

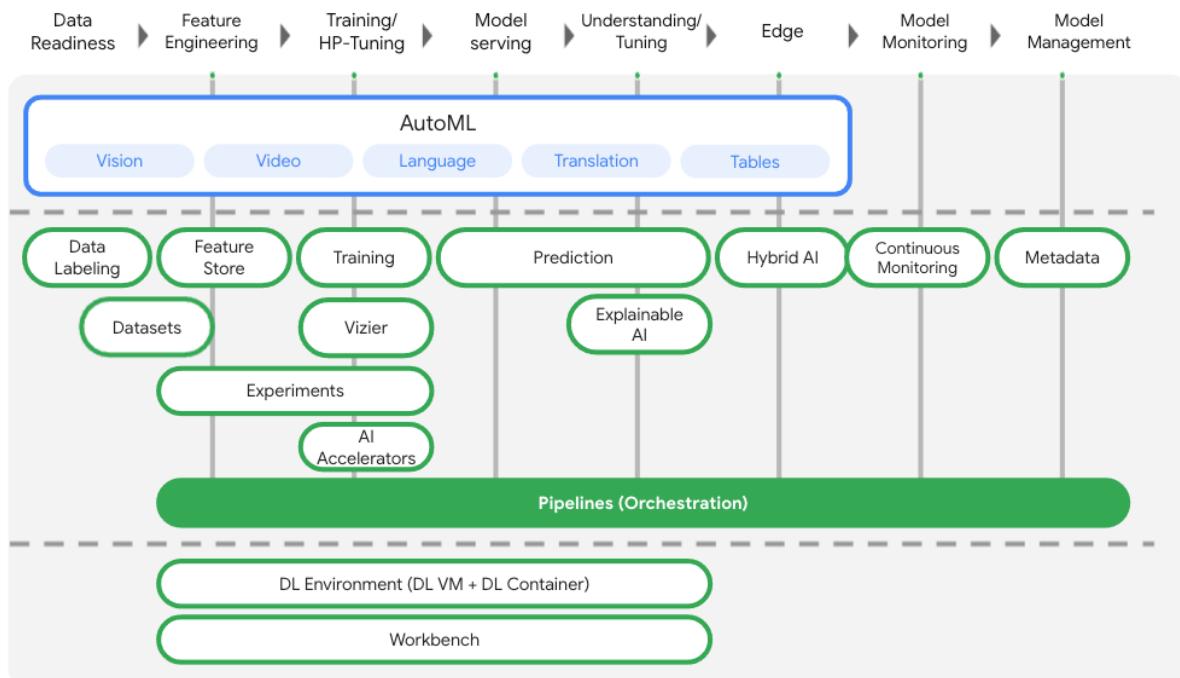
Assignment 7: optional catchup assignment 2 - VERTTEX AI - for midterm and quiz - this will catch up midterm.

a) Vertex AI pipeline Intro

Reference: <https://codelabs.developers.google.com/vertex-pipelines-intro#0>

Objectives:

- Use the Kubeflow Pipelines SDK to build scalable ML pipelines
- Create and run a 3-step intro pipeline that takes text input
- Create and run a pipeline that trains, evaluates, and deploys an AutoML classification model
- Use pre-built components for interacting with Vertex AI services, provided through the `google_cloud_pipeline_components` library
- Schedule a pipeline job with Cloud Scheduler



Cloud environment setup:

