USC Viterbi
School of Engineering

ISE-543

MODULE 7 HOMEWORK

Bruce Wilcox, PhD

OVERVIEW

For this assignment you are going to be building custom Vertex AI training and prediction pipelines to predict the likelihood of patients developing breast cancer.

You will use the file breast_cancer_train.csv as the input to your training pipeline and breast_cancer_test.csv as the input to your prediction pipeline.

You are to do each of these steps as described in the following pages, paste screenshots of your results, and paste the screenshots back into this file for upload to Gradescope (as a PDF) when you are completed

OVERVIEW

Training Pipeline

For this assignment you are going to be designing and building custom Vertex AI training and prediction pipelines that performs the following steps:

- Loads the dataset into the pipeline via a pipeline parameter
- Splits the dataset into training and validation partitions
- Normalizes the dataset
- Trains a Logistic regression model
- Assesses the performance of this model using the validation partition by calculating the model's accuracy and F1 score.

Then, you will compile the Vertex AI pipeline and run the pipeline.

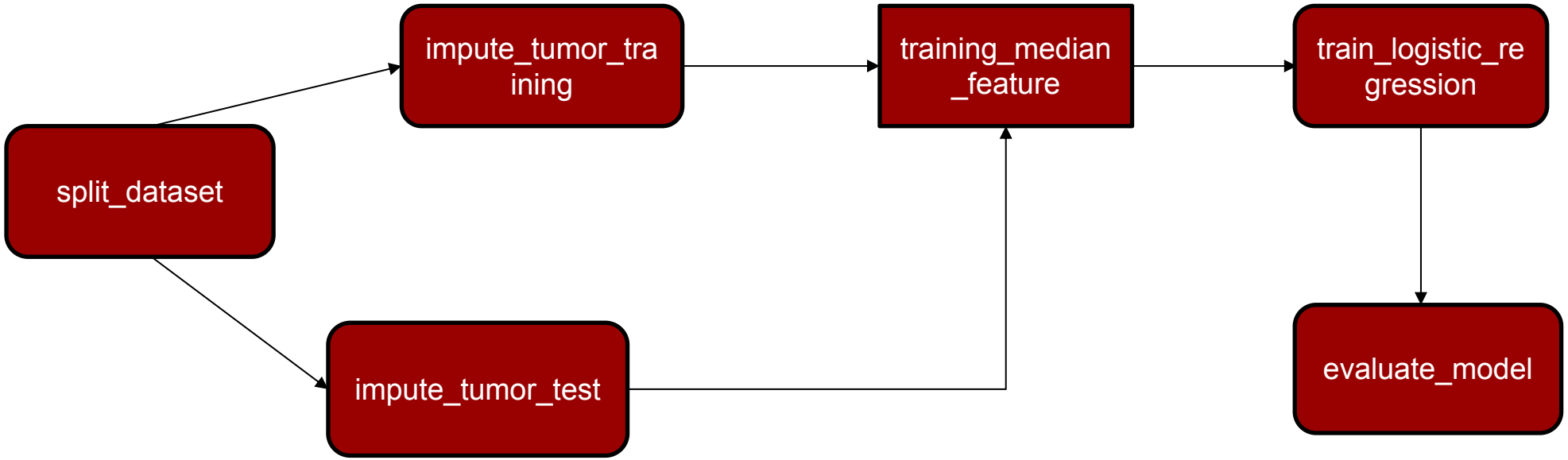
You are to do each of these steps as described in the following pages, paste screenshots of your results, and paste the screenshots back into this file for upload to Gradescope (as a PDF) when you are completed

USC Viterbi School of Engineering

STEP 1

Pipeline Design – Directed Acyclic Graph

Using the PowerPoint drawing tool, draw the directed acyclic graph (DAG) for your pipeline design. Be sure to label every arc. Be careful to avoid model leakage in your pipeline design



USC Viterbi School of Engineering

Step 2

Pipeline Input-Output Matrix

Create a Pipeline Input-Output Matrix for your pipeline design below

Component	Input Name	Type	Output Name	Type
split_dataset	`dataset_path`	string	`training_dataset`	Dataset
split_dataset	`dataset_path`	string	`test_dataset`	Dataset
impute_tumor_training`	`training_dataset`	Dataset	`imputed_training_dataset`	Dataset
impute_tumor_training`	`training_dataset`	Dataset	`training_median_feature`	Dataset
impute_tumor_test	`test_dataset`	Dataset	`imputed_test_dataset`	Dataset
impute_tumor_test	`training_median_feature`	Dataset	`imputed_test_dataset`	Dataset
train_logistic_regression	`imputed_training_dataset`	Dataset	`trained_model`	Model
evaluate_model	`imputed_test_dataset`	Dataset	`eval_metrics`	Metrics
evaluate_model	`trained_model`	Model	`eval_metrics`	Metrics

