# AIG130 Lab 5

# AutoML Step-by-Step Walkthrough

# Group 5

## Part 1:

### Introduction and AutoML Platform Selection

**Platform Chosen:** *Google Cloud Platform*

**Introduction:**

This project demonstrates the application of Google Cloud Vertex AI AutoML for solving real-world machine learning problems involving both classification and regression tasks. As part of AIG130 Lab 5, the goal was to explore an AutoML platform, understand its workflow, and apply it to one or more datasets to build and evaluate predictive models without manually coding each model.

Google Vertex AI AutoML was selected for its ease of integration, automated workflow, and support for various ML tasks. With its powerful Python SDK and cloud-based infrastructure, Vertex AI simplifies the entire machine learning pipeline—from data preparation to model training, evaluation, deployment, and prediction.

In this lab, two datasets were used to showcase the flexibility and efficiency of Vertex AI AutoML:

* A ***classification task*** using the Phishing Websites dataset (from the UCI repository) to detect malicious websites.
* A ***regression* *task*** using the Flight Price Prediction dataset (from Kaggle) to estimate airline ticket prices.

By leveraging AutoML, the project demonstrates how modern cloud platforms can streamline machine learning development, enabling rapid experimentation, reduced manual effort, and scalable deployment. The lab provides a hands-on experience of building end-to-end ML solutions using AutoML, fulfilling both technical and learning objectives of the course.

### Features and Capabilities of Vertex AI AutoML

* **End-to-End Automation**: Automates data preprocessing, model training, tuning, evaluation, and deployment.
* **Supports Multiple Tasks**: Handles classification and regression on tabular data.
* **Python SDK Integration**: Easy scripting using the google-cloud-aiplatform library.
* **Built-in Evaluation**: Provides metrics like accuracy, AUC, MAE, RMSE, and more.
* **Cloud-Powered**: Uses Google Cloud infrastructure for scalable and fast training.
* **Model Deployment**: One-click deployment to endpoints for real-time or batch predictions.
* **Resource Management**: Easy cleanup of models, datasets, and endpoints to control costs.
* **User Interface**: Offers a web-based console to monitor training and view results.

## Part 2:

**Project Chosen from AIG100:** [*https://github.com/jcp-tech/Seneca\_Class\_Notes/blob/master/Semester%201/AIG100%20-%20Machine%20Learing/Project%202/Project%202%20-%20AIG100%20-%20Jonathan%20Chacko.ipynb*](https://github.com/jcp-tech/Seneca_Class_Notes/blob/master/Semester%201/AIG100%20-%20Machine%20Learing/Project%202/Project%202%20-%20AIG100%20-%20Jonathan%20Chacko.ipynb)

### 1. Environment Setup and Library Installation

The project began with setting up the Python environment, installing all required libraries including:

* *google-cloud-aiplatform* for accessing Vertex AI services,
* *google-cloud-storage* for interacting with Cloud Storage,
* *ucimlrepo* and *kagglehub* for dataset retrieval,
* And standard data processing libraries such as *pandas*, *numpy*, and *scikit-learn*.

Project credentials, region, and the storage bucket were configured via environment variables.

### 2. Dataset Preparation

Two distinct datasets were selected to demonstrate Vertex AI AutoML’s versatility:

* A **classification dataset**: *Phishing Websites* from the UCI Machine Learning Repository, used to identify malicious websites.
* A **regression dataset**: *Flight Price Prediction* from Kaggle, used to estimate airline ticket prices.

