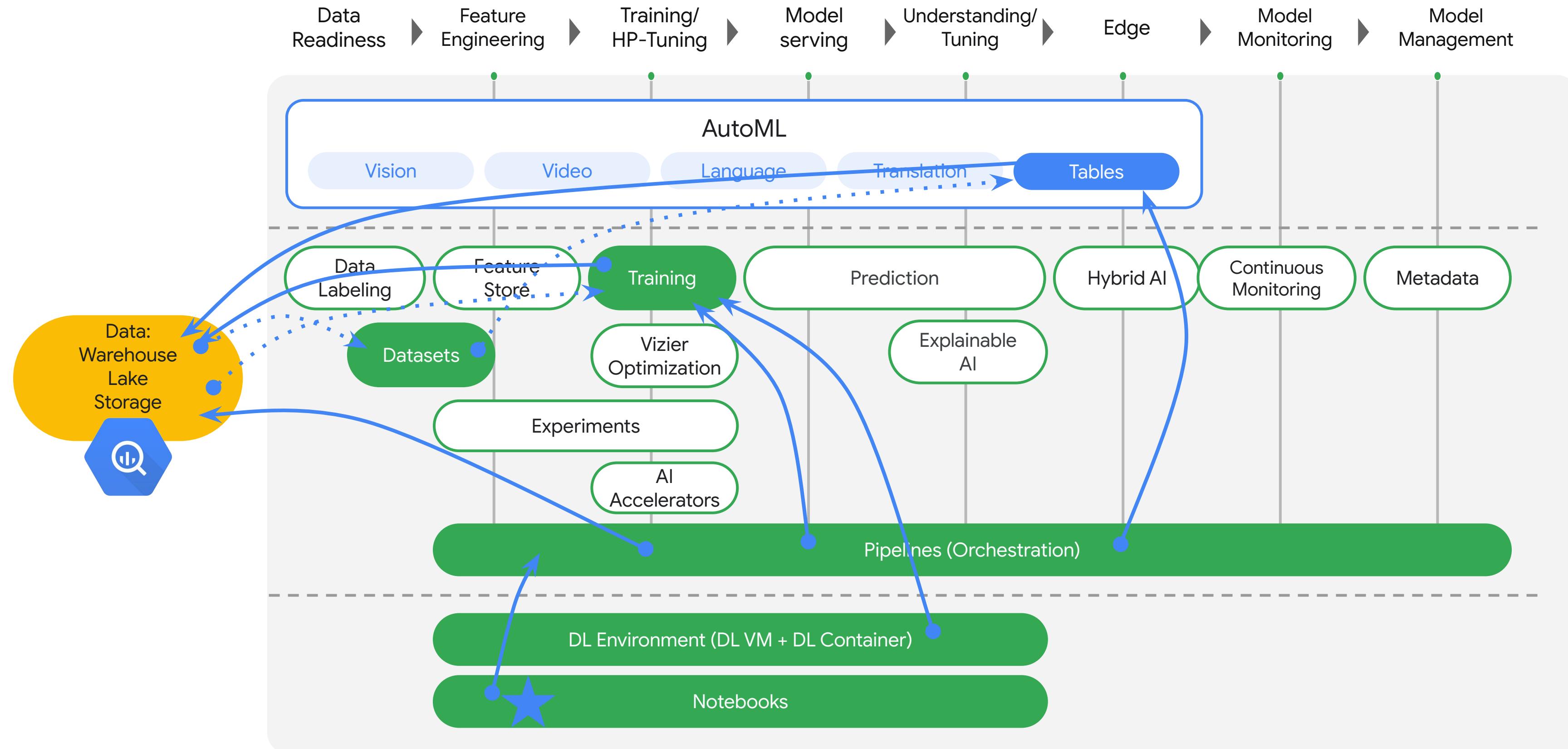
Time Series Forecasting

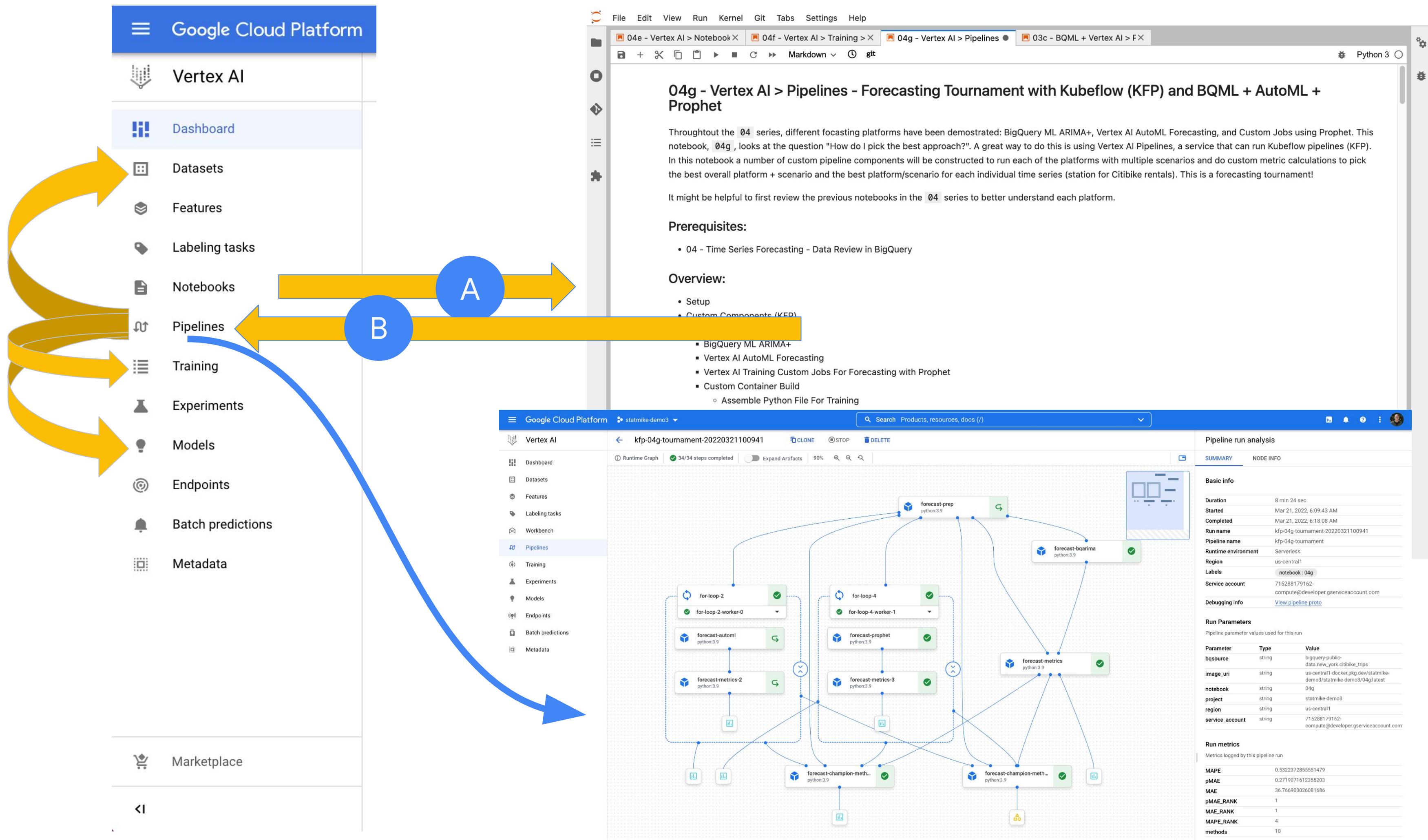
Vertex AI



Custom Job With Custom Container
Using Prophet in Vertex AI Training

04g

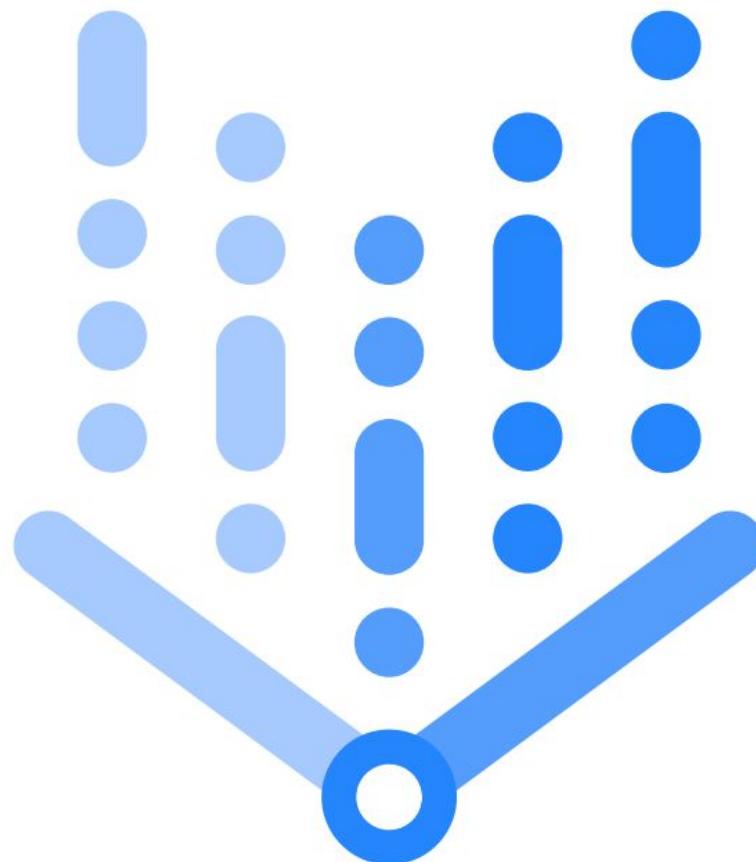






Time Series Forecasting

Vertex AI



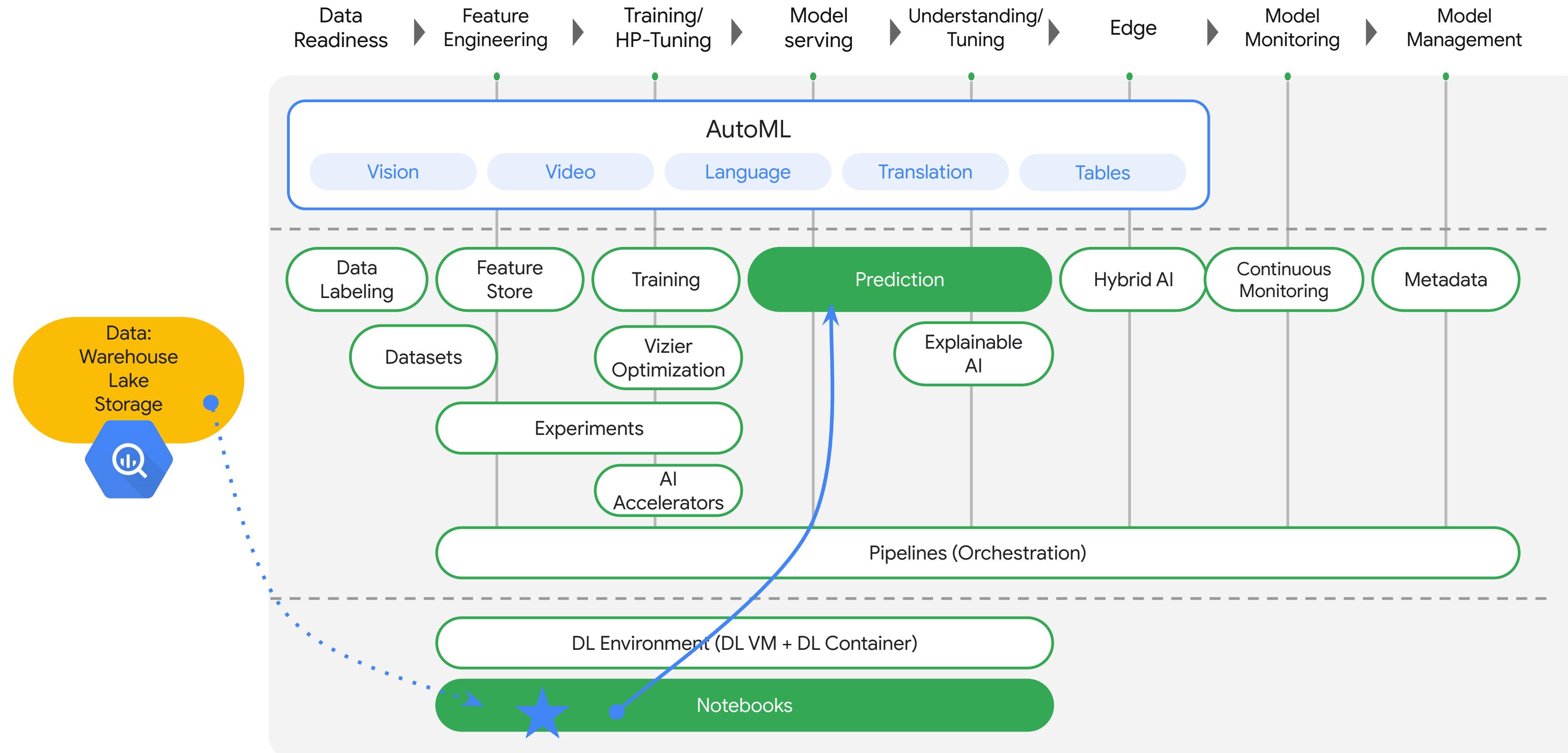
Forecast Tournament Pipeline

BigQuery ML ARIMA+, AutoML and Prophet

05

Notebook: 05

Vertex AI Overview



A

B

C

D

05 - Vertex AI > Notebooks - Models Built in Notebooks with Tensorflow

Where a model gets trained is where it consumes computing resources. With Vertex AI, you have choices for configuring the computing resources available at training. This notebook is an example of an execution environment. When it was set up there were choices for machine type and accelerators (GPUs).