Step 3

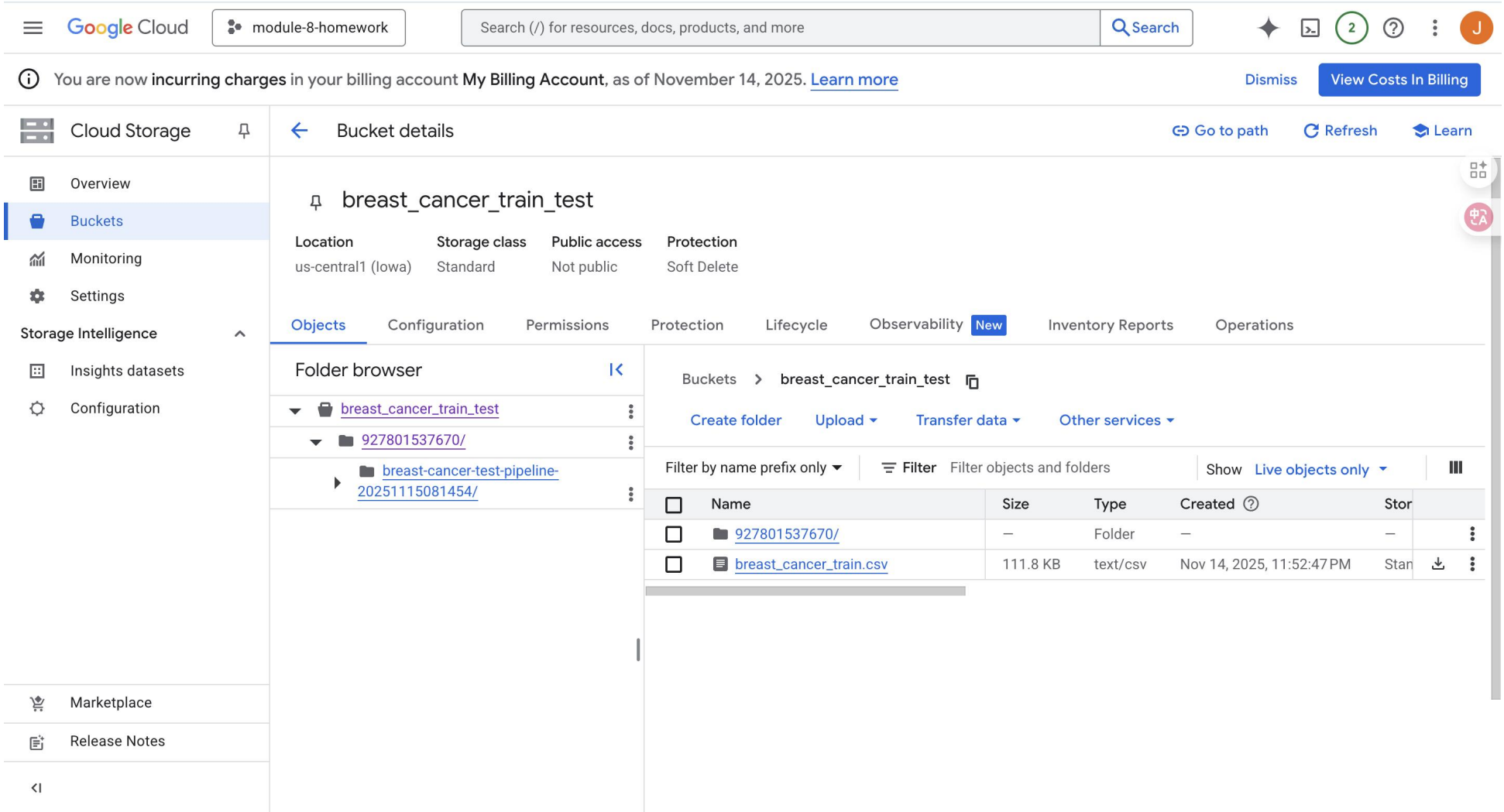
Load training and test data into GCS

- Create a new bucket in GCS for this homework assignment
- Load your training and validation datasets into this bucket
- On the following page, paste a screenshot showing your bucket with the training dataset

USC Viterbi School of Engineering

Step 3

Load training and test data into GCS



STEP 4

Environment Setup

Using Colab Enterprise in Vertex AI, perform the following steps

- Configure your project_id and location
- Load and initialize required libraries
- Create a Python string variable that holds the paths to your training data

On the following page, paste a screenshot of your Colab code

USC Viterbi School of Engineering

STEP 4

Environment Setup

[4] ✓ 0s

```
# Step 1: Set your project configuration
project_id = "module-8-homework-478307"
location = "us-central1"
bucket_name = "breast_cancer_train_test"
pipeline_root = f"gs://{bucket_name}/pipeline_root"

print("Project ID:", project_id)
print("Location:", location)
print("Pipeline Root:", pipeline_root)
```

```
Project ID: module-8-homework-478307
Location: us-central1
Pipeline Root: gs://breast_cancer_train_test/pipeline_root
```

[3] ✓ 0s

```
# Step 2: Import and initialize libraries
from google.cloud import aiplatform

# Initialize Vertex AI
aiplatform.init(project=project_id, location=location)
print("Vertex AI initialized successfully.")
```

```
Vertex AI initialized successfully.
```

[5] ✓ 0s

```
# Step 3: Create a variable for your training data path
training_data_path = "gs://breast_cancer_train_test/breast_cancer_train.csv"

print("Training data path:", training_data_path)
```

```
Training data path: gs://breast_cancer_train_test/breast_cancer_train.csv
```

