

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

Export and deploy a BigQuery Machine Learning Model for Prediction

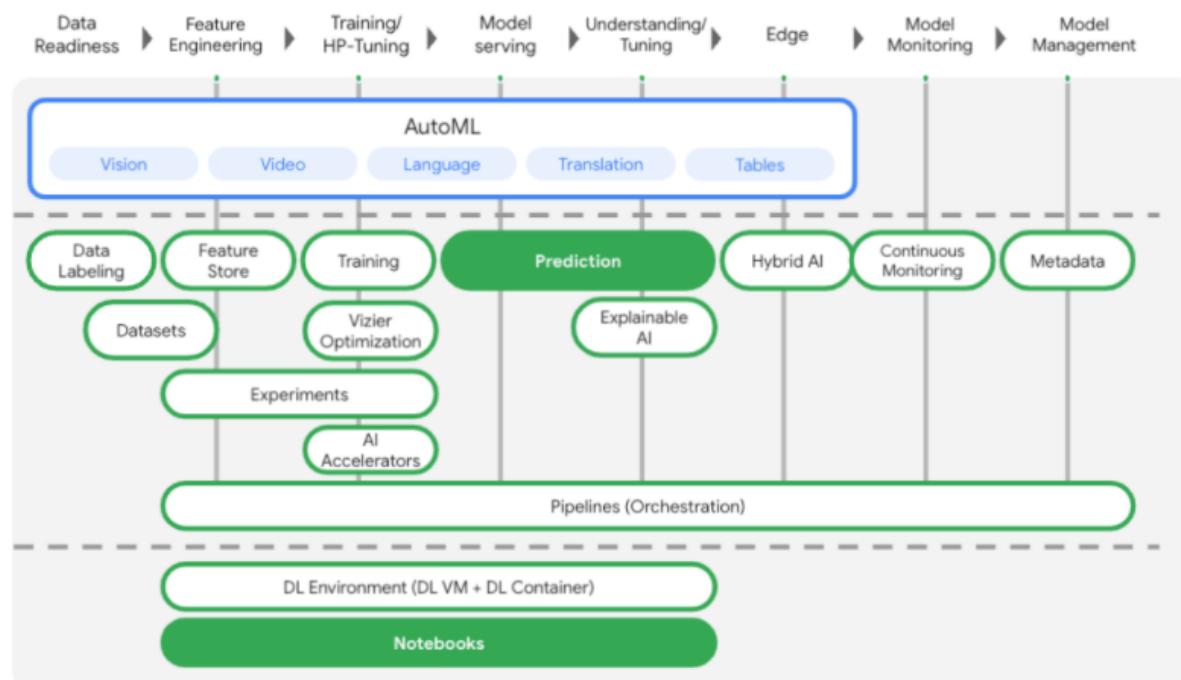
Reference: <https://codelabs.developers.google.com/codelabs/bqml-vertex-prediction#0>

Objectives:

- Train a model with BigQuery Machine Learning (BQML)
- Export your BQML model to Cloud Storage
- Deploy your trained BQML to Vertex AI
- Get predictions on your deployed model