This notebook shows training a model directly within the runtime of the notebook environment. Then the model is saved and moved to GCS for deployment to a Vertex AI Endpoint for online predictions. The model training is done with [Tensorflow](#), [Keras](#), and was designed to show a neural network approach to logistic regression. Training data batches are read from BigQuery using [Tensorflow I/O](#).

Prerequisites:

- 01 - BigQuery - Table Data Source

Overview:

- Use Python Client for BigQuery
 - Read the tables schema from BigQuery INFORMATION_SCHEMA
 - Prepare the feature information for Tensorflow
- Define a function that remaps the input data into features and target variables where target is one-hot encoded (classification model with 10 classes)
- Set Tensorflow I/O read session
- Demonstrate reading a single batch
- Train a Tensorflow model
 - Define the model layers



Custom Training



Vertex AI

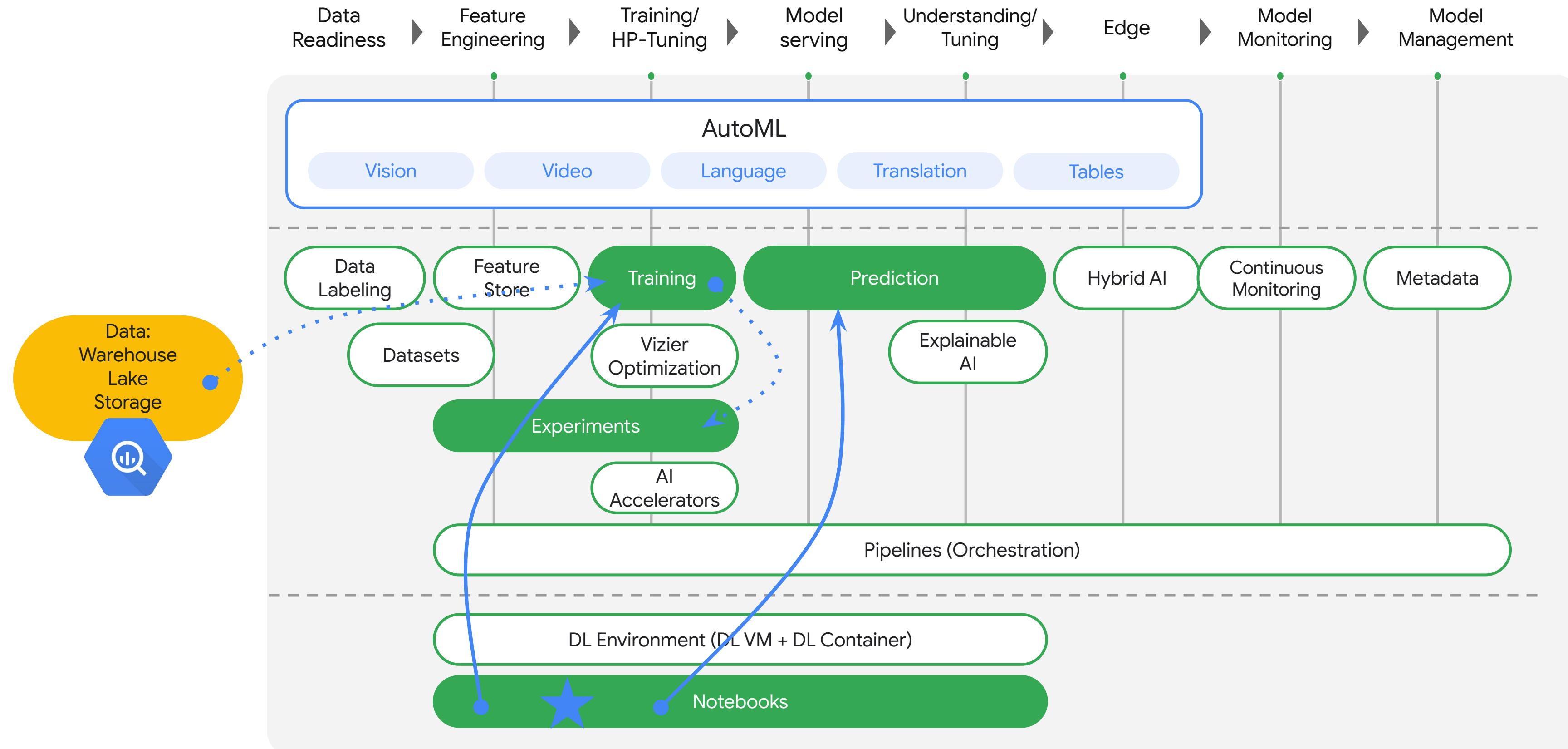


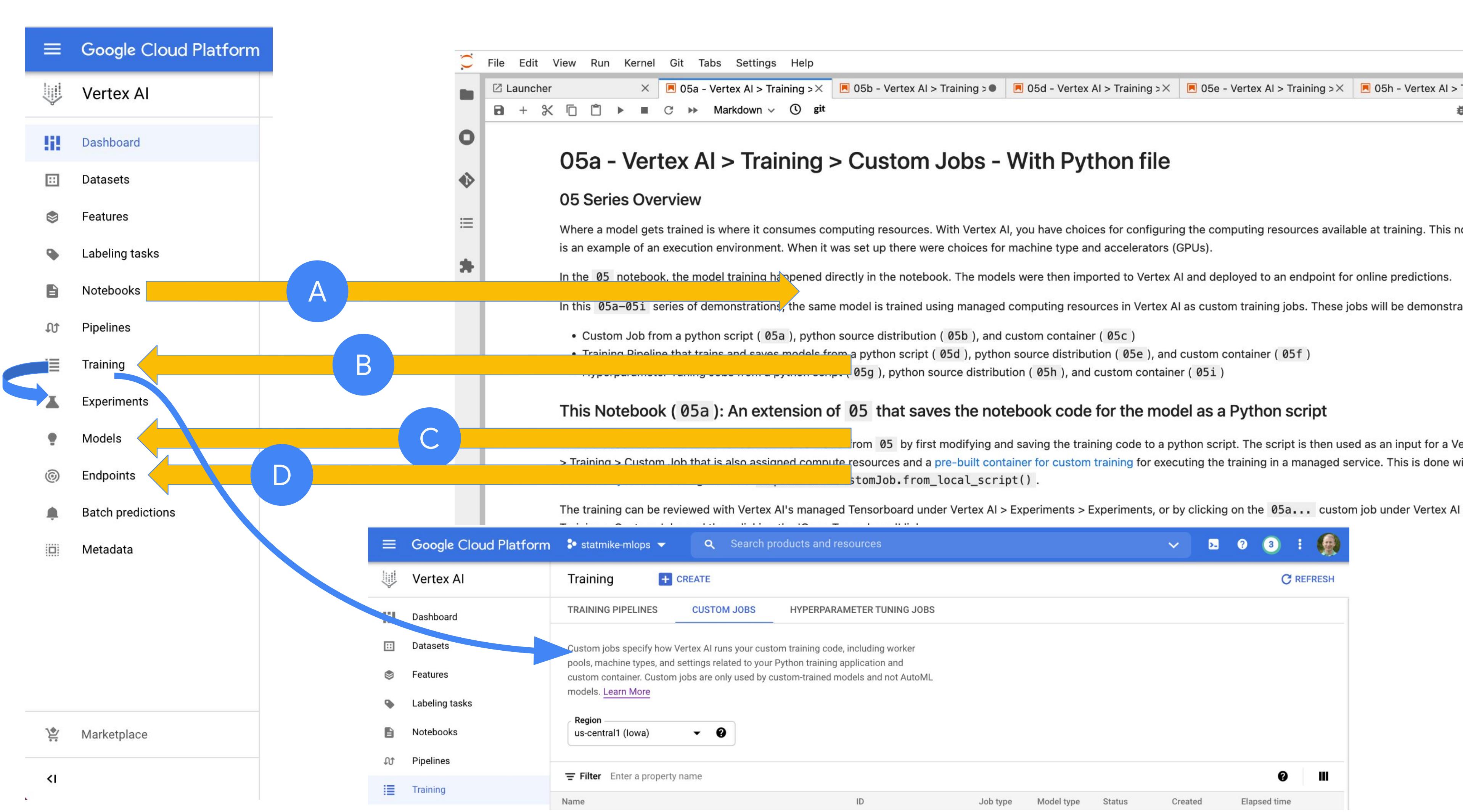
Custom Training in Notebook

05a

Notebook: 05a

Vertex AI Overview







Custom Training



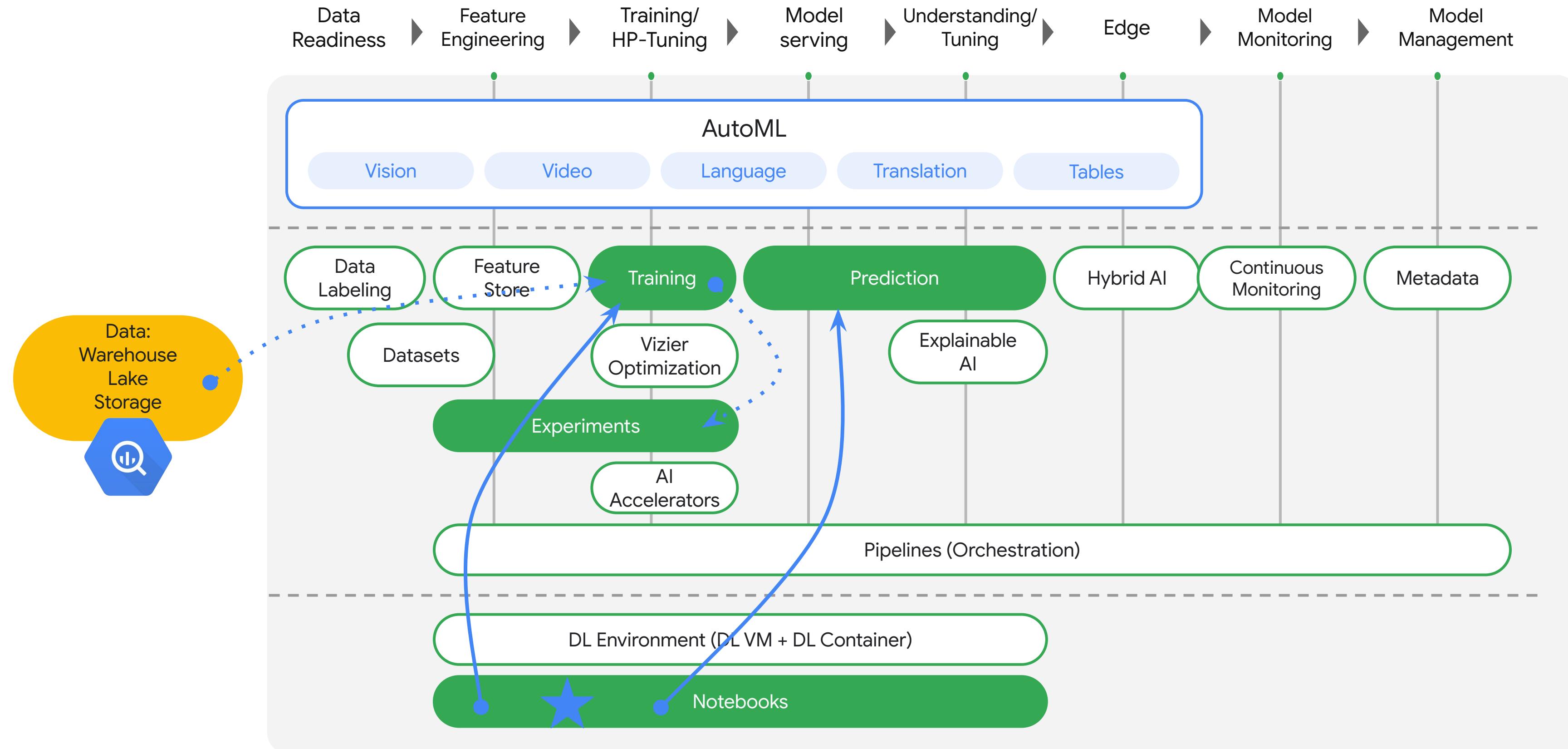
Vertex AI

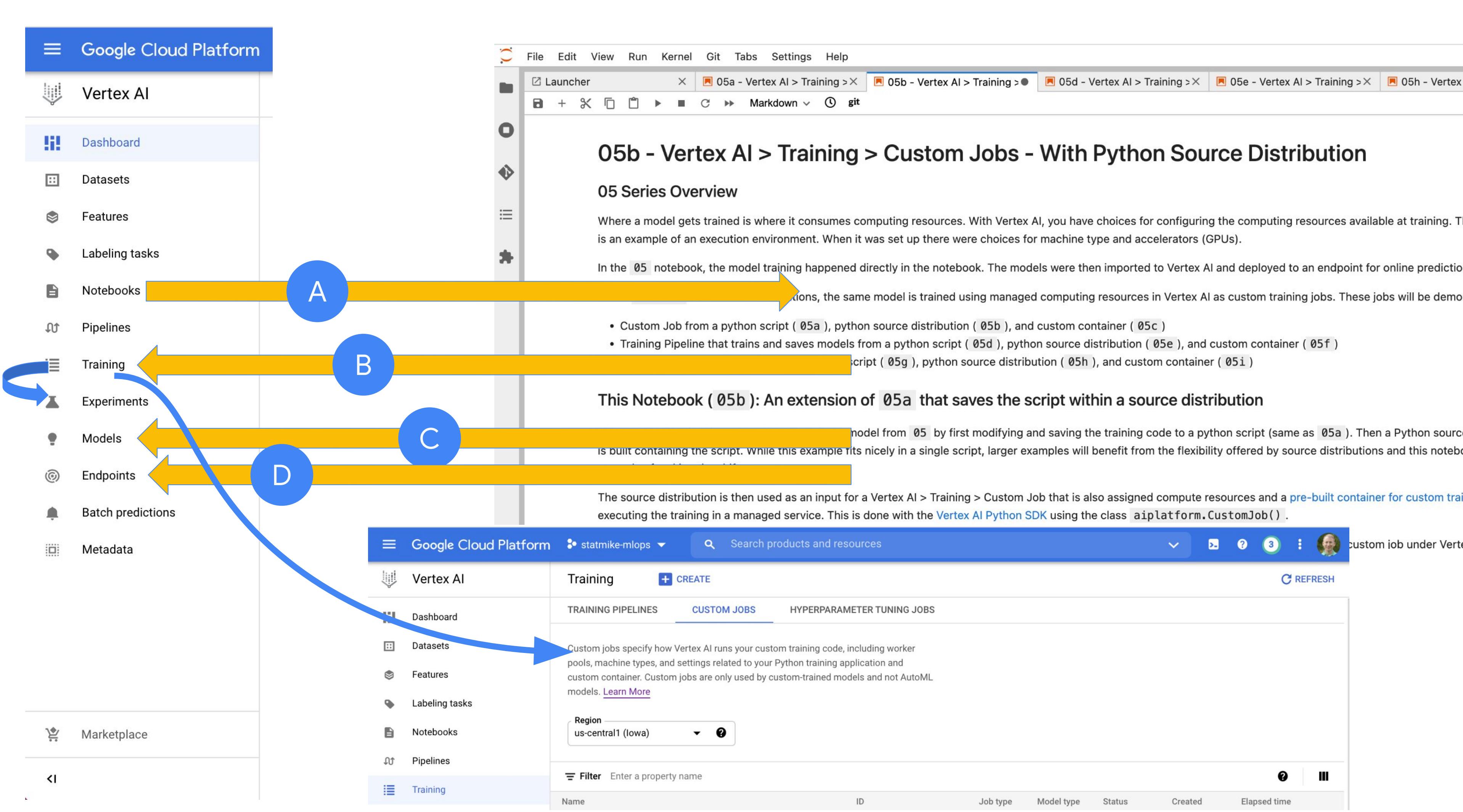


Custom Job

With Python File

05b







Custom Training



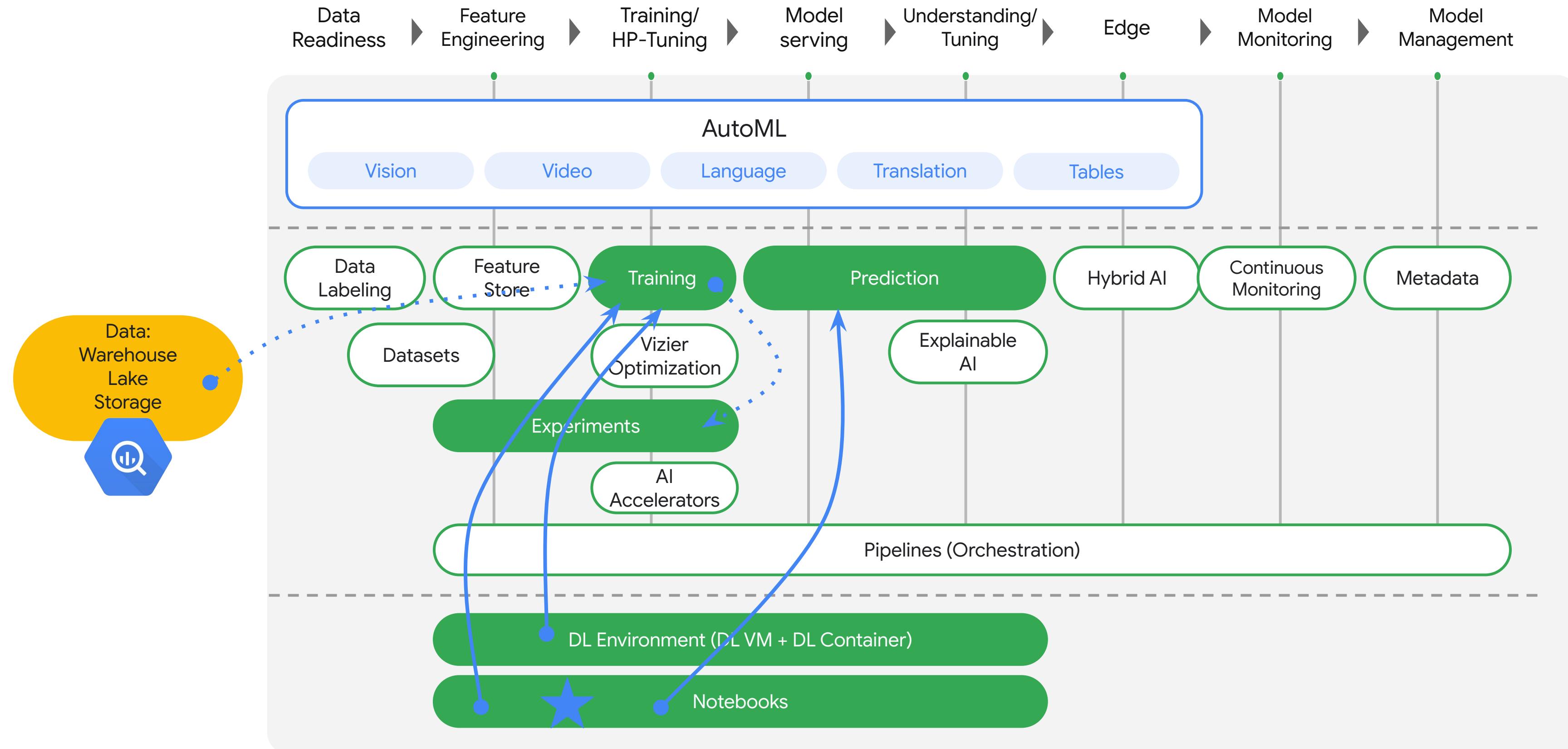
Vertex AI

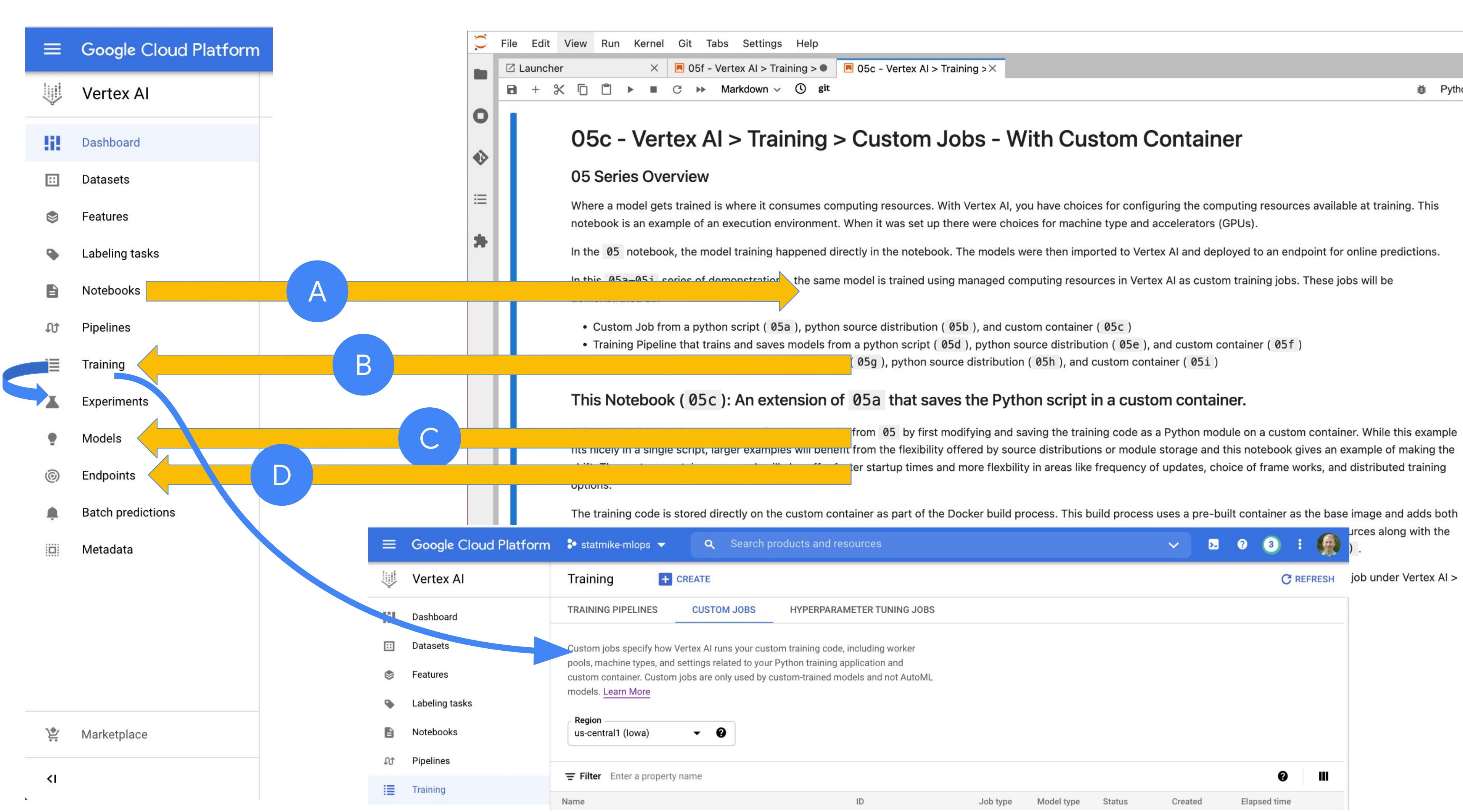


Custom Job

With Python Source Distribution

05C



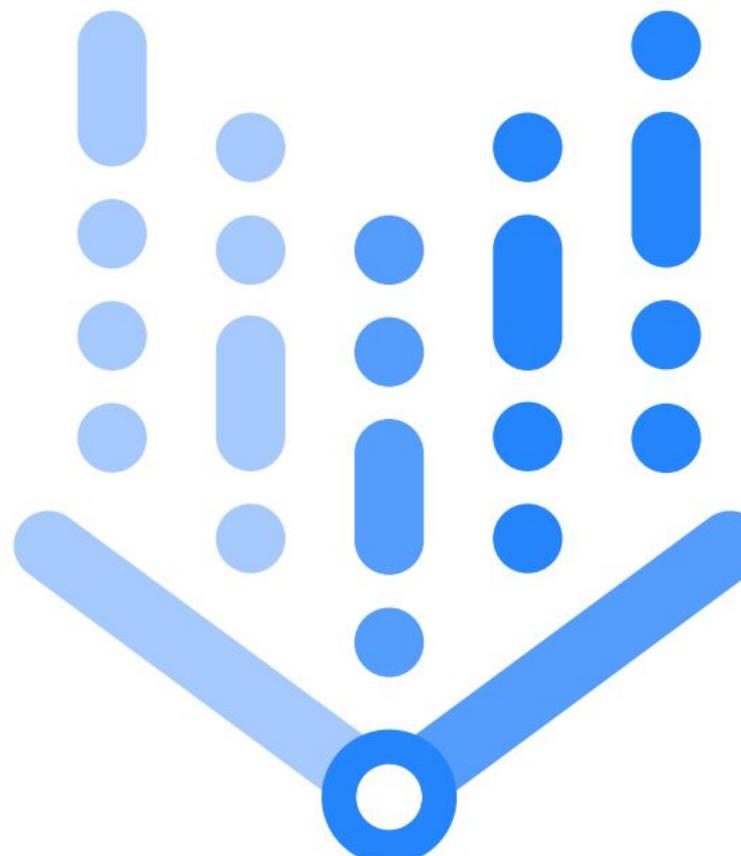




Custom Training



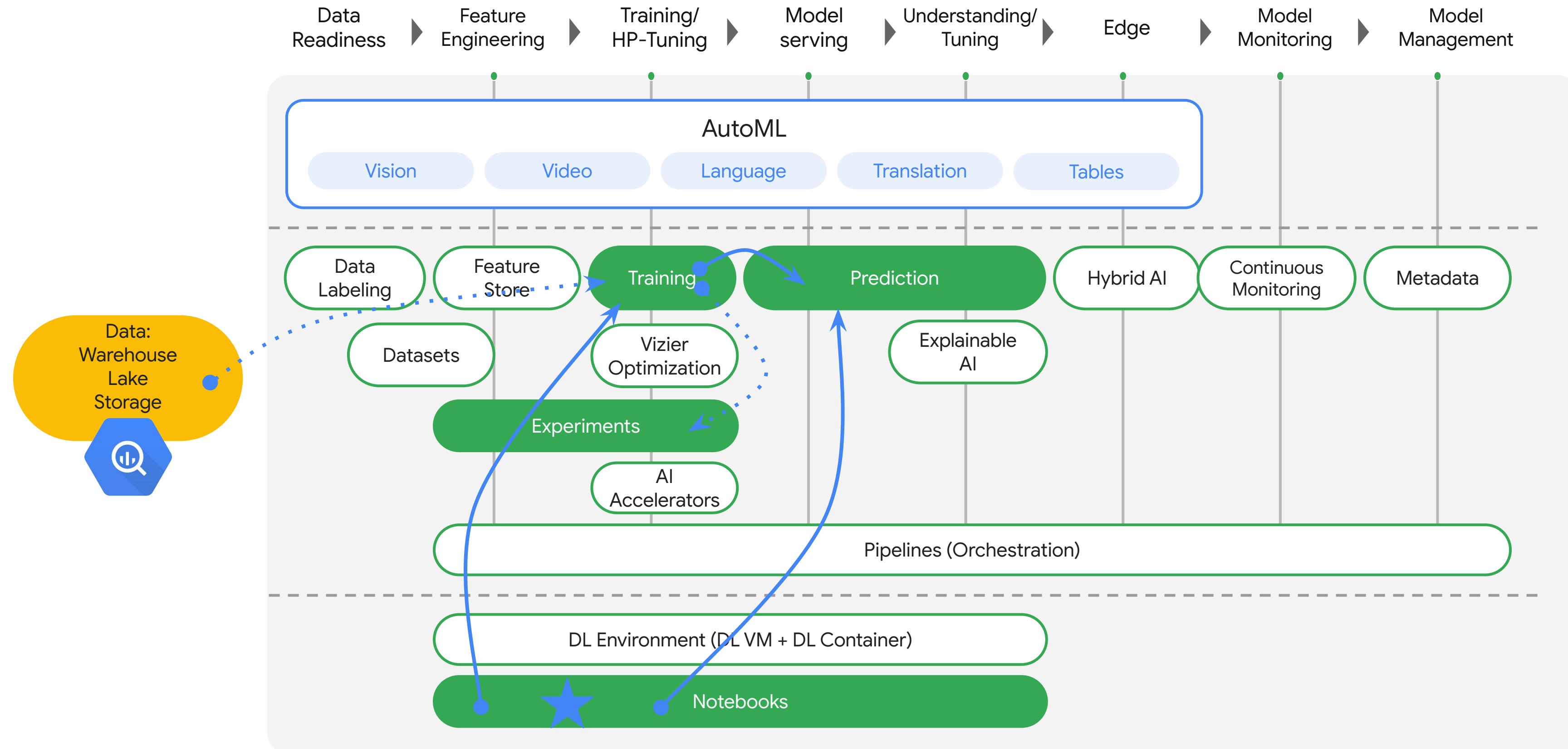
Vertex AI

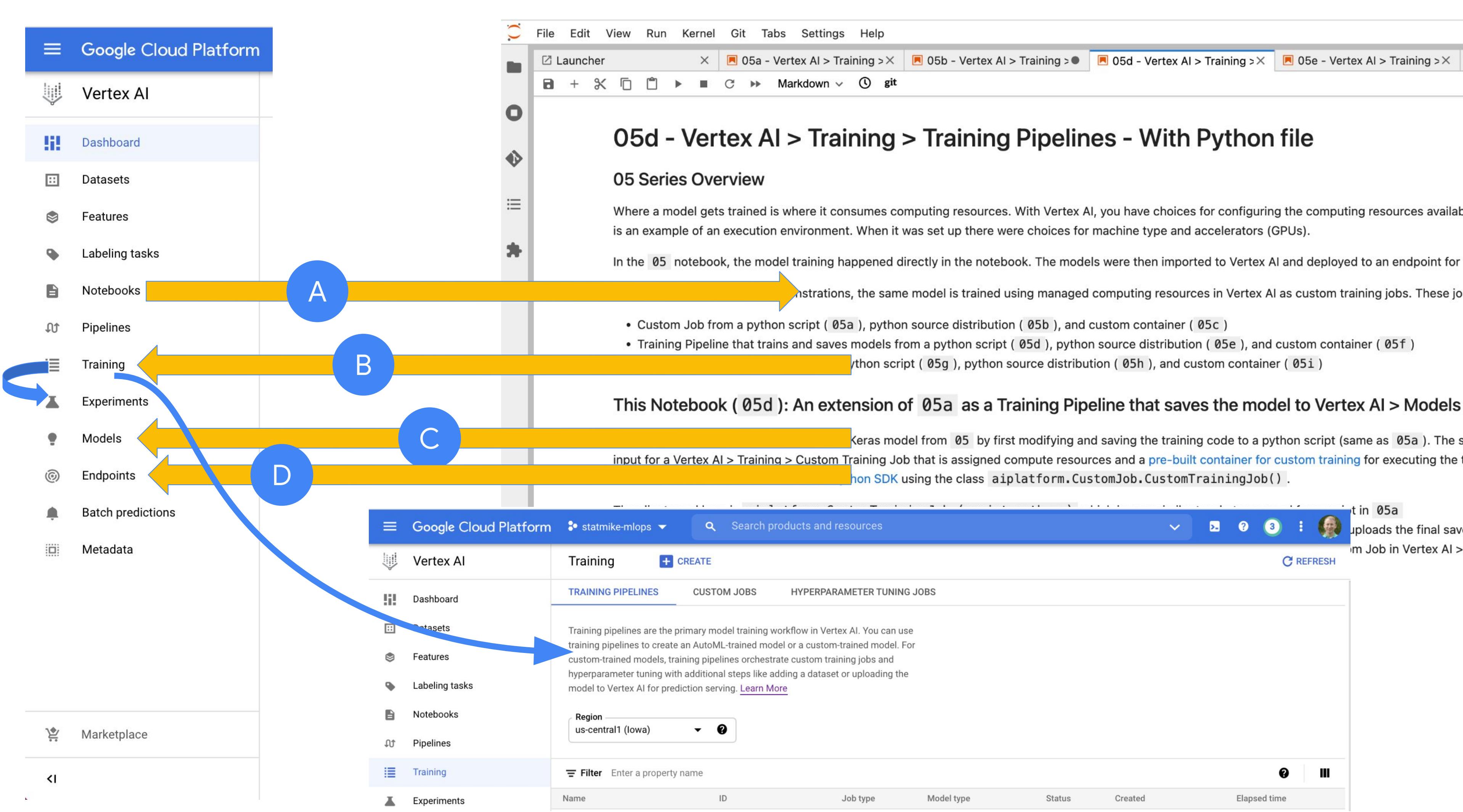


Custom Job

With Custom Container

05d



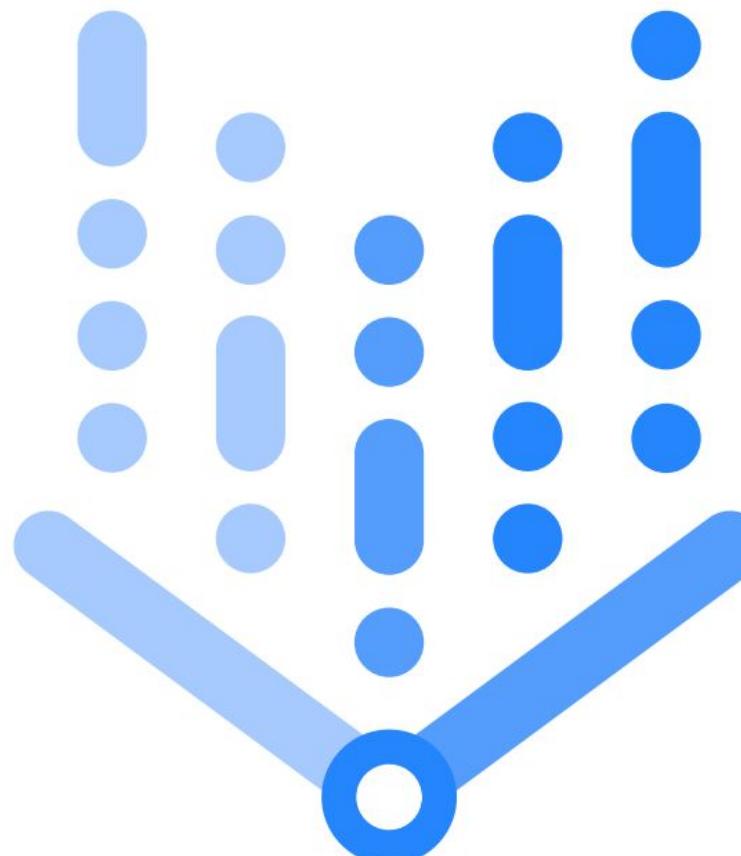




Custom Training

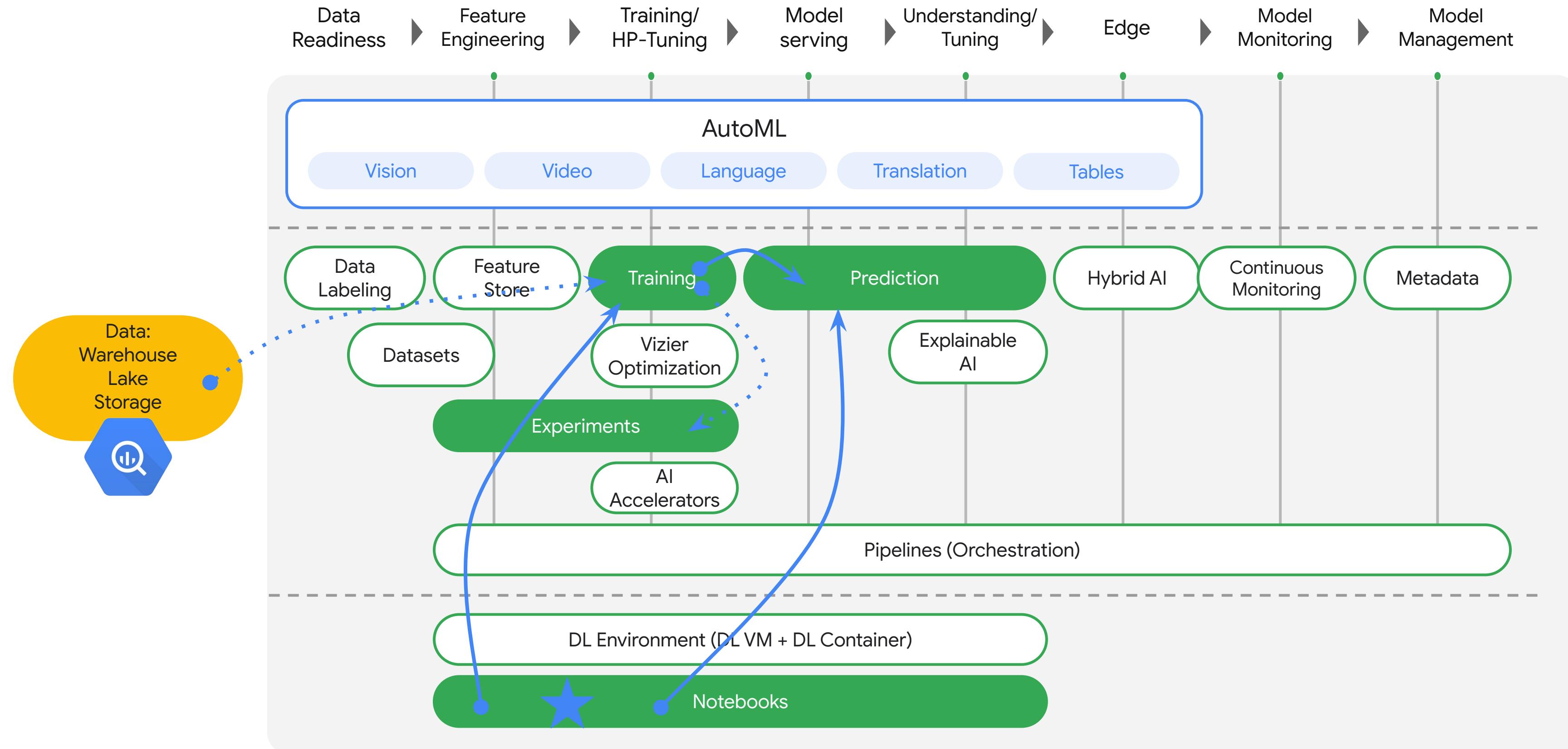


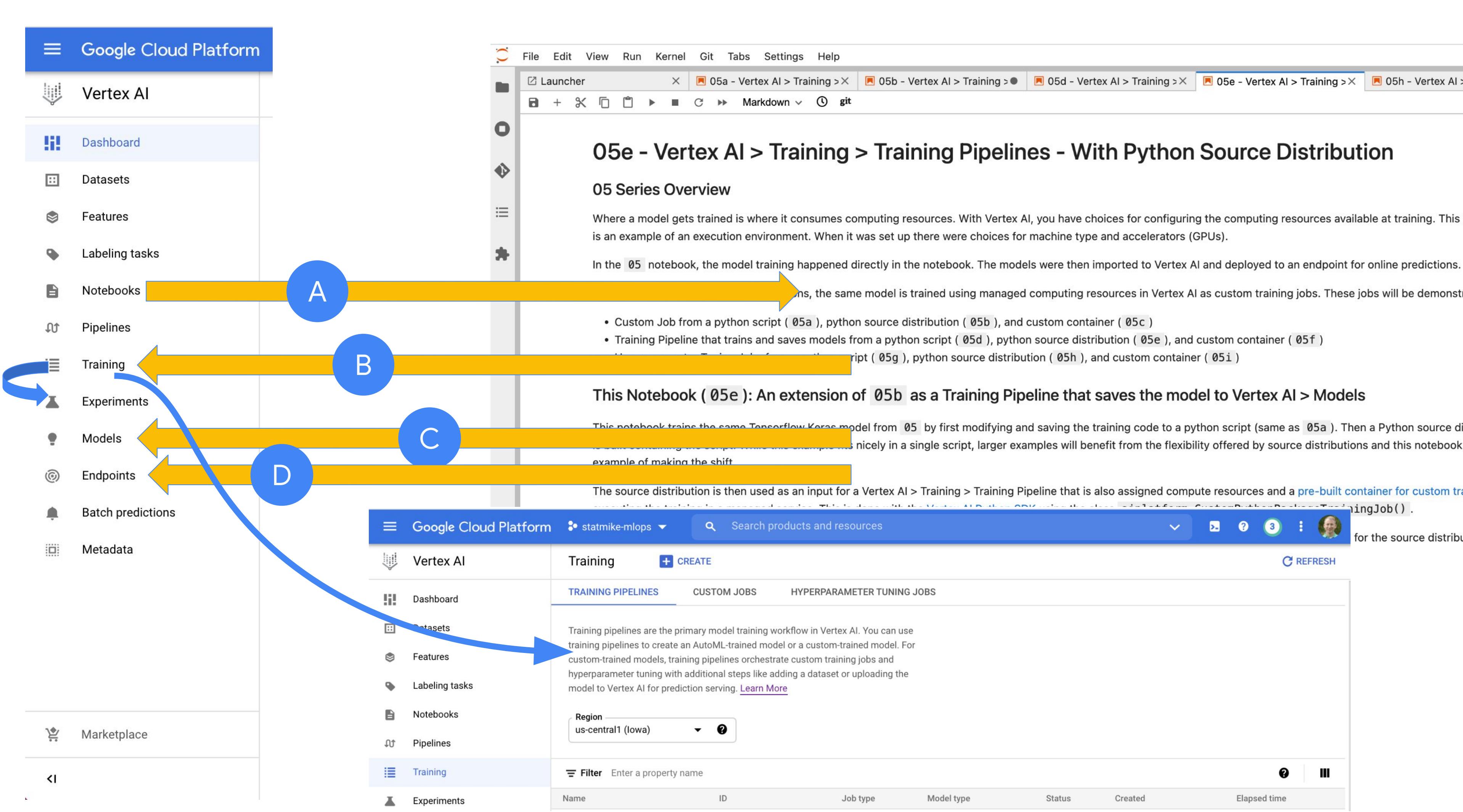
Vertex AI



Training Pipeline
With Python File

05e



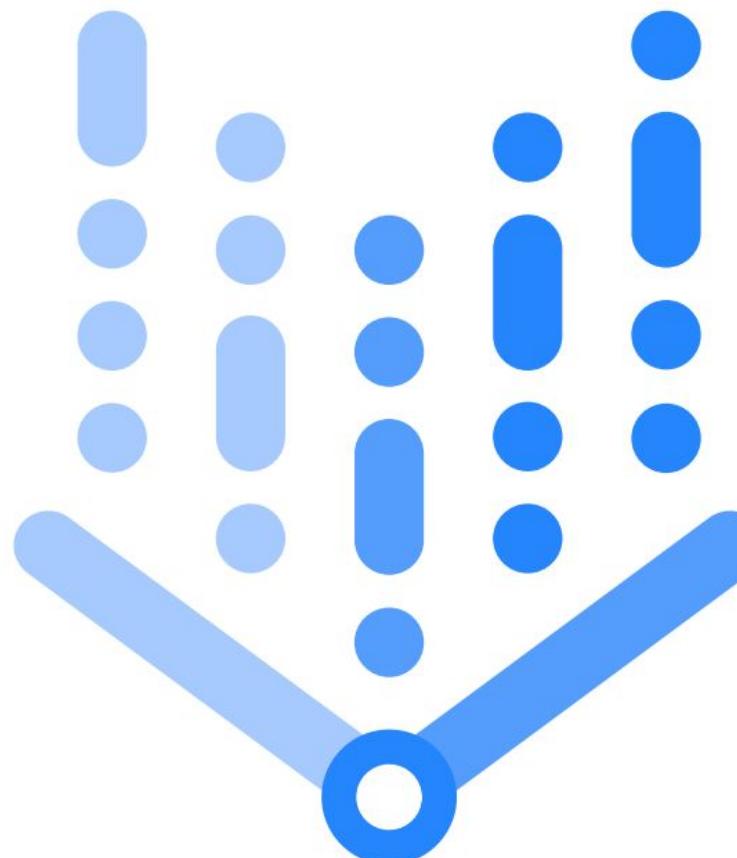