STEP 5

Configure and Test Pipeline Setup

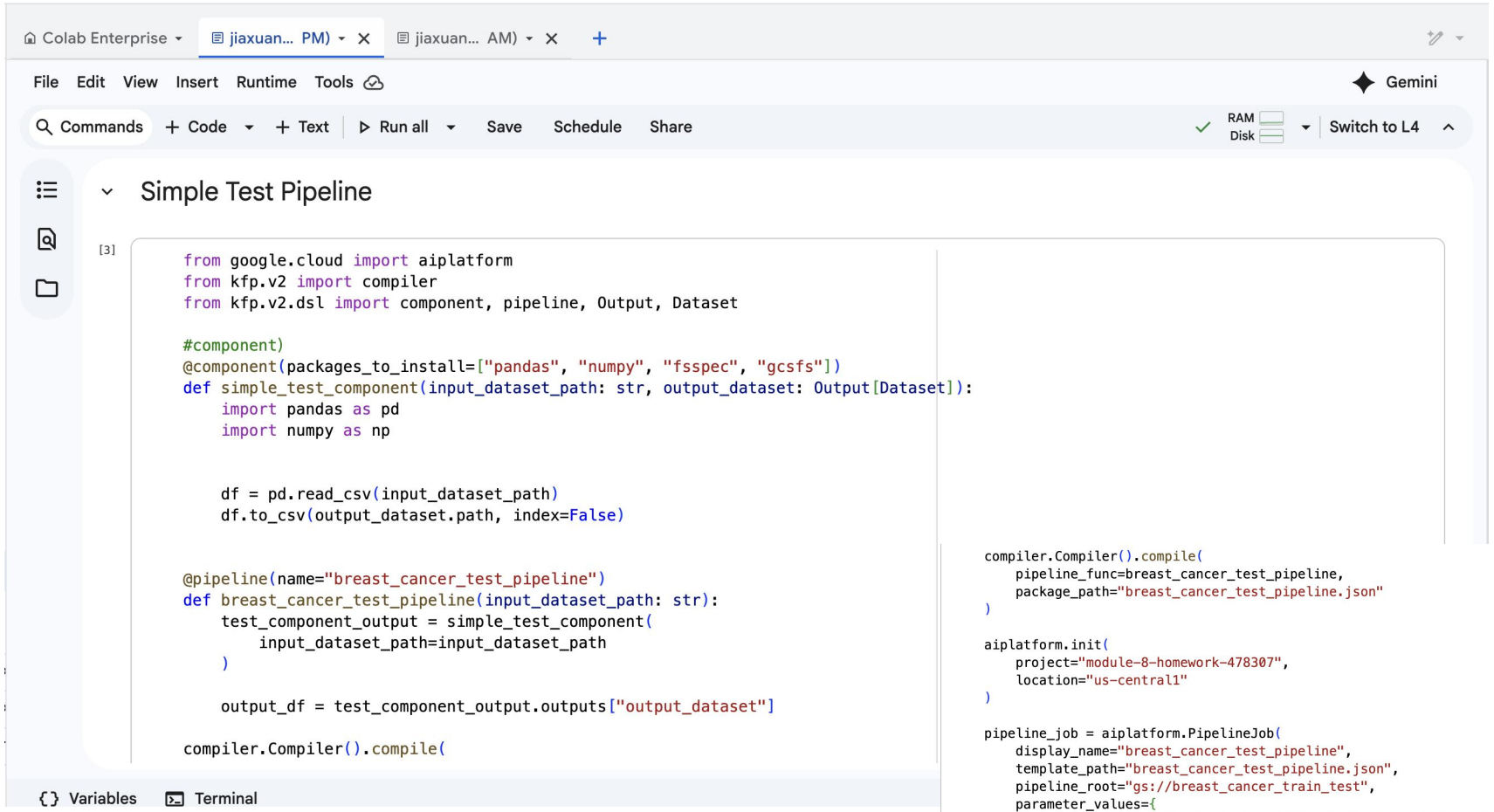
In this step, create a simple test pipeline to verify correct overall pipeline setup:

- Define a pipeline function and corresponding pipeline job that has the same inputs and outputs that your project will have
  - In this case, there will be one input (the path to the training dataset)
- Make the pipeline function code call one simple component. Pass it any required inputs for our and have it perform the minimal amount of processing to verify its operation
  - In this case, we will read in the dataset and then save it to an artifact as a csv file
- Review the lecture PowerPoint for an example of how to set this up ("Validate Pipeline Setup")
- On the following pages, paste the indicated screenshots to demonstrate correct setup