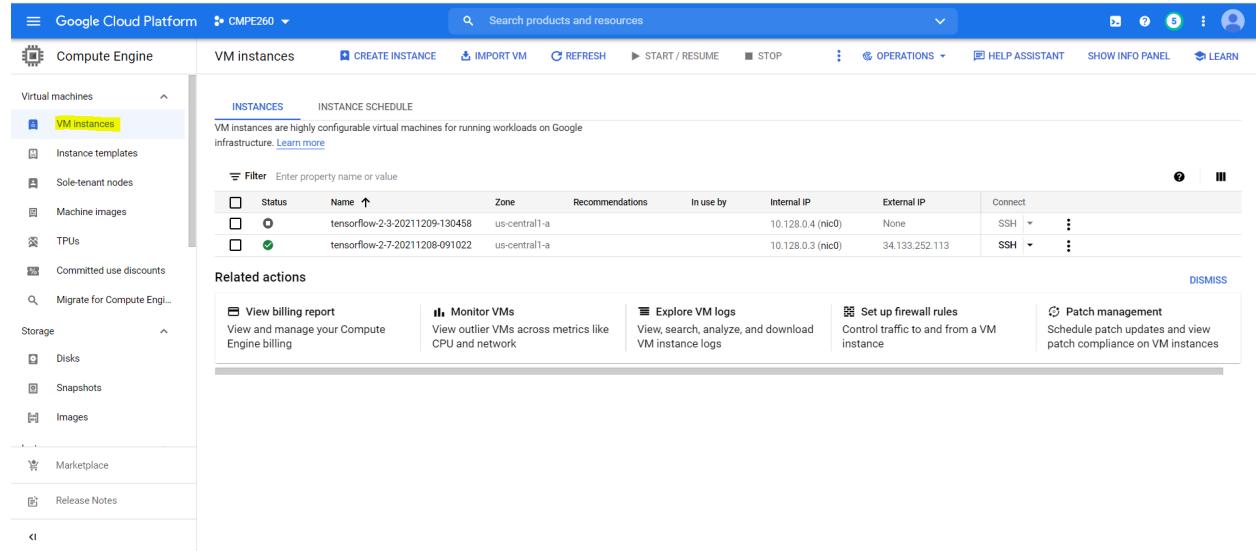
Intro to Vertex AI

Vertex AI includes many different products to support end-to-end ML workflows. This lab will focus on the products highlighted below: Prediction and Notebooks.



Set up your environment

Enable the Compute Engine API



The screenshot shows the Google Cloud Platform Compute Engine VM instances page. The left sidebar is collapsed, showing options like VM instances, Instance templates, Sole-tenant nodes, Machine images, TPUs, Committed use discounts, Migrate for Compute Engine, Disks, Snapshots, Images, Marketplace, and Release Notes. The main area displays two VM instances:

Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
Running	tensorflow-2-3-20211209-130458	us-central1-a			10.128.0.4 (nic0)	None	SSH
Running	tensorflow-2-7-20211208-091022	us-central1-a			10.128.0.3 (nic0)	34.133.252.113	SSH

Below the table, there are related actions: View billing report, Monitor VMs, Explore VM logs, Set up firewall rules, and Patch management.

Enable the Vertex AI API

Google Cloud Platform CMPE260 Search products and resources

Vertex AI Dashboard

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](#)

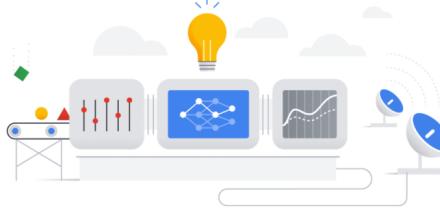
Try an interactive tutorial to learn how to train, evaluate, and deploy a Vertex AI AutoML or custom-trained model. [VIEW TUTORIALS](#)

Region: us-central1 (Iowa)

Recent datasets: Iowa_daily (37 minutes ago), +CREATE DATASET

Recent models: beans-model-pipeline (3 hours ago), beans-model-pipeline (3 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



Enable the Container Registry API

Google Cloud Platform CMPE260 Search products and resources

Container Registry Repositories

Images Settings

Transition to Artifact Registry

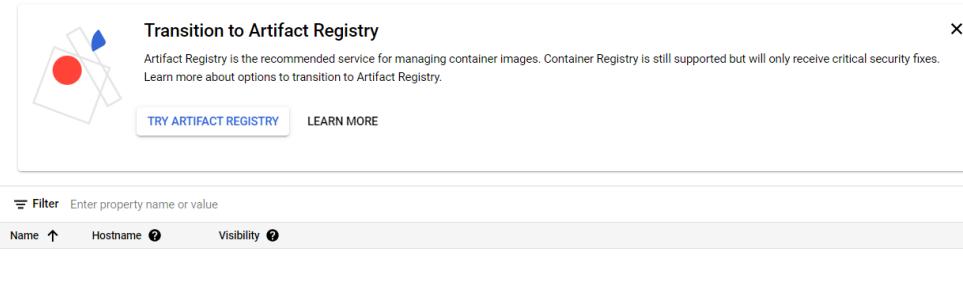
Artifact Registry is the recommended service for managing container images. Container Registry is still supported but will only receive critical security fixes. Learn more about options to transition to Artifact Registry.