Custom Training

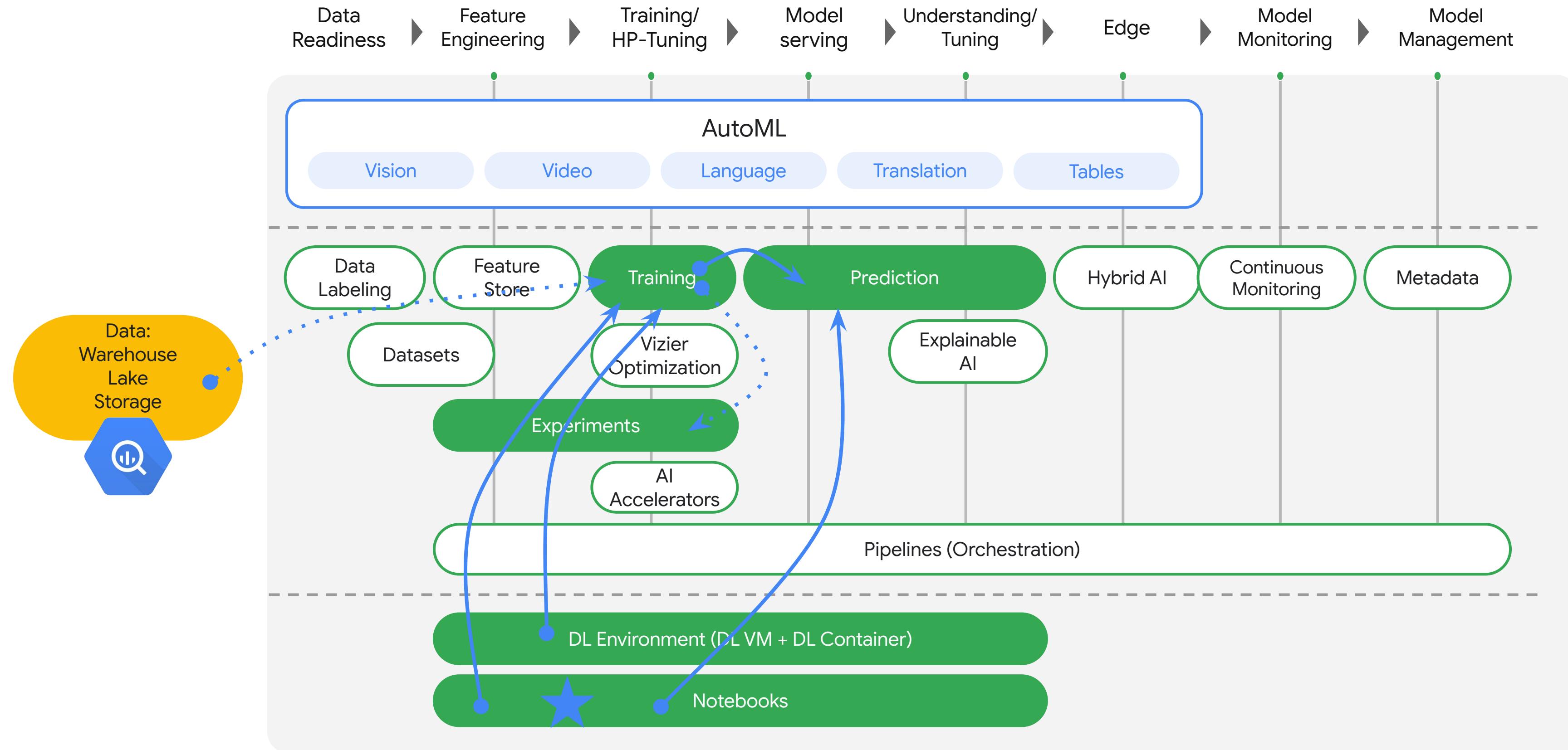


Vertex AI



**Training Pipeline
With Python Source Distribution**

05f



The diagram illustrates the Vertex AI training pipeline workflow, showing the progression from Notebooks to Training Pipelines, and how various components interact along the way.

Workflow Progression:

- Notebooks (A):** Represented by a blue circle at the top left. A yellow arrow points from this stage towards the right, indicating the flow of the process.
- Training (B):** Represented by a blue circle below Notebooks. A yellow arrow points from Notebooks to Training.
- Models (C):** Represented by a blue circle below Training. A yellow arrow points from Training to Models.
- Endpoints (D):** Represented by a blue circle below Models. A yellow arrow points from Models to Endpoints.

Annotations and Context:

- Section Header:** "05f - Vertex AI > Training > Training Pipelines - With Custom Container" (Top Right)
- Section Overview:** "05 Series Overview" (Top Left)
- Text:** "Where a model gets trained is where it consumes computing resources. With Vertex AI, you have choices for configuring the computing resources available at training. This notebook is..." (Top Left)
- Text:** "In the 05 notebook, the model training happened directly in the notebook. The models were then imported to Vertex AI and deployed to an endpoint for online predictions." (Top Left)
- Text:** "In this 05a–05i series of demonstrations, the same model is trained using managed computing resources in Vertex AI as custom training jobs. These jobs will be demonstrated as:" (Top Left)
- List:**
 - Custom Job from a python script (05a), python source distribution (05b), and custom container (05c)
 - Hyperparameter tuning jobs from a python script (05d), python source distribution (05e), and custom container (05f)