USC Viterbi School of Engineering

STEP 5

Colab Screenshot



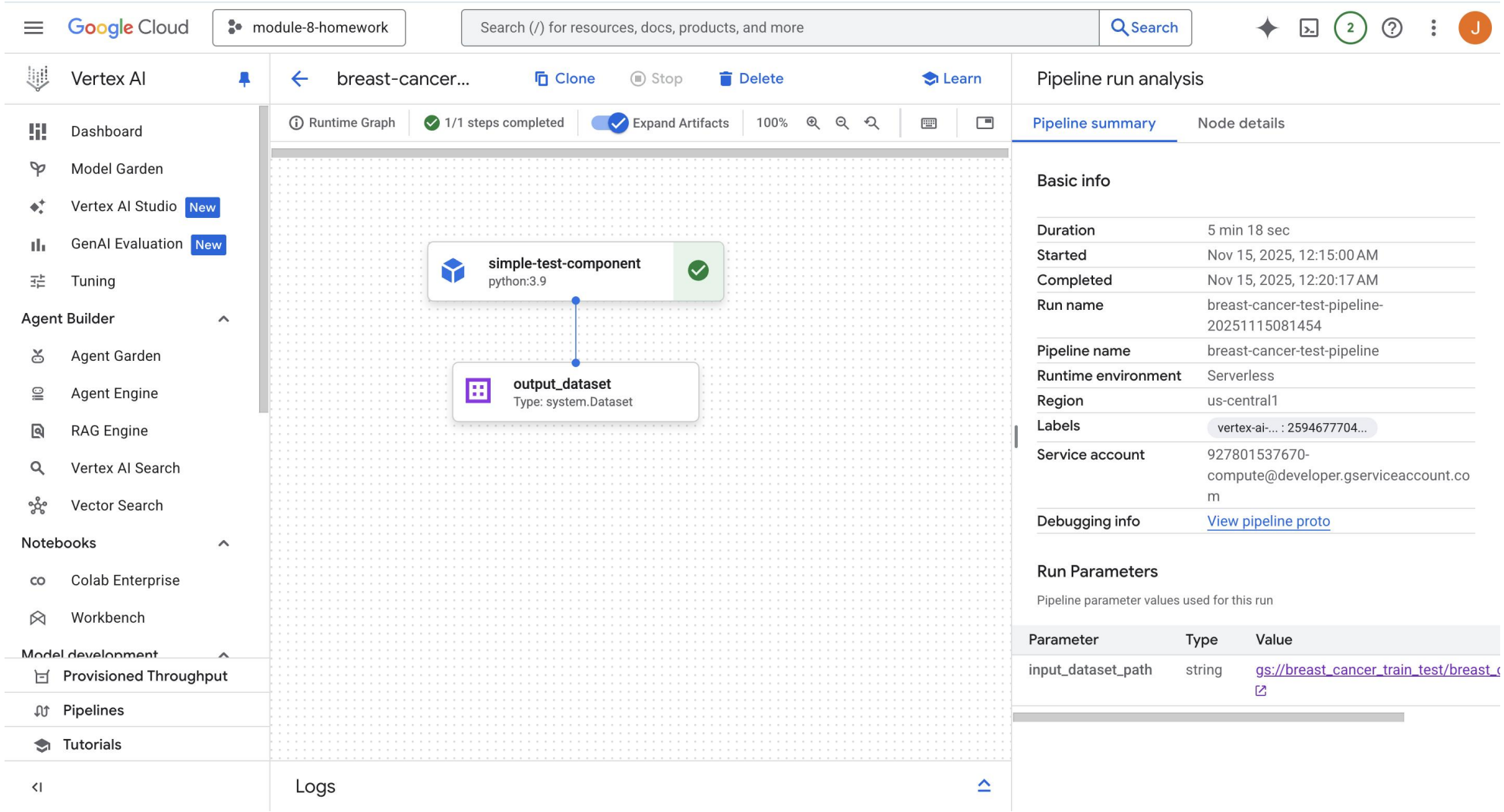

```
compiler.Compiler().compile(
    pipeline_func=breast_cancer_test_pipeline,
    package_path="breast_cancer_test_pipeline.json"
)

aiplatform.init(
    project="module-8-homework-478307",
    location="us-central1"
)

pipeline_job = aiplatform.PipelineJob(
    display_name="breast_cancer_test_pipeline",
    template_path="breast_cancer_test_pipeline.json",
    pipeline_root="gs://breast_cancer_train_test",
    parameter_values={
        "input_dataset_path": "gs://breast_cancer_train_test/breast_cancer_train.csv"
    }
)

pipeline_job.submit()
```

```
/usr/local/lib/python3.12/dist-packages/kfp/dsl/component_decorator.py:126: FutureWarning: The default base_image used by the @dsl.com
  return component_factory.create_component_from_func(
INFO:google.cloud.aiplatform.pipeline_jobs:Creating PipelineJob
```