Enable API services and create google bucket:

```

CLOUD SHELL Terminal (cmpe260) + ~
$ gcloud services enable compute.googleapis.com
HTTPError: type:googleapis.com/billing-enabled
- 'gRPC': type.googleapis.com/google.rpc.ErrorInfo
domain: serviceusage.googleapis.com/billing-enabled
metadata:
  project: '114833773523'
  services: compute.googleapis.com,compute.googleapis.com,compute.googleapis.com,containerregistry.googleapis.com,aiplatform.googleapis.com,cloudbuild.googleapis.com
  reason: URER PROJECT BILLING NOT FOUND
cmpe260sp1@cloudshell:~ (cmpe260)$ gcloud services enable compute.googleapis.com
                                          containerregistry.googleapis.com
                                          aiplatform.googleapis.com
loudbuild.googleapis.com
^Z [1]+ Stopped                  gcloud services enable compute.googleapis.com containerregistry.googleapis.com aiplatform.googleapis.com cloudbuild.googleapis.com
cmpe260sp1@cloudshell:~ (cmpe260)$ gcloud services enable compute.googleapis.com \
                                     containerregistry.googleapis.com \
                                     aiplatform.googleapis.com \
                                     cloudbuild.googleapis.com \
                                     cloudfunctions.googleapis.com
Operation "operations/acf-p214833773523-4e4d5cbf-47a2-ad90-b1facdf6ac3" finished successfully.
cmpe260sp1@cloudshell:~ (cmpe260)$ gsutil mb -r us-central1 $BUCKET_NAME
cmpe260sp1@cloudshell:~ (cmpe260)$ gsutil mb -r us-central1 $BUCKET_NAME
Creating gs://cmpe260-bucket/...
cmpe260sp1@cloudshell:~ (cmpe260)$

```

Create a Vertex AI workbench instance:

The screenshot shows the Google Cloud Platform Vertex AI Workbench interface. On the left sidebar, 'Workbench' is selected. In the center, a 'New notebook' dialog is open, prompting for a 'Notebook name' (set to 'tensorflow-23-20211125-092022'), 'Region' (set to 'us-west1 (Oregon)'), and 'Zone' (set to 'us-west1-b'). Below the dialog, the 'Notebook properties' section is visible, showing environment details like 'TensorFlow Enterprise 2.3 (with LTS and Intel® MKL-DNN/MKL)', machine type ('4 vCPUs, 15 GB RAM'), and disk configurations. At the bottom right of the dialog is a 'CREATE' button.

Vertex pipeline setup

```

File Edit View Run Kernel Git Tabs Settings Help
+ Filter files by name
Name Last Modified
src 9 minutes ago
tutorials 9 minutes ago
Untitled.ipynb a minute ago
Untitled.ipynb Python 3
KFP SDK version: 1.8.9
google_cloud_pipeline_components version: 0.2.0
[2]: import os
PROJECT_ID = ""

# Get your Google Cloud project ID from gcloud
if not os.getenv("IS_TESTING"):
    shell_output = gcloud config list --format='value(core.project)' 2>/dev/null
    PROJECT_ID = shell_output[0]
    print(f"Project ID: {PROJECT_ID}")

Project ID: cmpe260

[3]: if PROJECT_ID == "" or PROJECT_ID is None:
    PROJECT_ID = "cmpe260" # @param {type:"string"}

[5]: BUCKET_NAME="gs://" + PROJECT_ID + "-bucket"

Import Libraries

[6]: import kfp

from kfp.v2 import compiler, dsl
from kfp.v2.dsl import component, pipeline, Artifact, ClassificationMetrics, Input, Output, Model, Metrics

from google.cloud import aiplatform
from google.cloud.pipeline_components import aiplatform as gcc_aip
from typing import NamedTuple

[7]: PATH=os.path.join(os.path.dirname(os.path.abspath(__file__)), 'pipeline_root')
REGION="us-central1"
PIPELINE_ROOT = f'{BUCKET_NAME}/pipeline_root/'

env: PATH=/usr/local/cuda/bin:/opt/conda/bin:/opt/conda/condabin:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/home/jupyter/.local/bin
[7]: 'gs://cmpe260-bucket/pipeline_root/'

[ ]:

```

```

File Edit View Run Kernel Git Tabs Settings Help
+ Filter files by name
Name Last Modified
src 18 minutes ago
tutorials 18 minutes ago
first-component.yaml 5 minutes ago
intro_pipeline_job.json 2 minutes ago
Untitled.ipynb 2 minutes ago
Untitled.ipynb Python 3
name="hello-world",
description="An intro pipeline",
pipeline_root=PIPELINE_ROOT,
)

# You can change the 'text' and 'emoji_str' parameters here to update the pipeline output
def intro_pipeline(text: str = "Vertex Pipelines", emoji_str: str = "sparkles"):
    product_task = product_name(text)
    emoji_task = emoji(emoji_str)
    consumer_task = build_sentence(
        product_task.outputs["text"],
        emoji_task.outputs["emoji"],
        emoji_task.outputs["emoji_text"],
    )

[13]: compiler.Compiler().compile(
    pipeline_func=intro_pipeline, package_path="intro_pipeline_job.json"
)

/home/jupyter/.local/lib/python3.7/site-packages/kfp/v2/compiler/compiler.py:1266: FutureWarning: APIs imported from the v1 namespace (e.g. kfp.dsl, kf
p.components, etc) will not be supported by the v2 compiler since v2.0.0
    category=FutureWarning,
[14]: from datetime import datetime
TIMESTAMP = datetime.now().strftime("%Y%m%d%H%M%S")
[15]: job = aiplatform.PipelineJob(
    display_name="hello-world-pipeline",
    template_path="intro_pipeline_job.json",
    job_id="hello-world-pipeline-{0}".format(TIMESTAMP),
    enable_caching=True
)
[16]: job.submit()
INFO:google.cloud.aiplatform.pipeline_jobs:Creating PipelineJob
INFO:google.cloud.aiplatform.pipeline_jobs:PipelineJob created. Resource name: projects/714833773523/locations/us-central1/pipelineJobs/hello-world-pip
eline-202111251738087
INFO:google.cloud.aiplatform.pipeline_jobs:To use this PipelineJob in another session:
INFO:google.cloud.aiplatform.pipeline_jobs:pipeline_job = aiplatform.PipelineJob.get('projects/714833773523/locations/us-central1/pipelineJobs/hello-w
rld-pipeline-202111251738087')
INFO:google.cloud.aiplatform.pipeline_jobs:View Pipeline Job:
https://console.cloud.google.com/vertex-ai/locations/us-central1/pipelines/runs/hello-world-pipeline-20211125173807?project=714833773523
[ ]:

```

<https://console.cloud.google.com/vertex-ai/locations/us-central1/pipelines/runs/hello-world-pipeline-20211125173807?project=714833773523>

The screenshot shows the Google Cloud Platform Vertex AI Pipelines interface. On the left, the sidebar navigation includes: Vertex AI, Dashboard, Datasets, Features, Labeling tasks, Workbench, Pipelines (selected), Training, Experiments, Models, Endpoints, Batch predictions, and Metadata. The main area displays a runtime graph for the pipeline run "hello-world-pipeline-20211125173807". The graph consists of three components: "emoji" (python:3.7), "product-name" (python:3.9), and "build-sentence" (python:3.7). Arrows indicate dependencies from "emoji" and "product-name" to "build-sentence". The "build-sentence" component is highlighted with a blue border. At the top of the graph area, there are buttons for CLONE, STOP, and DELETE. To the right of the graph, a "Pipeline run analysis" panel is open, showing the "SUMMARY" tab. The summary details the pipeline run with the following information:

Display name	build-sentence
Name	build-sentence
Type	container
Duration	0 sec
Started	
Completed	

Below the summary, the "NODE INFO" tab is selected, showing "Execution Info" with a status of "Pending". The "Input Parameters" section lists:

Parameter	Type	Value
emoji	string	<undefined_till_runtime>
product	string	<undefined_till_runtime>
emojitext	string	<undefined_till_runtime>

The "Output Parameters" section lists:

Parameter	Type	Value
Output	string	<undefined_till_runtime>

At the bottom of the page, there are links for Marketplace and Logs.