Each dataset was cleaned and prepared:

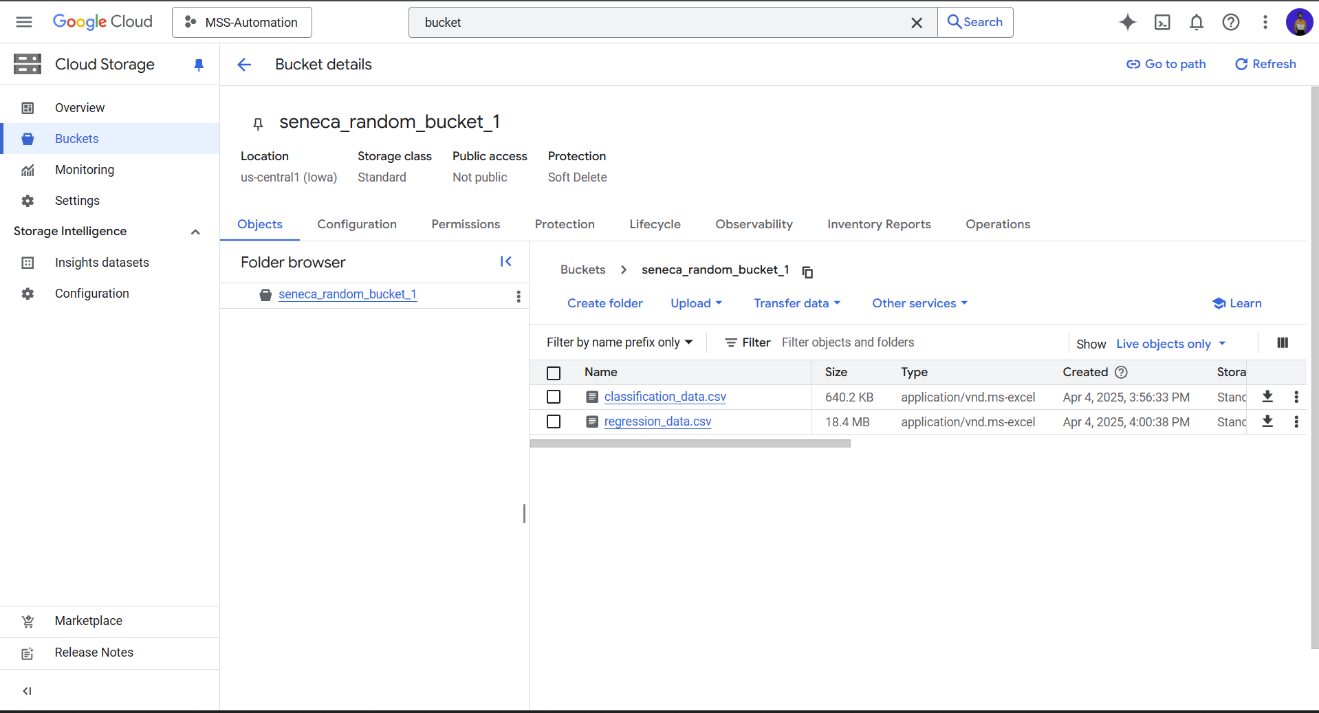
* Unnecessary columns were removed (e.g., index columns like 'Unnamed: 0').
* Features and target columns were clearly separated.
* Data was split into 80% training and 20% testing using train\_test\_split.

### 3. Upload to Google Cloud Storage

The training portions of each dataset were saved as CSV files:

* classification\_data.csv
* regression\_data.csv

These were uploaded to a designated Google Cloud Storage bucket (seneca\_random\_bucket\_1) using the Google Cloud Storage client. This ensured that the data could be accessed by Vertex AI for model training.