USC Viterbi School of Engineering

STEP 5

Vertex Pipeline Graphic



USC Viterbi School of Engineering

STEP 5

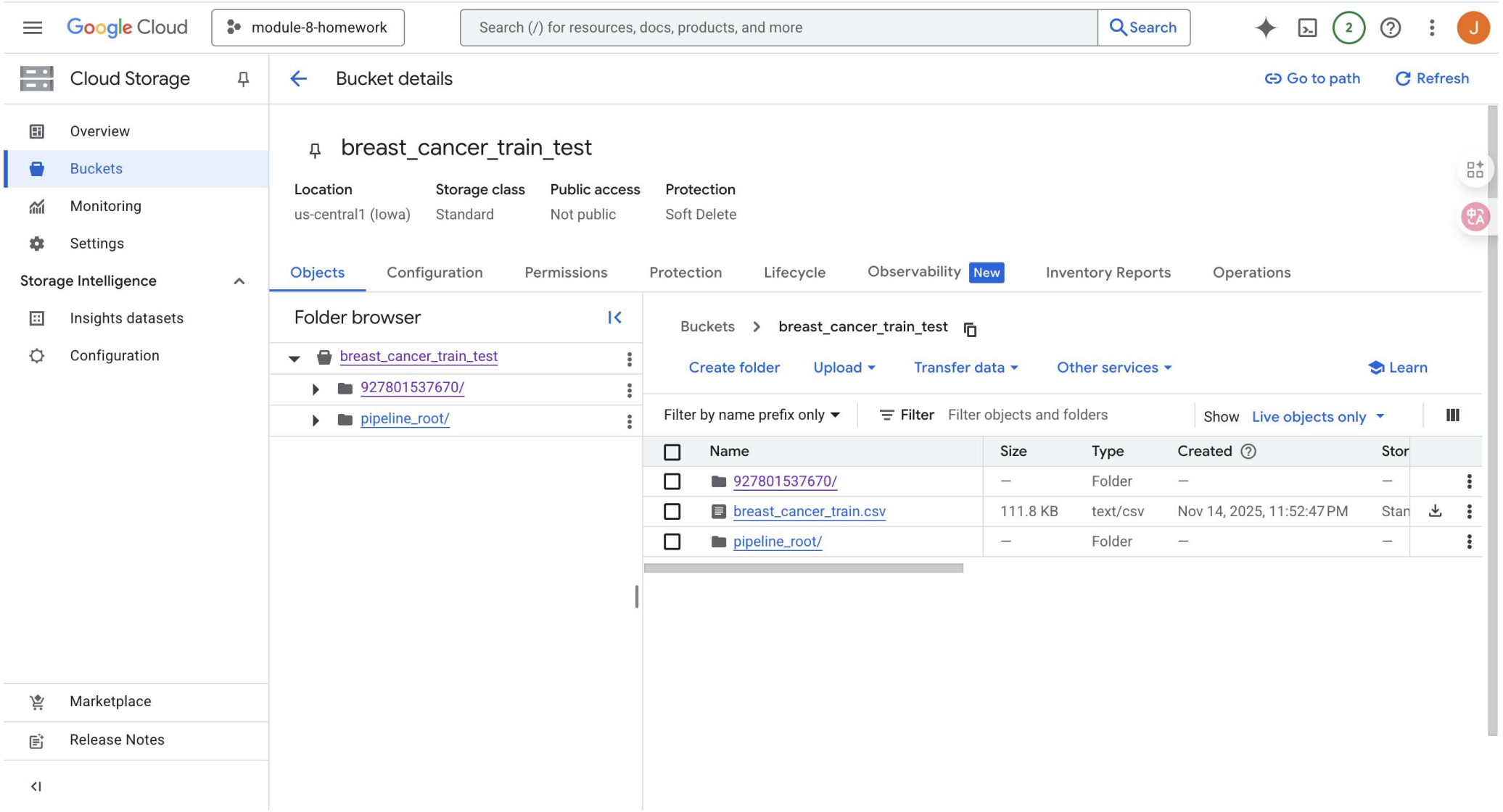
GCS Screenshot Showing the Output Dataset Path

```
patient id,mean radius,mean texture,mean perimeter,mean area,mean smoothness,mean compactness,mean concavity,mean concave points,mean symmetry,mean fractal dimension,radius
error,texture error,perimeter error,area error,smoothness error,compactness error,concavity error,concave points error,symmetry error,fractal dimension error,worst radius,worst
texture,worst perimeter,worst area,worst smoothness,worst compactness,worst concavity,worst concave points,worst symmetry,worst fractal dimension,target
8130402363,17.99,10.38,122.8,1001.0,0.1184,0.2776,0.3001,0.1471,0.2419,0.07871,1.095,0.9053,8.589,153.4,0.006399,0.04904,0.05373,0.01587,0.03003,0.006193,25.38,17.33,184.6,2019.0,0.
1622,0.6656,0.7119,0.2654,0.4601,0.1189,0
1732580124,20.57,17.77,132.9,1326.0,0.08474,0.07864,0.0869,0.07017,0.1812,0.05667,0.5435,0.7339,3.398,74.08,0.005225,0.01308,0.0186,0.0134,0.01389,0.003532,24.99,23.41,158.8,1956.0,
0.1238,0.1866,0.2416,0.186,0.275,0.08902,0
1439711531,19.69,21.25,130.0,1203.0,0.1096,0.1599,0.1974,0.1279,0.2069,0.05999,0.7456,0.7869,4.585,94.03,0.00615,0.04006,0.03832,0.02058,0.0225,0.004571,23.57,25.53,152.5,1709.0,0.1
444,0.4245,0.4504,0.243,0.3613,0.08758,0
4758381800,11.42,20.38,77.58,386.1,0.1425,0.2839,0.2414,0.1052,0.2597,0.09744,0.4956,1.156,3.445,27.23,0.00911,0.07458,0.05661,0.01867,0.05963,0.009208,14.91,26.5,98.87,567.7,0.2098
,0.8663,0.6869,0.2575,0.6638,0.173,0
6994024981,20.29,14.34,135.1,1297.0,0.1003,0.1328,0.198,0.1043,0.1809,0.05883,0.7572,0.7813,5.438,94.44,0.01149,0.02461,0.05688,0.01885,0.01756,0.005115,22.54,16.67,152.2,1575.0,0.1
374,0.205,0.4,0.1625,0.2364,0.07678,0
7075346032,12.45,15.7,82.57,477.1,0.1278,0.17,0.1578,0.08089,0.2087,0.07613,0.3345,0.8902,2.217,27.19,0.00751,0.03345,0.03672,0.01137,0.02165,0.005082,15.47,23.75,103.4,741.6,0.1791
,0.5249,0.5355,0.1741,0.3985,0.1244,0
2600402193,18.25,19.98,119.6,1040.0,0.09463,0.109,0.1127,0.074,0.1794,0.05742,0.4467,0.7732,3.18,53.91,0.004314,0.01382,0.02254,0.01039,0.01369,0.002179,22.88,27.66,153.2,1606.0,0.1
442,0.2576,0.3784,0.1932,0.3063,0.08368,0
1476064121,13.71,20.83,90.2,577.9,0.1189,0.1645,0.09366,0.05985,0.2196,0.07451,0.5835,1.377,3.856,50.96,0.008805,0.03029,0.02488,0.01448,0.01486,0.005412,17.06,28.14,110.6,897.0,0.1
654,0.3682,0.2678,0.1556,0.3196,0.1151,0
6778027432,13.0,21.82,87.5,519.8,0.1273,0.1932,0.1859,0.09353,0.235,0.07389,0.3063,1.002,2.406,24.32,0.005731,0.03502,0.03553,0.01226,0.02143,0.003749,15.49,30.73,106.2,739.3,0.1703
,0.5401,0.539,0.206,0.4378,0.1072,0
6004203110,12.46,24.04,83.97,475.9,0.1186,0.2396,0.2273,0.08543,0.203,0.08243,0.2976,1.599,2.039,23.94,0.007149,0.07217,0.07743,0.01432,0.01789,0.01008,15.09,40.68,97.65,711.4,0.185
3,1.058,1.105,0.221,0.4366,0.2075,0
9670227276,16.02,23.24,102.7,797.8,0.08206,0.06669,0.03299,0.03323,0.1528,0.05697,0.3795,1.187,2.466,40.51,0.004029,0.009269,0.01101,0.007591,0.0146,0.003042,19.19,33.88,123.8,1150.