Creating an end to end ML pipeline

The screenshot shows a Google Colab interface with a Jupyter Notebook titled "First_Pipeline.ipynb". The notebook contains Python code for building and running a Vertex AI pipeline. The code includes defining a project configuration, setting up a ModelDeployOp, and defining a pipeline job. The output shows the pipeline being compiled and submitted to Vertex AI.

```
project=project,
location=gcp_region,
display_name="train-automl-beans",
)

gcc_api.ModelDeployOp(
    model=training_op.outputs["model"],
    endpoint=endpoint.outputs["endpoint"],
    dedicated_resources_min_replica_count=1,
    dedicated_resources_max_replica_count=1,
    dedicated_resources_machine_type="n1-standard-4",
)

Compling and running the end to end pipeline

[20]: compiler.Compiler().compile(
    pipeline_func=pipeline, package_path="tab_classif_pipeline.json"
)

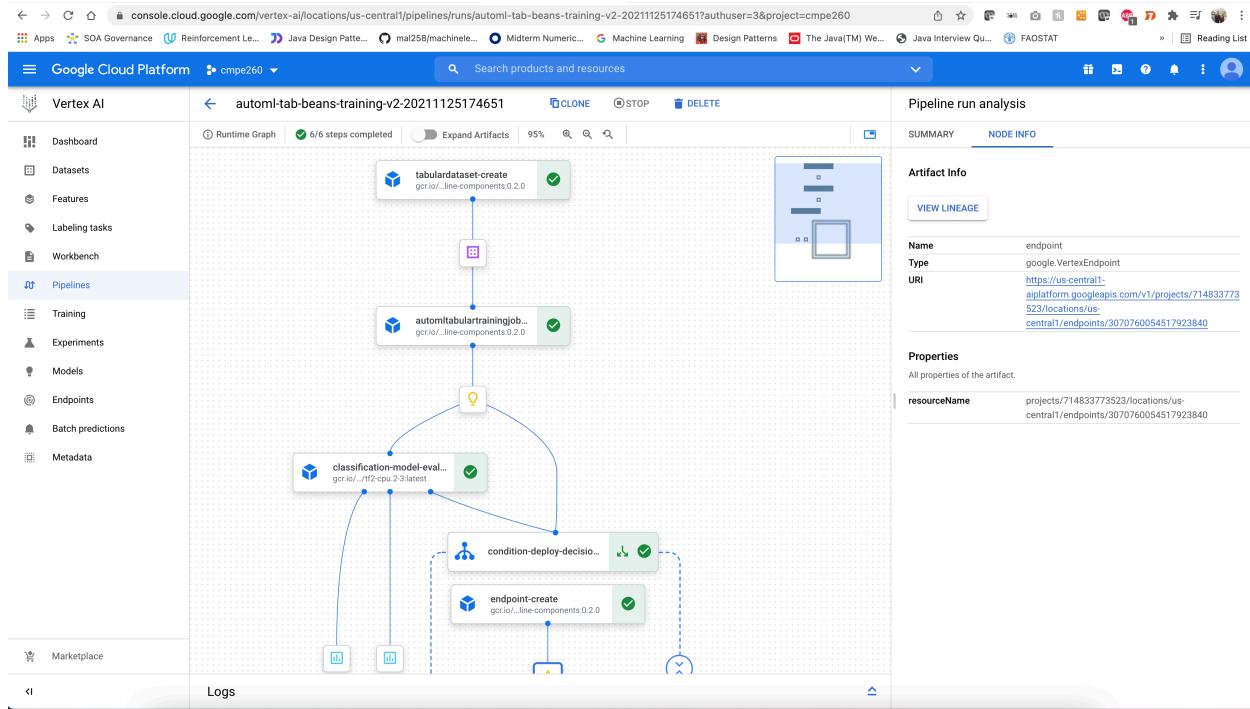
Defining the job

[21]: ml_pipeline_job = aiplatform.PipelineJob(