 - Hyperparameter tuning jobs from a python script (05g), python source distribution (05h), and custom container (05i)
- Section Description:** "This Notebook (05f): An extension of 05c as a Training Pipeline that saves the model to Vertex AI > Models" (Center Left)
- Text:** "From 05 by first modifying and saving the training code as a Python module on a custom container (same as 05c). While this nicely in a single script, larger examples will benefit from the flexibility offered by source distributions or module storage and this notebook gives an example of making the shift." (Center Left)
- Text:** "The training code is stored directly on the custom container as part of the Docker build process. This build process uses a pre-built container as the base image and adds both package dependencies and the training code to the container. This is then submitted as a Vertex AI > Training > Training Pipeline that is also assigned compute resources along with the custom container for execution. This is done with the [Vertex AI Python SDK](#) using the class `aiplatform.CustomContainerTrainingJob()`." (Center Left)
- Text:** "TrainingJob(container_uri =) which is very similar to what was used for custom container in 05c. The functional difference is that this method automatically uploads the final saved model to Vertex AI > Models. Running the job this way first triggers a job in Vertex AI > Training > Training Pipeline. This Training Pipeline then triggers a job in Vertex AI > Models to save the final trained model." (Center Left)
- Text:** "Under Vertex AI > Training > Training Pipelines, you can see the training pipeline that was created. You can also see the status of the pipeline and the status of the training job. You can also see the status of the training job and the status of the training pipeline." (Bottom Right)