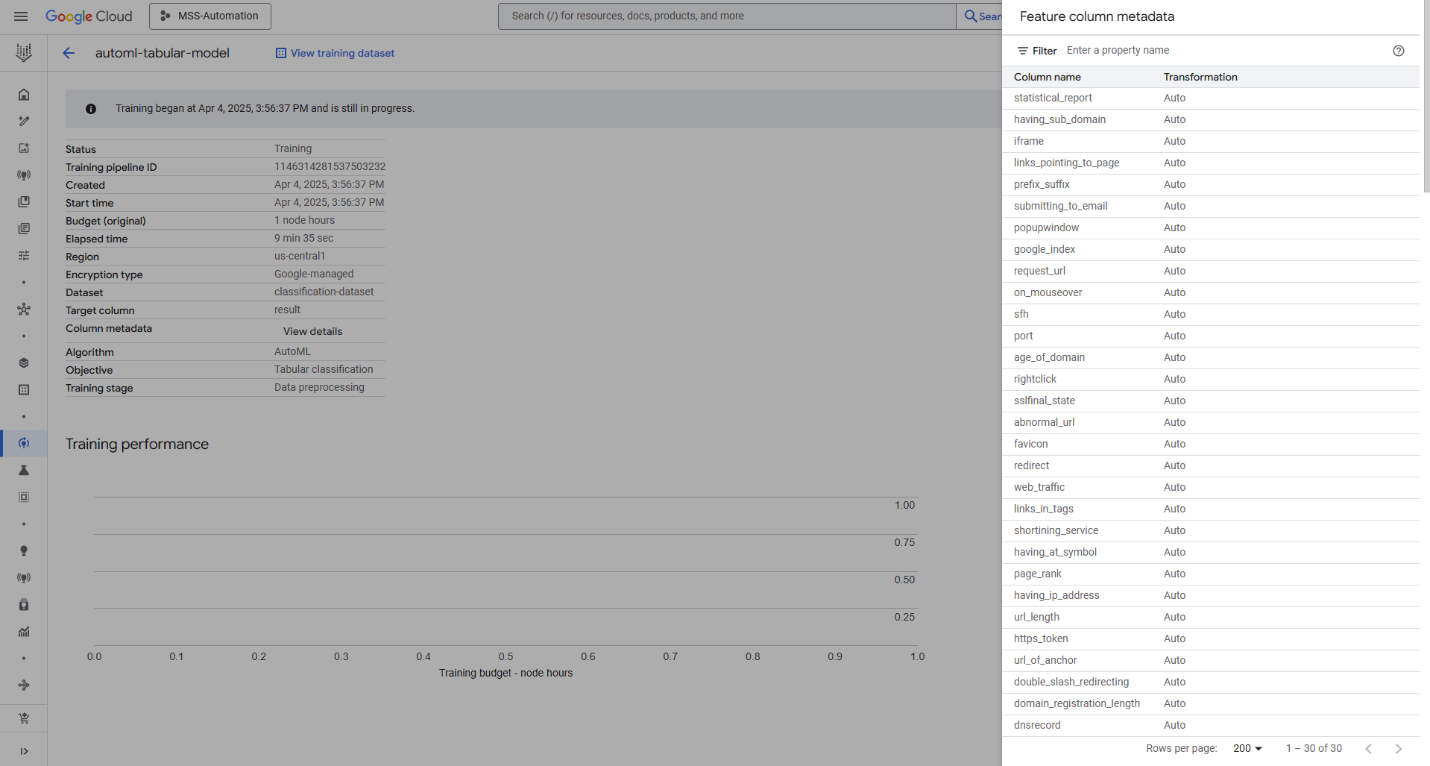


### 4. Vertex AI Dataset Registration

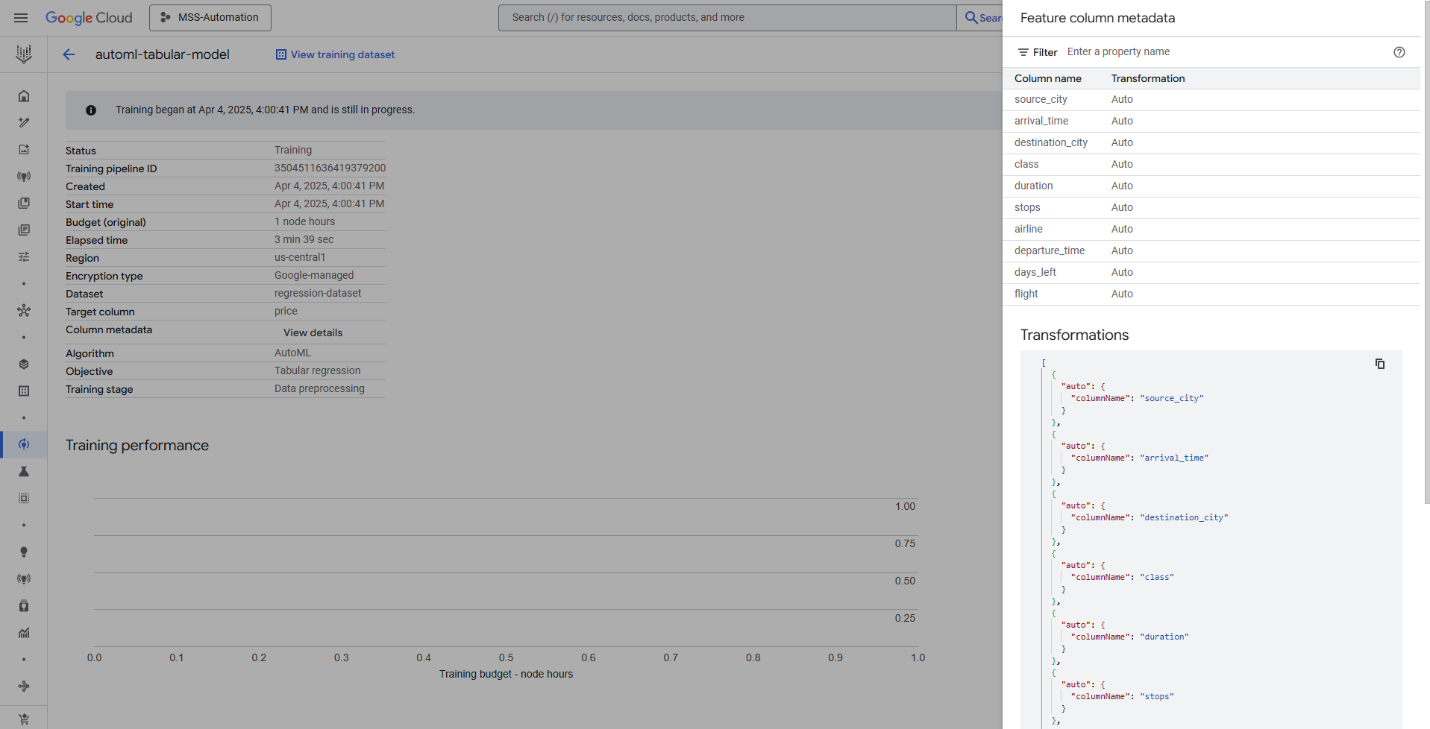
The uploaded datasets were then registered with Vertex AI as **Tabular Datasets**.

This step enabled Vertex AI to interpret the structure and content of the data and begin the AutoML process.