    display_name="automl-tab-beans-training",
    template_path="tab_classif_pipeline.json",
    pipeline_root=PIPELINE_ROOT,
    parameter_values={"project": PROJECT_ID, "display_name": DISPLAY_NAME},
    enable_caching=True
)

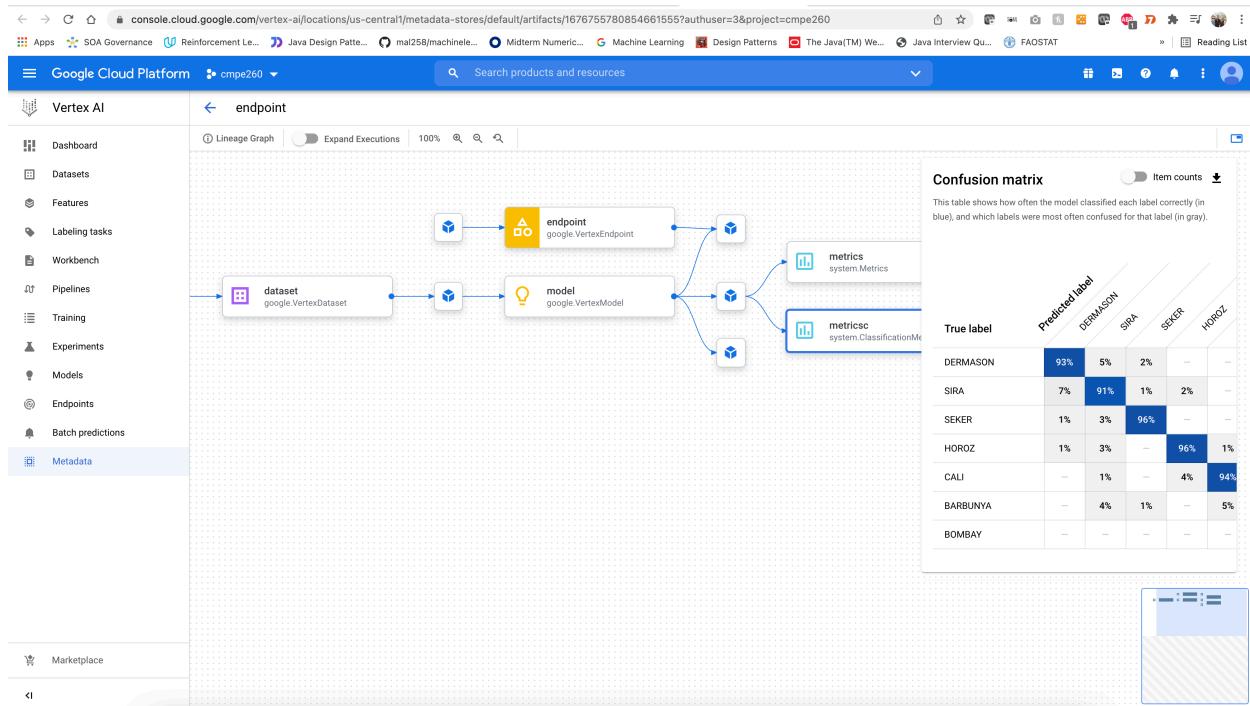
[22]: ml_pipeline_job.submit()

INFO:google.cloud.aiplatform.pipeline_jobs:Creating PipelineJob
INFO:google.cloud.aiplatform.pipeline_jobs:PipelineJob created. Resource name: projects/714833773523/locations/us-central1/pipelineJobs/automl-tab-bean
s-training-v2-2021125174651
INFO:google.cloud.aiplatform.pipeline_jobs:To use this PipelineJob in another session:
INFO:google.cloud.aiplatform.pipeline_jobs:jobs:pipeline_job = aiplatform.PipelineJob.get('projects/714833773523/locations/us-central1/pipelineJobs/automl-t
ab-beans-training-v2-2021125174651')
INFO:google.cloud.aiplatform.pipeline_jobs:View Pipeline Job:
https://console.cloud.google.com/vertex-ai/locations/us-central1/pipelines/runs/automl-tab-beans-training-v2-2021125174651?project=714833773523
```