Custom Training



Vertex AI

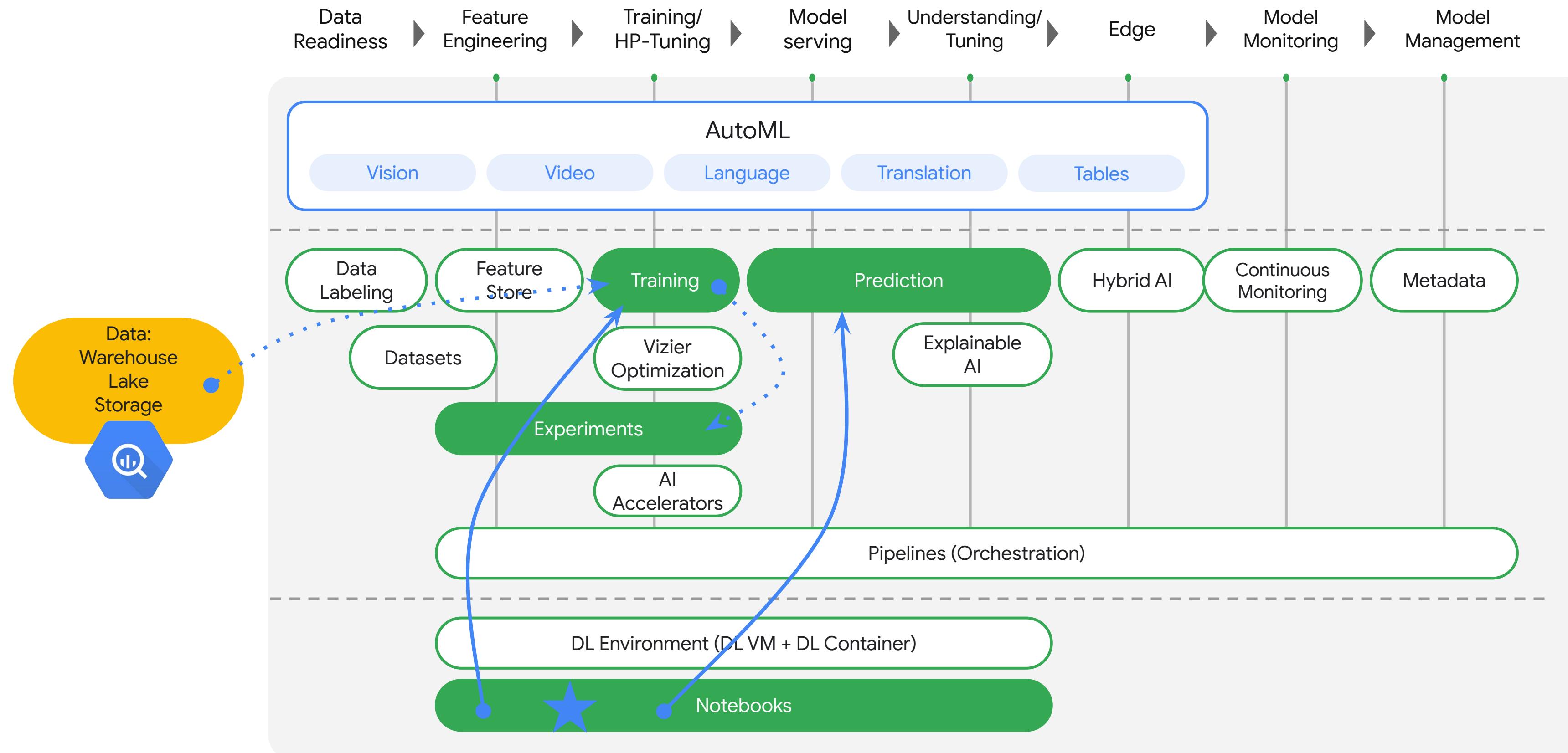


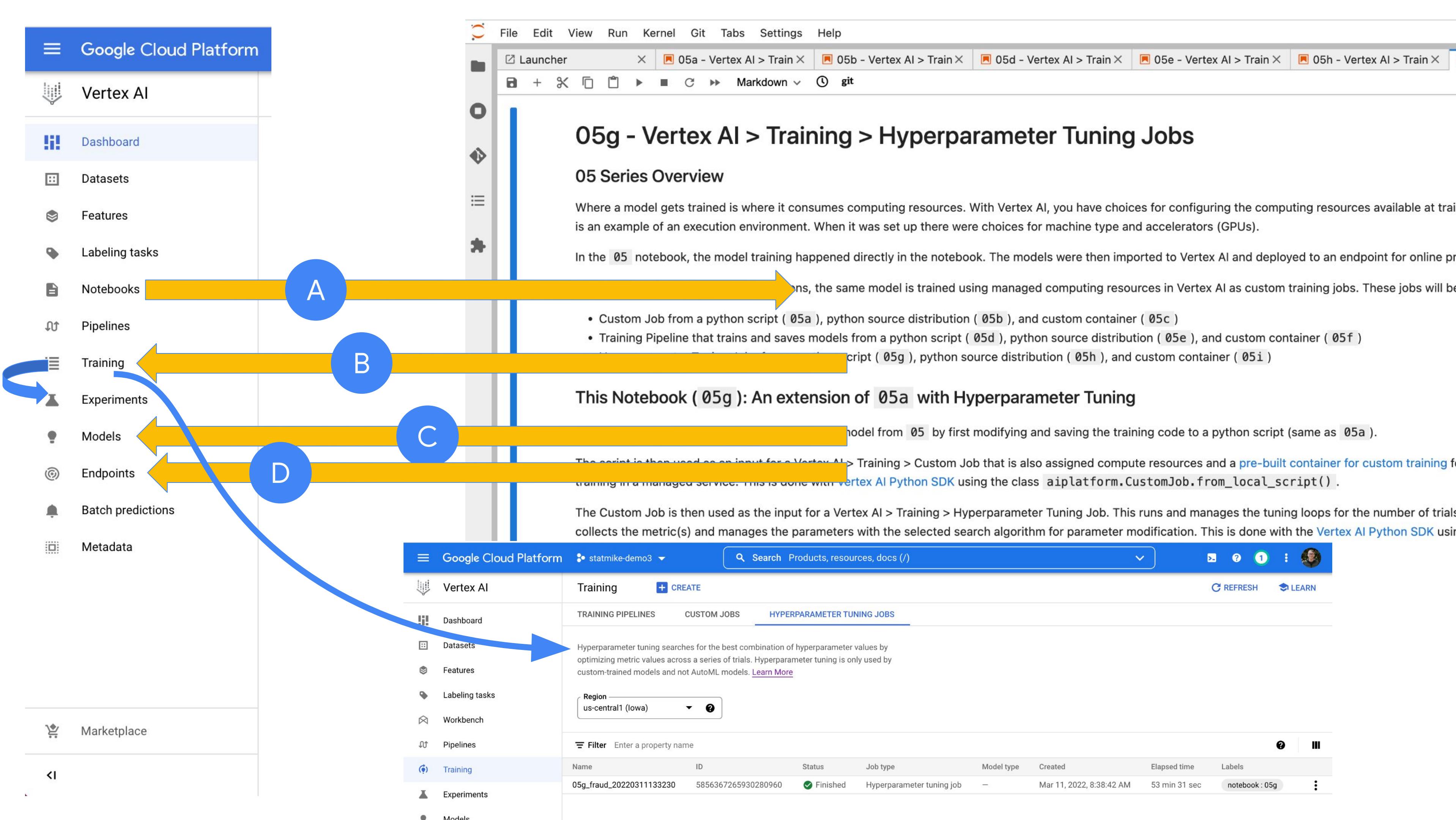
**Training Pipeline
With Custom Container**