0,0.1181,0.1551,0.1459,0.09975,0.2948,0.08452,0
2002618651,15.78,17.89,103.6,781.0,0.0971,0.1292,0.09954,0.06606,0.1842,0.06082,0.5058,0.9849,3.564,54.16,0.005771,0.04061,0.02791,0.01282,0.02008,0.004144,20.42,27.28,136.5,1299.0,
0.1396,0.5609,0.3965,0.181,0.3792,0.1048,0
5298473044,19.17,24.8,132.4,1123.0,0.0974,0.2458,0.2065,0.1118,0.2397,0.078,0.9555,3.568,11.07,116.2,0.003139,0.08297,0.0889,0.0409,0.04484,0.01284,20.96,29.94,151.7,1332.0,0.1037,0
.3903,0.3639,0.1767,0.3176,0.1023,0
1901123024,15.85,23.95,103.7,782.7,0.08401,0.1002,0.09938,0.05364,0.1847,0.05338,0.4033,1.078,2.903,36.58,0.009769,0.03126,0.05051,0.01992,0.02981,0.003002,16.84,27.66,112.0,876.5,0
.1131,0.1924,0.2322,0.1119,0.2809,0.06287,0
2716693257,13.73,22.61,93.6,578.3,0.1131,0.2293,0.2128,0.08025,0.2069,0.07682,0.2121,1.169,2.061,19.21,0.006429,0.05936,0.05501,0.01628,0.01961,0.008093,15.03,32.01,108.8,697.7,0.16
51,0.7725,0.6943,0.2208,0.3596,0.1431,0
5257560611,14.54,27.54,96.73,658.8,0.1139,0.1595,0.1639,0.07364,0.2303,0.07077,0.37,1.033,2.879,32.55,0.005607,0.0424,0.04741,0.0109,0.01857,0.005466,17.46,37.13,124.1,943.2,0.1678,
0.6577,0.7026,0.1712,0.4218,0.1341,0
5779890757,14.68,20.13,94.74,684.5,0.09867,0.072,0.07395,0.05259,0.1586,0.05922,0.4727,1.24,3.195,45.4,0.005718,0.01162,0.01998,0.01109,0.0141,0.002085,19.07,30.88,123.4,1138.0,0.14
64,0.1871,0.2914,0.1609,0.3029,0.08216,0
3231520341,16.13,20.68,108.1,798.8,0.117,0.2022,0.1722,0.1028,0.2164,0.07356,0.5692,1.073,3.854,54.18,0.007026,0.02501,0.03188,0.01297,0.01689,0.004142,20.96,31.48,136.8,1315.0,0.17
89,0.4233,0.4784,0.2073,0.3706,0.1142,0
5563282388,19.81,22.15,130.0,1260.0,0.09831,0.1027,0.1479,0.09498,0.1582,0.05395,0.7582,1.017,5.865,112.4,0.006494,0.01893,0.03391,0.01521,0.01356,0.001997,27.32,30.88,186.8,2398.0,
0.1512,0.315,0.5372,0.2388,0.2768,0.07615,0
5111781226,13.54,14.36,87.46,566.3,0.09779,0.08129,0.06664,0.04781,0.1885,0.05766,0.2699,0.7886,2.058,23.56,0.008462,0.0146,0.02387,0.01315,0.0198,0.0023,15.11,19.26,99.7,711.2,0.14
4,0.1773,0.239,0.1288,0.2977,0.07259,1
2076524640,13.08,15.71,85.63,520.0,0.1075,0.127,0.04568,0.0311,0.1967,0.06811,0.1852,0.7477,1.383,14.67,0.004097,0.01898,0.01698,0.00649,0.01678,0.002425,14.5,20.49,96.09,630.5,0.13
12,0.2776,0.189,0.07283,0.3184,0.08183,1
6663547230,9.504,12.44,60.34,273.9,0.1024,0.06492,0.02956,0.02076,0.1815,0.06905,0.2773,0.9768,1.909,15.7,0.009606,0.01432,0.01985,0.01421,0.02027,0.002968,10.23,15.66,65.13,314.9,0
.1324,0.1148,0.08867,0.06227,0.245,0.07773,1
8753397216,15.34,14.26,102.5,704.4,0.1073,0.2135,0.2077,0.09756,0.2521,0.07032,0.4388,0.7096,3.384,44.91,0.006789,0.05328,0.06446,0.02252,0.03672,0.004394,18.07,19.08,125.1,980.9,0.
139,0.5954,0.6305,0.2393,0.4667,0.09946,0
5901207698,21.16,23.04,137.2,1404.0,0.09428,0.1022,0.1097,0.08632,0.1769,0.05278,0.6917,1.127,4.303,93.99,0.004728,0.01259,0.01715,0.01038,0.01083,0.001987,29.17,35.59,188.0,2615.0,
0.1401,0.26,0.3155,0.2009,0.2822,0.07526,0
```