* + For **Classification**



* + For **Regression**

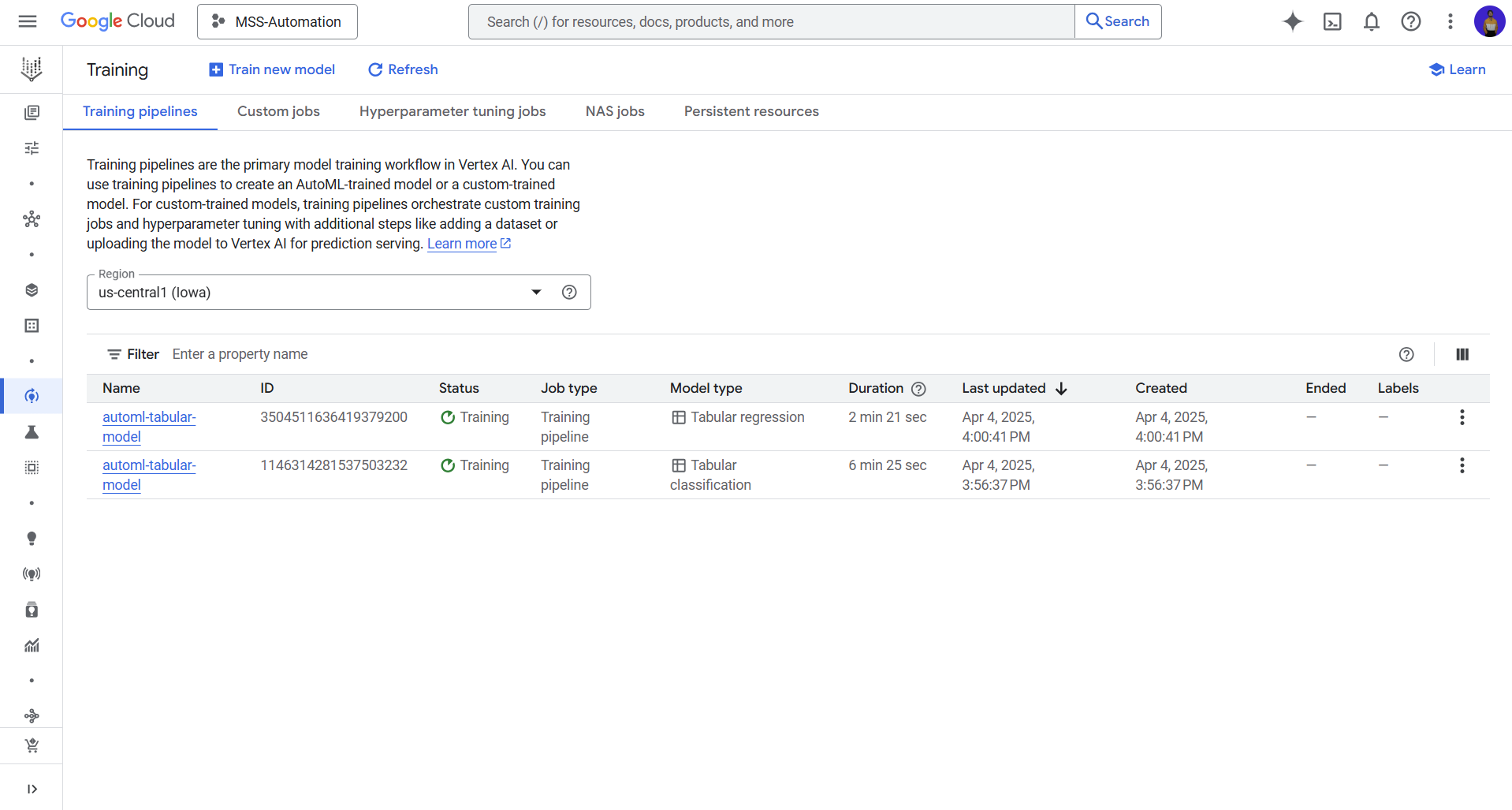


### 5. AutoML Model Training

Two separate **AutoML training jobs** were launched:

* A **tabular classification** job targeting the result column (phishing dataset),
* A **tabular regression** job targeting the price column (flight dataset).

The training process was tracked via the Vertex AI console, confirming the jobs were successfully created and executed.



### 6. Automatic Feature Transformation

Vertex AI AutoML automatically analyzed all feature columns and applied appropriate transformations (e.g., encoding categorical variables, normalizing numeric values). All features were labeled with the Auto transformation type, confirming that the platform managed preprocessing without manual intervention.

### 7. Model Deployment

Upon successful completion of training, each model was deployed to its own Vertex AI endpoint.

This step made the models accessible for real-time online predictions via REST APIs or SDK calls.

### 8. Inference and Prediction

The held-out test data was passed to the deployed.

This generated:

* Predicted labels for the classification task,
* Estimated flight prices for the regression task.

These predictions were collected for analysis and validation.

### 9. Evaluation and Validation

Model performance was evaluated using:

* **Accuracy, precision, recall, and AUC** for the classification model (viewed through Vertex AI console).
* **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)** for the regression model.

These metrics confirmed the effectiveness of the models built using AutoML and validated their suitability for their respective use-cases.

## Conclusion

This lab demonstrated how Google Cloud Vertex AI AutoML simplifies the machine learning workflow for both classification and regression tasks. By automating data preprocessing, model training, and deployment, Vertex AI enabled efficient development using real-world datasets. The platform’s ease of use and powerful features proved effective in building accurate, scalable models with minimal manual effort, highlighting AutoML’s value in accelerating ML projects.