Compiling and running the pipeline



Data Set and End Point View



The screenshot shows the Google Cloud Platform interface for Vertex AI. The left sidebar is collapsed, and the main content area is titled "Endpoints". A sub-header "CREATE ENDPOINT" is visible. Below this, a note explains what endpoints are and how they work. A dropdown menu for "Region" is open, showing "us-central1 (Iowa)" as the selected option. A table lists one endpoint entry:

Name	ID	Status	Models	Region	Monitoring	Most recent alerts	Last updated	API	Notification	Labels
train-automl-beans	3070760054517923840	Active	1	us-central1	Disabled	—	Nov 25, 2021, 12:19:55 PM	Sample request		?

At the bottom of the page, there are links for "Batch predictions" and "Metadata".

Model Output

console.cloud.google.com/vertex-ai/locations/us-central1/models/3476009254190579712/evaluate?authuser=3&project=cmpe260

Google Cloud Platform ccmpe260 Search products and resources

Vertex AI automl-beans1637862305 VIEW DATASET EXPORT

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

EVALUATE DEPLOY & TEST BATCH PREDICTIONS MODEL PROPERTIES

Filter Confidence threshold 0.5

All labels

	BOMBAY	SEKER	DERMAZON	CALI	BARBUNYA	HOROZ	SIRA
PR AUC	0.98	0.994	0.99528	0.97961	0.97727	0.97720	0.94841
ROC AUC	0.994	0.994	0.98905	0.97961	0.97727	0.97720	0.94841
Log loss	0.2	0.2	0.98905	0.97961	0.97727	0.97720	0.94841
F1 score	0.9304574	0.9304574	0.9304574	0.9304574	0.9304574	0.9304574	0.9304574
Precision	93.5%	93.5%	93.5%	93.5%	93.5%	93.5%	93.5%
Recall	92.6%	92.6%	92.6%	92.6%	92.6%	92.6%	92.6%
Created	Nov 25, 2021, 12:13:21 PM						

To evaluate your model, set the confidence threshold to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some [example scenarios](#) to learn how evaluation metrics can be used.

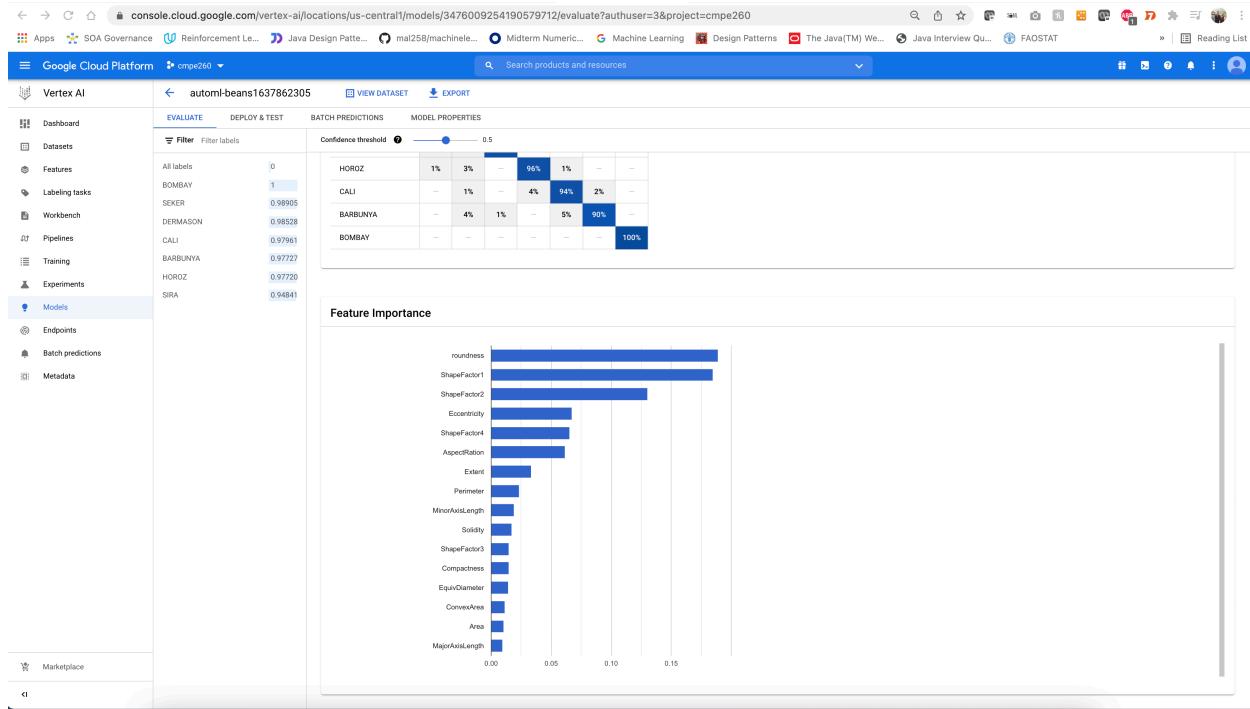
Precision-recall curve ROC curve Precision-recall by threshold

Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray).

True label	Predicted label						
	DERMAZON	SIRA	SEKER	HOROZ	CALI	BARBUNYA	BOMBAY
DERMAZON	93%	5%	2%	—	—	—	—
SIRA	7%	91%	1%	2%	—	0%	—
SEKER	1%	3%	95%	—	—	—	—
HOROZ	1%	3%	—	96%	1%	—	—
CALI	—	1%	—	4%	94%	2%	—
BARBUNYA	—	4%	1%	—	5%	90%	—
BOMBAY	—	—	—	—	—	100%	—

Marketplace



The screenshot shows a Jupyter Notebook interface with a Python 3 kernel. The notebook file is named 'First_Pipeline.ipynb'. The code in the notebook includes:

```

gcc_api.ModelDeploymentOp(
    model=training_op.outputs["model"],
    endpoint=endpoint_op.outputs["endpoint"],
    dedicated_resources_min_replica_count=1,
    dedicated_resources_max_replica_count=1,
    dedicated_resources_machine_type="n1-standard-4",
)

```

Compiling and running the end to end pipeline

```

[20]: compiler.Compiler().compile(
    pipeline_func=pipeline,
    package_path="tab_classif_pipeline.json"
)

```

Defining the job

```

[21]: ml_pipeline_job = aiplatform.PipelineJob(
    display_name="automl-tab-beans-training",
    template_path="tab_classif_pipeline.json",
    pipeline_root=PIPELINE_ROOT,
    parameter_values={"project": PROJECT_ID, "display_name": DISPLAY_NAME},
    enable_caching=True
)

```

```

[22]: ml_pipeline_job.submit()
INFO:google.cloud.aiplatform.pipeline_jobs:Creating PipelineJob
INFO:google.cloud.aiplatform.pipeline_jobs:PipelineJob created. Resource name: projects/714833773523/locations/us-central1/pipelineJobs/automl-tab-bean
s-training-v2-20211125174651
INFO:google.cloud.aiplatform.pipeline_jobs:To use this PipelineJob in another session:
INFO:google.cloud.aiplatform.pipeline_jobs:pipeline_job = aiplatform.PipelineJob.get("projects/714833773523/locations/us-central1/pipelineJobs/automl-
tab-beans-training-v2-20211125174651")
INFO:google.cloud.aiplatform.pipeline_jobs:View Pipeline Job:
https://console.cloud.google.com/vertex-ai/locations/us-central1/pipelines/runs/automl-tab-beans-training-v2-20211125174651?project=714833773523

```

```

[23]: pipeline_df = aiplatform.get_pipeline_df(pipeline="automl-tab-beans-training-v2")
small_pipeline_df = pipeline_df.head(2)
small_pipeline_df

```

pipeline_name	run_name	param.input:display_name	param.input:thresholds_dict_str	param.input:bq_source	param.input:gcp_region	param.input:api_endpoint	param.input:pro
automl-tab-beans-training-v2	automl-beans-training-v2-20211125174651	automl-beans1637862305	{"auRoc": 0.95}	bq://ju-dev-demos.beans1	us-central1	aiplatform.googleapis.com	cmpe