USC Viterbi School of Engineering

STEP 5

Screenshot of Bucket Contents



Step 6

Building the Components

Build each of the components in your DAG. On the following pages paste screenshots of each component in your pipeline.

- Create a copy of the following page for each component in your DAG
- Put the name of the component in the title block of each page

USC Viterbi School of Engineering

Step 6

Component Colab Screenshot: <Component Name>

impute_tumor_training

[111]

```python
@component(packages_to_install=["pandas"])
def impute_tumor_training(
    training_dataset: Input[Dataset],
    imputed_training_dataset: Output[Dataset],
    training_median_feature: Output[Dataset],
):
    import pandas as pd

    df = pd.read_csv(training_dataset.path)

    feature_col = "mean radius"
    median_val = df[feature_col].median()

    df[feature_col] = df[feature_col].fillna(median_val)

    df.to_csv(imputed_training_dataset.path, index=False)
    pd.DataFrame({"training_median_feature": [median_val]}).to_csv(
        training_median_feature.path, index=False
    )
```

```
/usr/local/lib/python3.12/dist-packages/kfp/dsl/component_decorator.py:126: FutureWarning: The default base_image used by the @dsl.com
  return component_factory.create_component_from_func(
```

Step 6

impute_tumor_test

[112]
✓ 0s

```python
@component(packages_to_install=["pandas"])
def impute_tumor_test(
    test_dataset: Input[Dataset],
    training_median_feature: Input[Dataset],
    imputed_test_dataset: Output[Dataset],
):
    import pandas as pd

    df = pd.read_csv(test_dataset.path)

    median_df = pd.read_csv(training_median_feature.path)
    median_val = float(median_df["training_median_feature"].iloc[0])

    feature_col = "mean radius"   # 你的 breast cancer 特征列
    df[feature_col] = df[feature_col].fillna(median_val)

    df.to_csv(imputed_test_dataset.path, index=False)
```

Step 6

train_logistic_regression

[113]

```python
@component(packages_to_install=["pandas", "scikit-learn", "joblib"])
def train_logistic_regression(
    imputed_training_dataset: Input[Dataset],
    trained_model: Output[Model],
):
    import pandas as pd
    from sklearn.linear_model import LogisticRegression
    import joblib

    df = pd.read_csv(imputed_training_dataset.path)

    label_col = "target"

    X_train = df.drop(label_col, axis=1)
    y_train = df[label_col]

    model = LogisticRegression(max_iter=1000)
    model.fit(X_train, y_train)

    joblib.dump(model, trained_model.path)
```

Step 6

evaluate_model

[129] ✓ 0s

```python
from kfp.dsl import component, Input, Output, Dataset, Model, Metrics

@component(
    packages_to_install=["pandas", "scikit-learn", "joblib"]
)
def evaluate_model(
    imputed_test_dataset: Input[Dataset],
    trained_model: Input[Model],
    eval_metrics: Output[Metrics],
):
    import pandas as pd
    import joblib
    from sklearn.metrics import accuracy_score, f1_score

    # Load test dataset
    df = pd.read_csv(imputed_test_dataset.path)

    # Split features and target
    X_test = df.drop("target", axis=1)
    y_test = df["target"]

    # Load trained model
    model = joblib.load(trained_model.path)

    # Predict
    preds = model.predict(X_test)

    # Compute metrics
    accuracy = accuracy_score(y_test, preds)
    f1 = f1_score(y_test, preds)

    # Log metrics
    eval_metrics.log_metric("accuracy", float(accuracy))
    eval_metrics.log_metric("f1_score", float(f1))
```

Step 7

Create the Pipeline

Write the Colab code to implement your pipeline. On the following page, paste a screenshot showing your pipeline code

[115]

```
from kfp.v2.dsl import pipeline

@pipeline(name="breast-cancer-training-pipeline")
def breast_cancer_pipeline(breast_cancer_dataset_path: str):

    # Step 1: split raw dataset
    split_task = split_dataset(
        dataset_path=breast_cancer_dataset_path
    )

    # Step 2: impute training set
    impute_train_task = impute_tumor_training(
        training_dataset=split_task.outputs["training_dataset"]
    )

    # Step 3: impute test set using training median
    impute_test_task = impute_tumor_test(
        test_dataset=split_task.outputs["test_dataset"],
        training_median_feature=impute_train_task.outputs["training_median_feature"],
    )

    # Step 4: train logistic regression model
    train_task = train_logistic_regression(
        imputed_training_dataset=impute_train_task.outputs["imputed_training_dataset"]
    )
```

Step 7

Create the Pipeline – Colab Pipeline Code

```
# Step 3: impute test set using training median
impute_test_task = impute_tumo●_test(
    test_dataset=split_task.outputs["test_dataset"],
    training_median_feature=impute_train_task.outputs["training_median_feature"],
)

# Step 4: train logistic regression model
train_task = train_logistic_regression(
    imputed_training_dataset=impute_train_task.outputs["imputed_training_dataset"]
)

# Step 5: evaluate on test set
evaluate_model(
    imputed_test_dataset=impute_test_task.outputs["imputed_test_dataset"],
    trained_model=train_task.outputs["trained_model"],
)
```