05g

Notebook: 05g

Vertex AI Overview







Custom Training

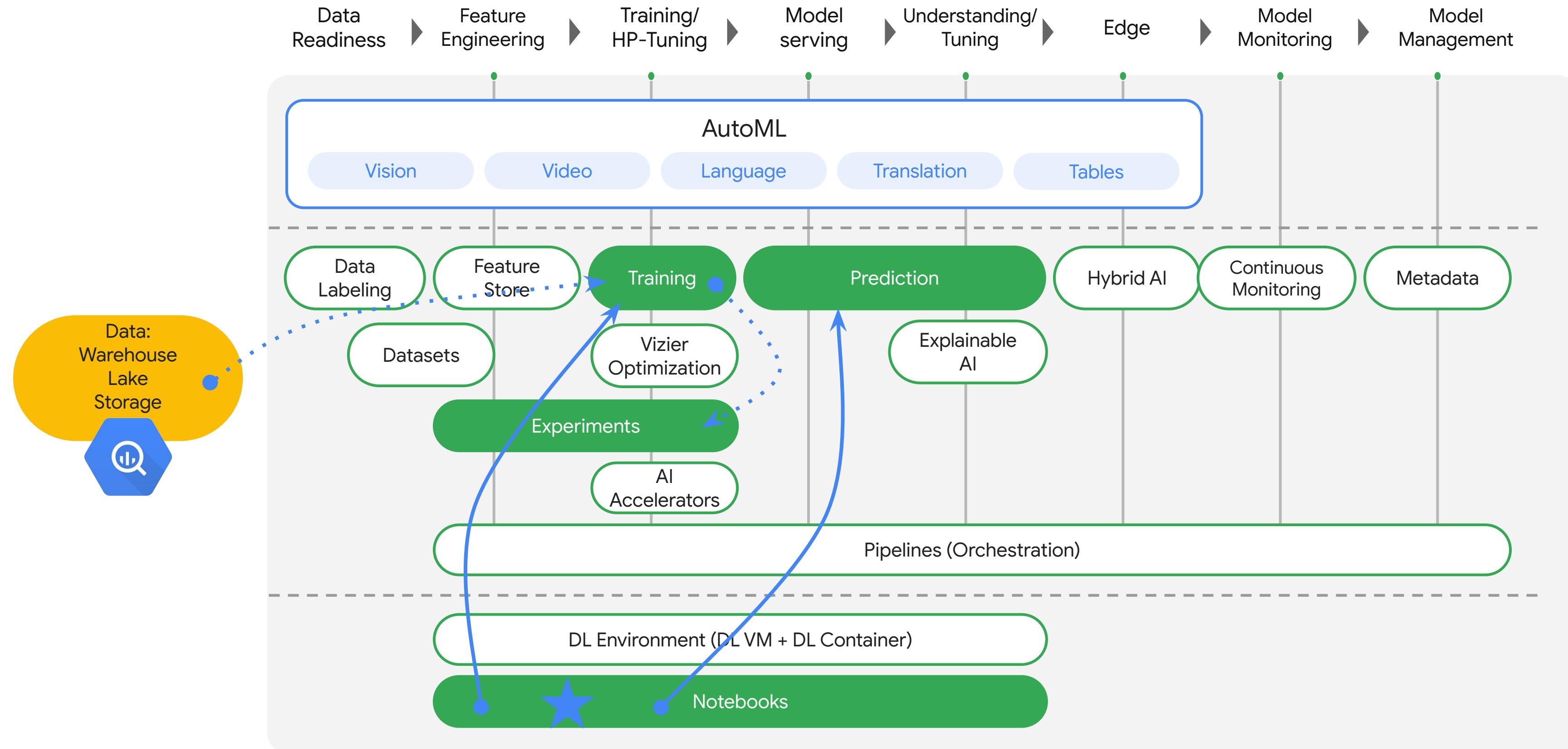


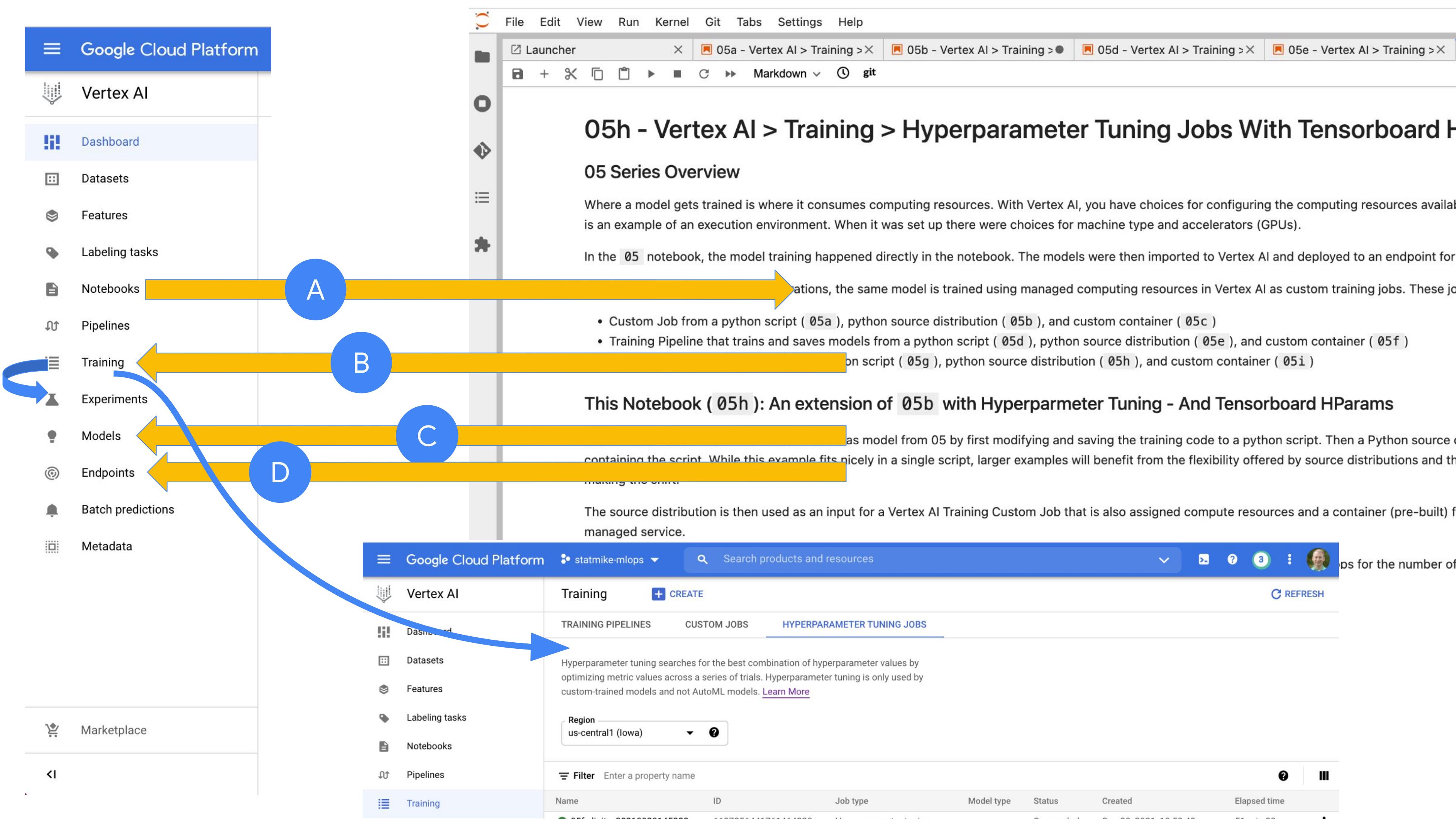
Vertex AI



**Hyperparameter Tuning Job
With Python File**

05h







Custom Training

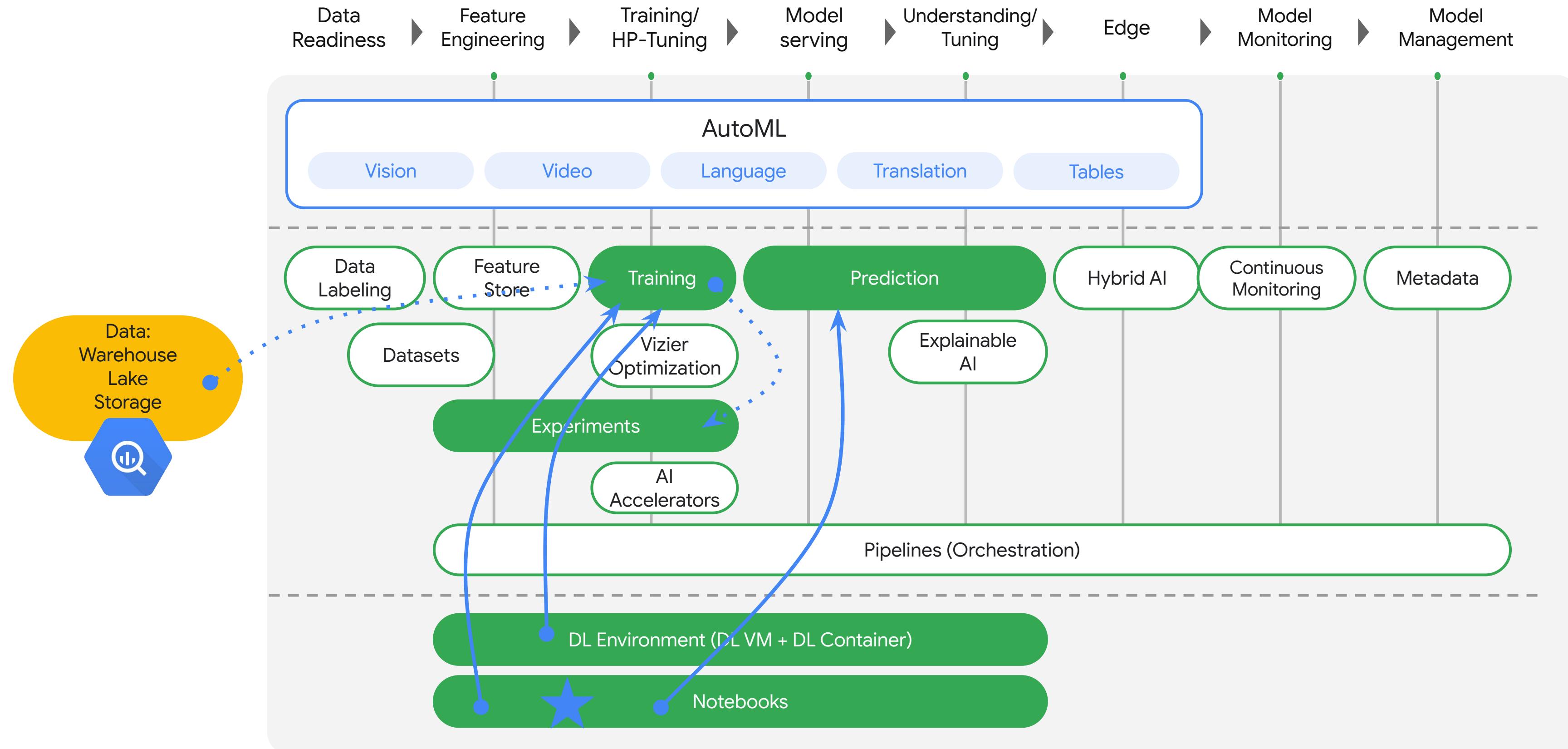


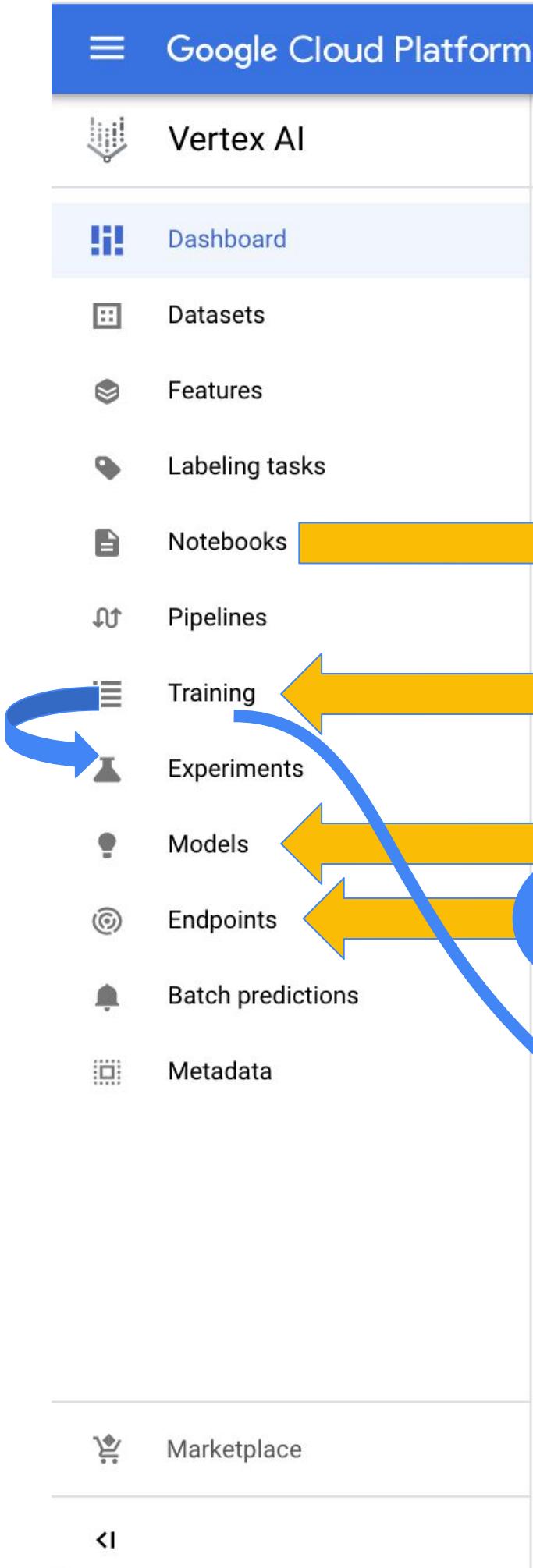
Vertex AI



**Hyperparameter Tuning Job
With Python Source Distribution**

05i





File Edit View Run Kernel Git Tabs Settings Help

05i - Vertex AI > Training > X

Markdown git

05i - Vertex AI > Training > Hyperparameter Tuning Jobs - With Custom Container

05 Series Overview

Where a model gets trained is where it consumes computing resources. With Vertex AI, you have choices for configuring the computing resources available at training. This notebook is an example of an execution environment. When it was set up there were choices for machine type and accelerators (GPUs).

In the [05](#) notebook, the model training happened directly in the notebook. The models were then imported to Vertex AI and deployed to an endpoint for online predictions.

This notebook ([05i](#)) is an extension of [05c](#) with Hyperparameter Tuning - And Tensorboard HParams

The training code is stored directly on the custom container as part of the Docker build process. This build process uses a pre-built container as the base image and adds both packages and the training code as a Python module. This container is specified in the setup of a custom training job and also assigned compute resources

Hyperparameter tuning searches for the best combination of hyperparameter values by optimizing metric values across a series of trials. Hyperparameter tuning is only used by custom-trained models and not AutoML models. [Learn More](#)

Region us-central1 (Iowa)

Filter Enter a property name



Custom Training



Vertex AI

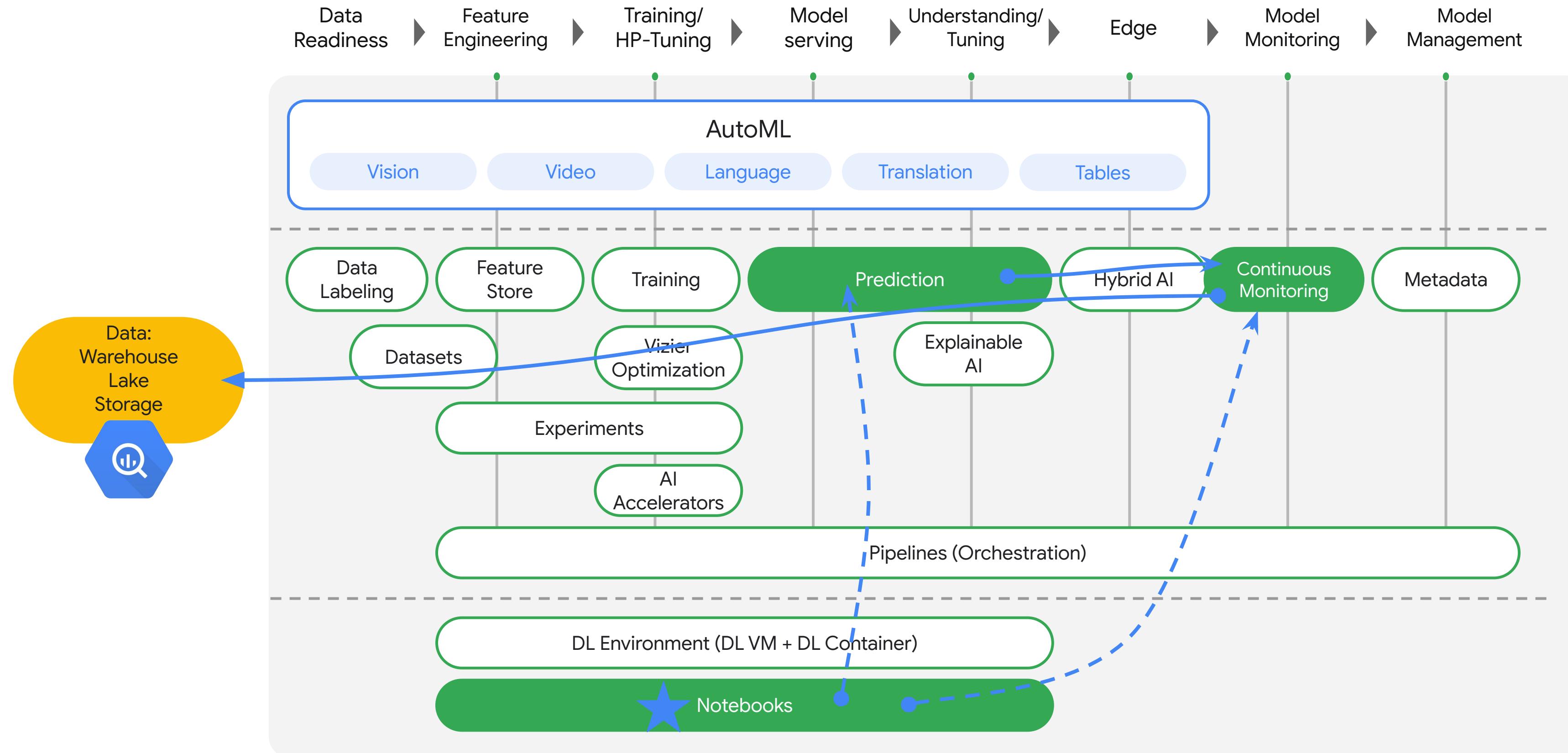


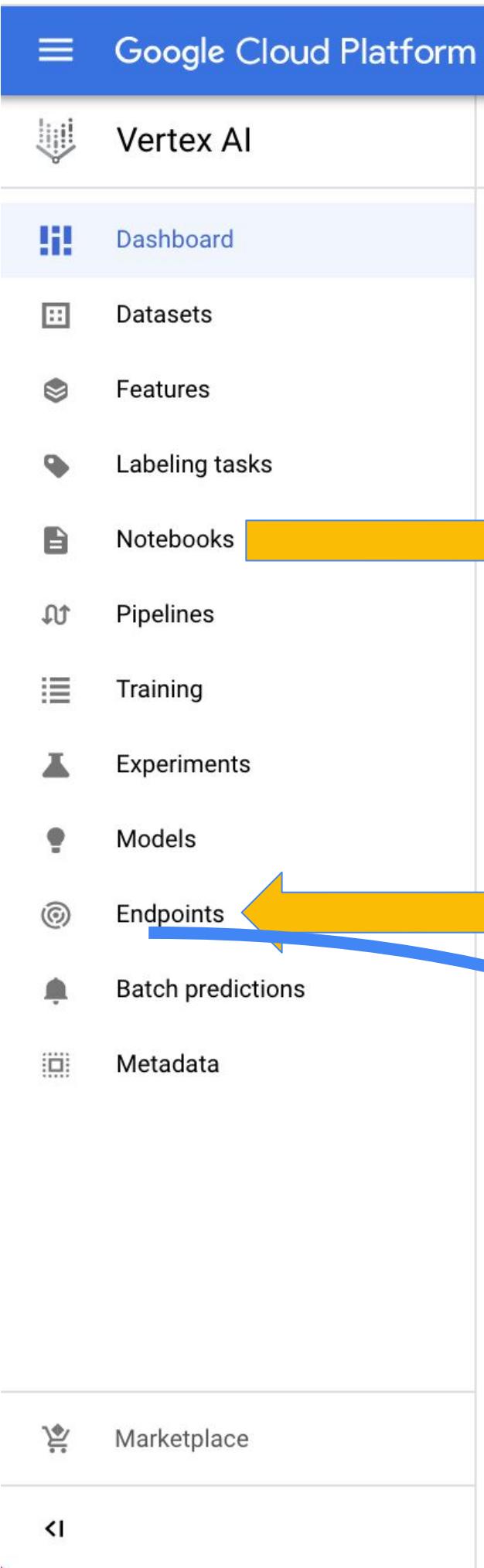
**Hyperparameter Tuning Job
With Custom Container**

óá

Notebook: 06a

Vertex AI Overview





06a - Vertex AI > Model Monitoring

In other notebooks the end-to-end workflows include serving trained ML models on Vertex AI endpoints. In this notebook, an endpoint will be extended by enabling model monitoring. This enables continuous scheduled monitoring of selected model features for deviations:

- Training-serving skew: feature distribution is different from the feature distribution in the training data
- Prediction drift: feature distribution is different over time

The monitoring is set up with a threshold that is used to create alerts

- numerical features difference is calculated with Jensen-Shannon divergence
- categorical features difference is calculated with L-infinity distance

Prerequisites:

- 02a - Vertex AI - AutoML in GCP Console (no code)
 - or any other notebook that creates a Vertex AI Endpoint
 - Picking 02a because it has an endpoint setup

Overview:

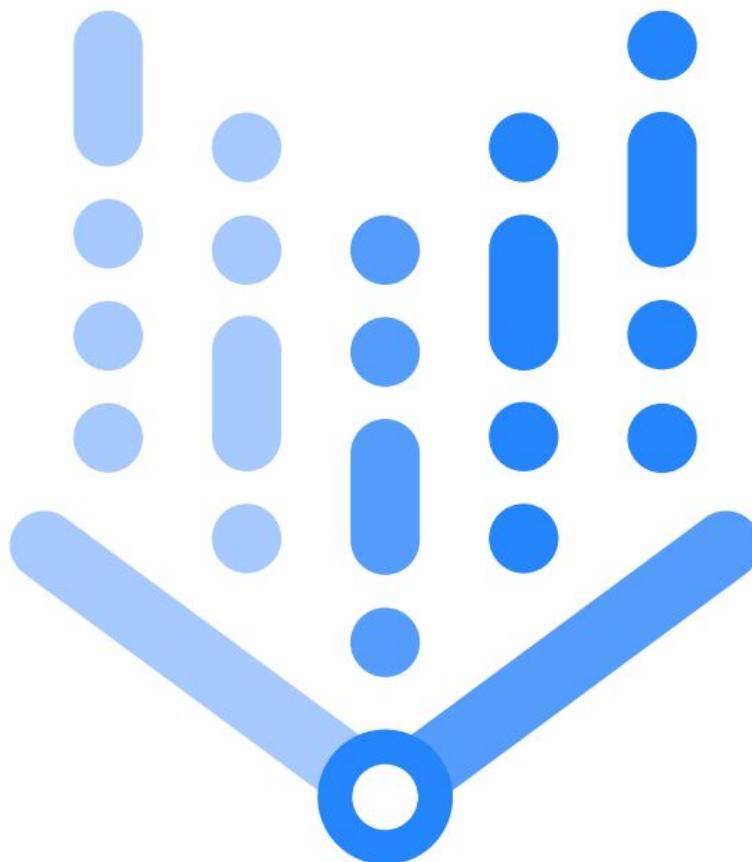
- Find Existing Endpoint
- Prediction from endpoint using Python API
- Setup Monitoring Client
- Setup Monitoring Job
- Run Prediction with Training Data
 - Review Alerts
- Run Predictions with Test Data
 - Review Alerts
- Extended Run of Predictions with Noise
 - Review Alerts and Distributions
- Pause and Delete Monitoring Job

The screenshot shows the Google Cloud Platform Vertex AI Model Monitoring interface. It displays two charts: 'Feature distribution' and 'Latest prediction stats distribution'. The 'Feature distribution' chart shows a bar chart for 'Feature: Amount' with a distribution deviation value of 0.45 and a detection threshold of 0.001. The 'Latest prediction stats distribution' chart shows a histogram of monitoring jobs from February 25, 2022, at 2:00:00 AM to 3:00:00 PM. The interface also includes a sidebar with navigation links like Dashboard, Datasets, Features, Labeling tasks, Workbench, Pipelines, Training, Experiments, Models, Endpoints (highlighted), Batch predictions, and Metadata.