Step 8

Compile and Run Pipeline

Create the Colab code to compile your pipeline and submit the pipeline for execution. On the following pages, paste screenshots showing:

- Your Colab code
- The Vertex AI pipeline graphic showing its successful execution

Step 8

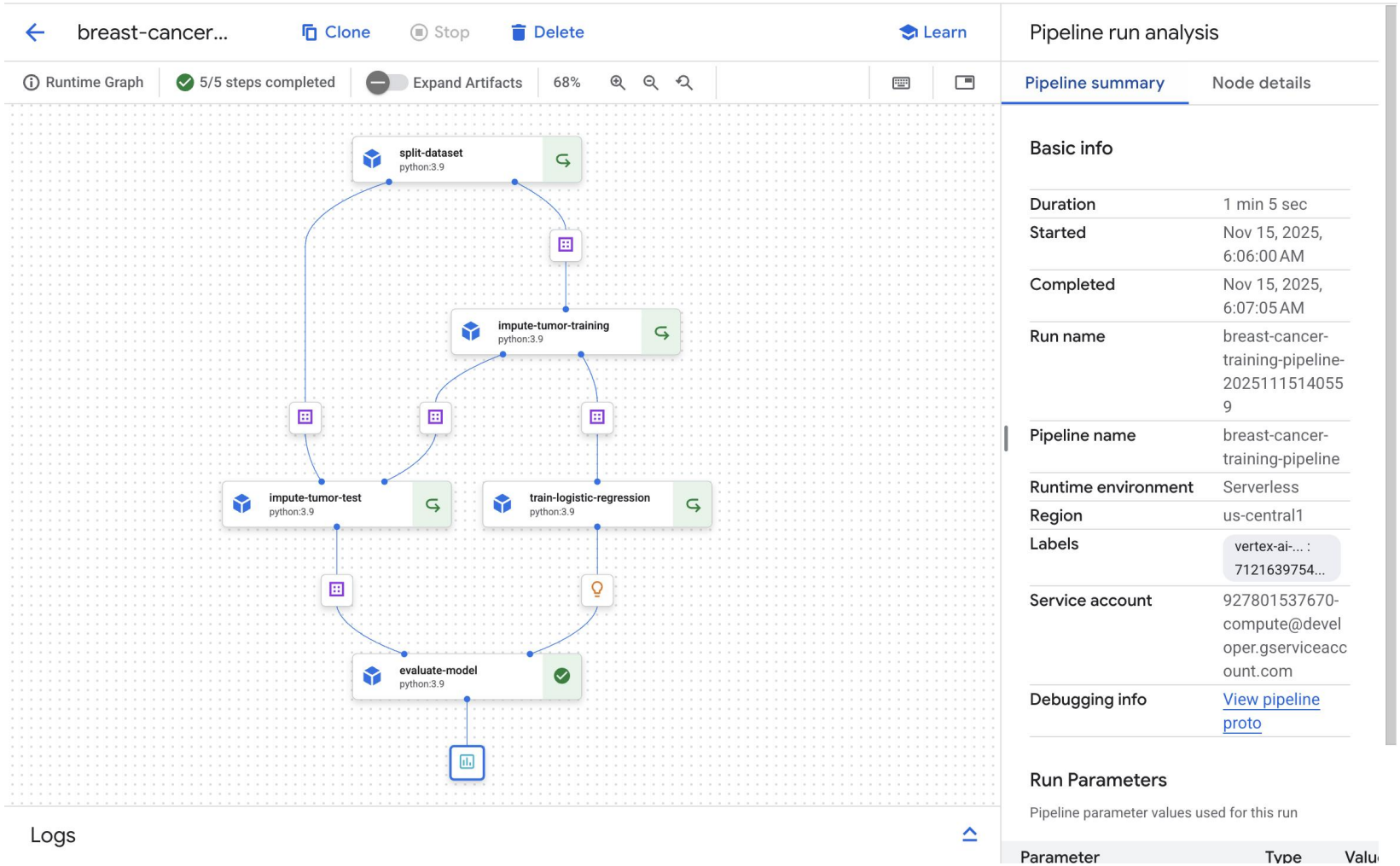
Colab Screenshot

```
from google.cloud import aiplatform
from kfp.v2 import compiler

PROJECT_ID = "module-8-homework-478307"
LOCATION   = "us-central1"
BUCKET     = "gs://breast_cancer_train_test"
INPUT_CSV  = "gs://breast_cancer_train_test/breast_cancer_train.csv"

aiplatform.init(project=PROJECT_ID, location=LOCATION, staging_bucket=BUCKET)

# Compile
compiler.Compiler().compile(
    pipeline_func=breast_cancer_pipeline,
    package_path="breast_cancer_pipeline.json",
)

# Run
pipeline_job = aiplatform.PipelineJob(
    display_name="breast_cancer_pipeline",
    template_path="breast_cancer_pipeline.json",
    pipeline_root=BUCKET,
    parameter_values={
        "breast_cancer_dataset_path": INPUT_CSV,
    },
    enable_caching=True,
)

pipeline_job.submit()
```

USC Viterbi School of Engineering

Step 8

Vertex AI Pipeline Graphic



Step 9

Review Model Performance

On the following page, paste a screenshot showing the completed pipeline with the evaluation node selected and showing the accuracy and F1 score metrics.

USC Viterbi School of Engineering

Step 9

Review Model Performance