Continuous Monitoring

Vertex AI

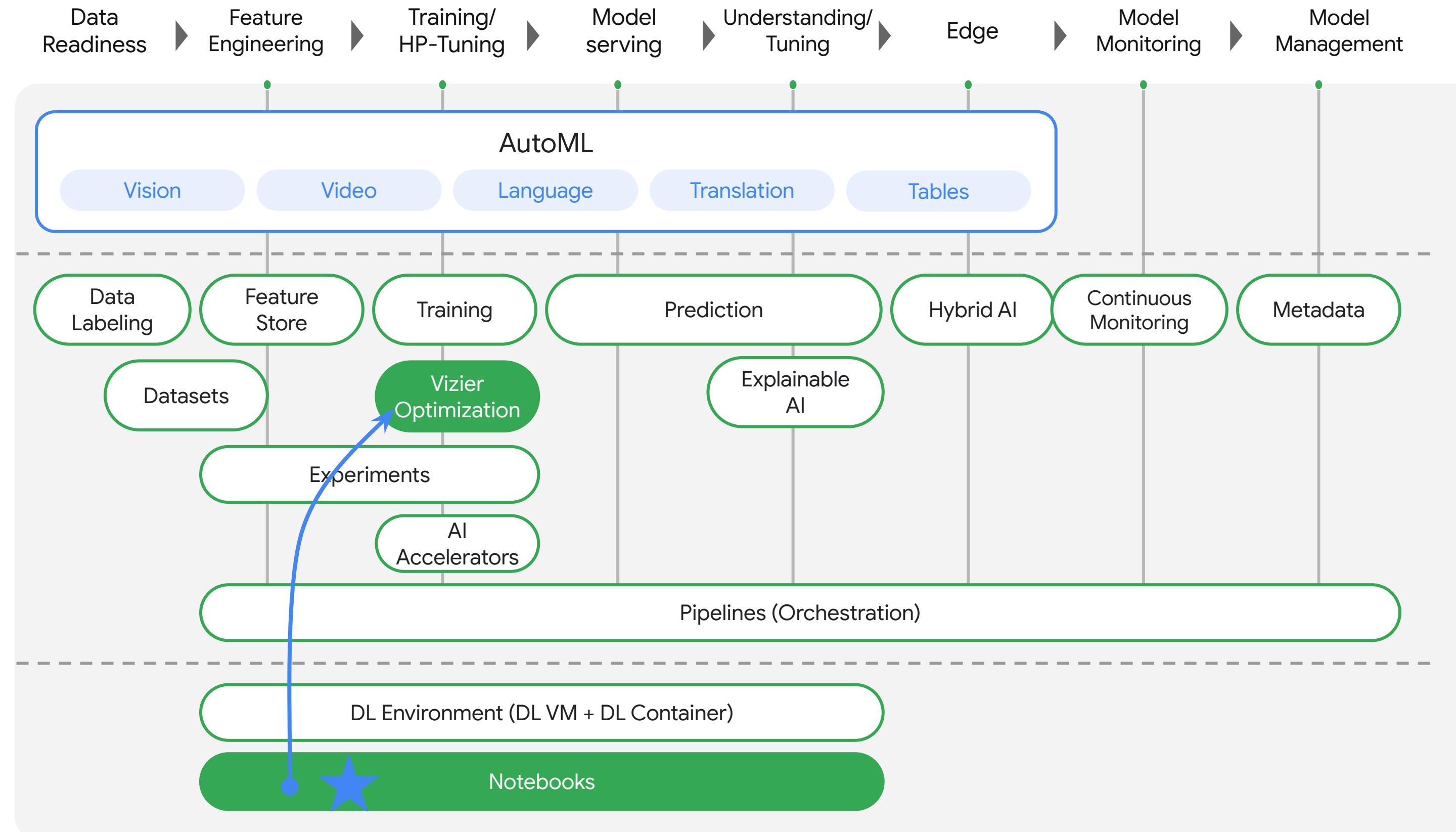


Model Skew and Drift

10

Notebook: 10

Vertex AI Overview



Google Cloud Platform

Vertex AI

- Dashboard
- Datasets
- Features
- Labeling tasks
- Notebooks**
- Pipelines
- Training
- Experiments
- Models
- Endpoints
- Batch predictions
- Metadata
- Marketplace

File Edit View Run Kernel Git Tabs Settings Help

Launcher X 05a - Vertex AI X 05b - Vertex AI X 05d - Vertex AI X 05e - Vertex AI X 05h - Vertex AI X 06a - Vertex AI X 10 - Vertex AI > ●

+

10 - Vertex AI > Experiments > Studies - Vizier Optimization Service

Vertex AI Vizier is an optimization service. It is used to optimize hyperparameters for machine learning models - called hyperparameter tuning. It can also optimize any system that can be evaluated. Even systems with multiple objectives.

In this demonstration, multiple objectives are set and the Vizier service is used to conduct a random search and a default search (Bayesian Optimization) for comparison.

To see an example of hyperparameter tuning see notebooks 05g, 05h, and 05i. Those notebooks use the aiplatform.HyperparameterTuningJob() to manage the process rather than interacting with the Vertex AI Vizier service directly. Also see [this example](#).

Prerequisites:

- 00 - Environment Setup

Overview:

- Minimize one function
- Maximize another function with same inputs
- Define a Vertex AI Vizier study with the parameters and metrics with objectives
- Define a trial loop

Google Cloud Platform statmike-mlops Search products and resources REFRESH

Vertex AI Studies

EXPERIMENTS PREVIEW STUDIES PREVIEW TENSORBOARD INSTANCES PREVIEW

Vertex Vizier is an optimization service that helps you tune hyperparameters in complex machine learning models. [Learn more](#)

Region us-central1 (Iowa)

Filter Enter property name or value

Study name	ID	Objective	Created
Study_06_Bayesian_Optimization	4214226082825	Minimize "blue" and Maximize "green"	Sep 16, 2021, 11:51:29 AM
Study_06_Random	639592116037	Minimize "blue" and Maximize "green"	Sep 16, 2021, 11:44:46 AM



Vertex AI

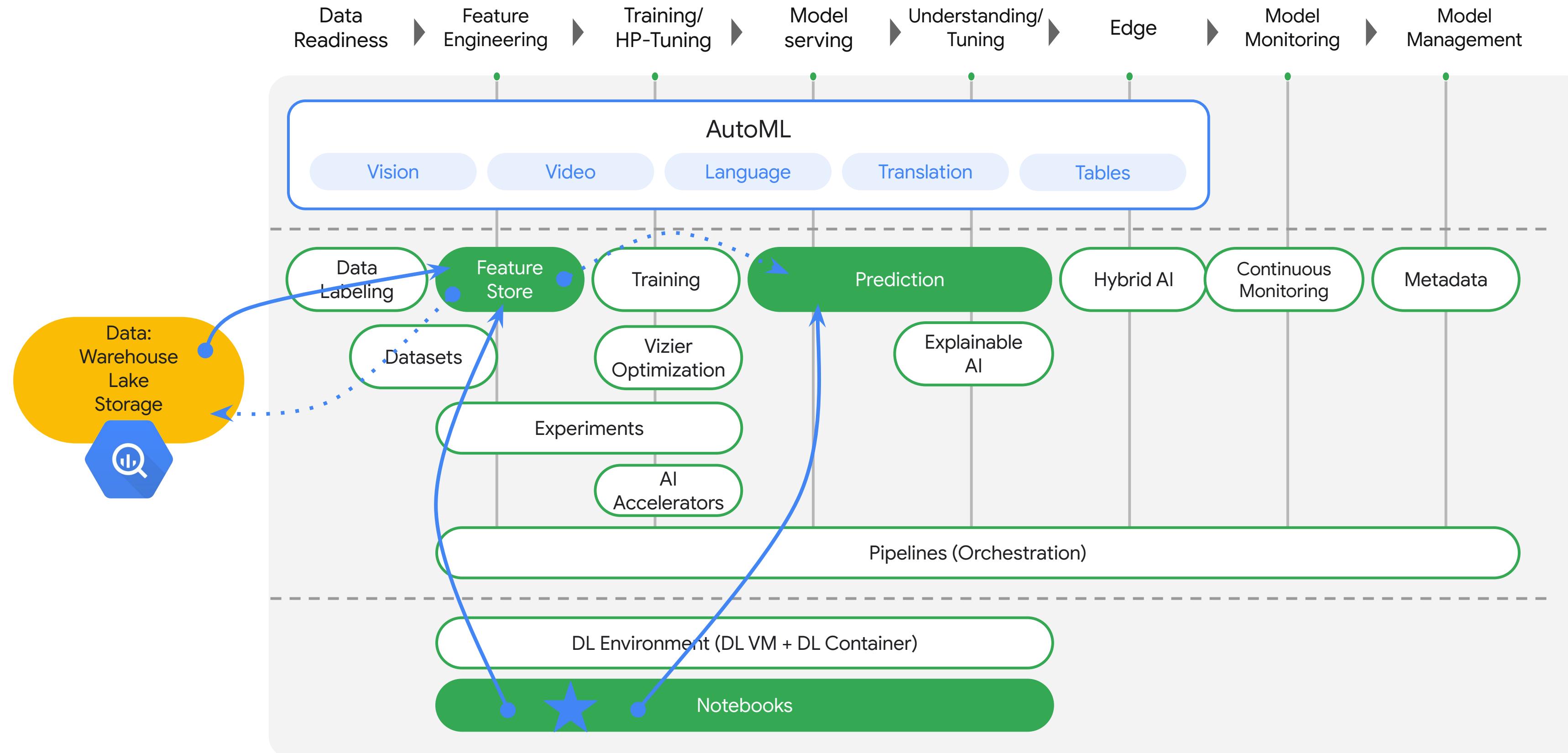


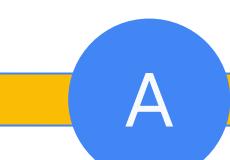
Vizier Optimization Service

11

Notebook: 11

Vertex AI Overview





11 - Vertex AI > Features - Feature Store

This is a demonstration of [Vertex AI Feature Store](#). A feature store is a central repository for organizing, storing, and retrieving features. This is a fully managed service that scales at points-in-time:

- Point-in-time lookups for retrieving features for model training. Retrieve feature values prior to a prediction to prevent data leakage.
- Manage training-serving skew

Prerequisites:

- 01 - BigQuery - Table Data Source
- Any of 02-05 That Deploy A Model To An Endpoint
 - Used to demonstrate online predictions with feature store serving features

Overview:

- Create a Feature Store
- Define an entity type

For this demonstration, use metadata from a BigQuery table to define features

Google Cloud Platform statmike-mlops Search products and resources

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

EDITOR DIGITS DIGITS_LR DIGITS_F...

COMPOSE NEW QUERY

Explorer + ADD DATA digits_fs_training

Type to search

Viewing pinned projects.

statmike-mlops digits Models (1) digits_lr digits digits_featurestore_import digits_prepended

SCHEMA DETAILS PREVIEW TABLE EXPLORER

Row	timestamp	entity_type_drawing	target	p0	p32	p16	p48	p8	p40	p24	p56	p4	p36	p20	p52	p12
1701	2021-09-14 21:04:52 UTC	a0c826c2-520d-46c1-aaef-461277eda3fa	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.0	16.0	2.0	6.0	11.0	
1702	2021-09-14 21:04:52 UTC	226aaaf8b-d8ff-43c7-89ca-b289f4fa7f12	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.0	16.0	0.0	1.0	8.0	
1703	2021-09-14 21:04:52 UTC	eefdf8f1f-3d1d-420c-a434-67cb01a6b8c0	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.0	16.0	0.0	8.0	10.0	
1704	2021-09-14 21:04:52 UTC	40e40f54-7f9d-4ad1-a68b-3e7723008894	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.0	16.0	0.0	8.0	4.0	
1705	2021-09-14 21:04:52 UTC	a7cd4041-fde9-4146-b1cc-299ea383545e	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.0	14.0	7.0	4.0	5.0	
1706	2021-09-14 21:04:52 UTC	c4e06400-bdad-4267-85e7-d1eb5d24d1f5	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.0	15.0	1.0	4.0	10.0	
1707	2021-09-14 21:04:52 UTC	cbf03781-98be-4fbe-9d5c-bc8d71fcf9a1	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.0	16.0	0.0	0.0	12.0	
1708	2021-09-14 21:04:52 UTC	f6aab929-cc5b-407d-87a4-10a503df4d0b	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.0	14.0	4.0	1.0	5.0	



Vertex AI



Feature Store