TRY ARTIFACT REGISTRY LEARN MORE

Filter Enter property name or value

Name ↑ Hostname ? Visibility ?

Google Container Registry provides secure, private Docker repository storage on Google Cloud Platform. You can now extend to much more than



Create a Vertex AI Workbench instance

Navigate to Vertex AI → Workbench

The screenshot shows the Google Cloud Platform dashboard for project CMPE260. The left sidebar has a 'Vertex AI' section expanded, with 'Workbench' selected. The main area displays 'RECOMMENDATIONS' and three cards: 'Compute Engine' (CPU % utilization), 'Google Cloud Platform status' (Cloud SQL migration), and 'Billing' (USD \$0.00). A link to 'Go to Compute Engine' is also present.

Select **TensorFlow Enterprise 2.3 (with LTS)** instance type **without GPUs**:

The screenshot shows the 'Notebooks' page under the 'Workbench' section of Vertex AI. It lists various notebook environments, with 'TensorFlow Enterprise' highlighted. A dropdown menu for 'TensorFlow Enterprise 2.3 (with LTS)' is open, showing the option 'Without GPUs' selected. Other options include 'With 1 NVIDIA Tesla T4' and 'With 1 NVIDIA V100'. The right side of the screen shows a preview of the selected environment.

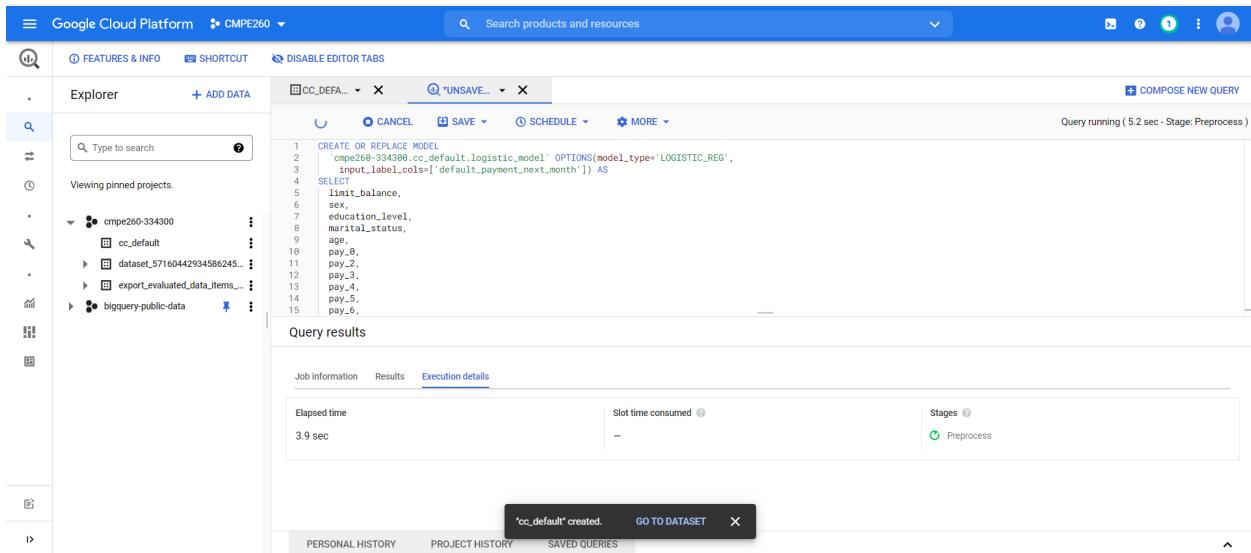
The screenshot shows the Google Cloud Platform Vertex AI Workbench interface. On the left, there's a sidebar with various options like Dashboard, Datasets, Features, Labeling tasks, Pipelines, Training, Experiments, Models, Endpoints, Batch predictions, and Metadata. The 'Workbench' section is selected. In the center, a 'New notebook' dialog box is open. It contains fields for 'Notebook name' (set to 'tensorflow-2-3-20211209-130458'), 'Region' (set to 'us-central1 (Iowa)'), and 'Zone' (set to 'us-central1-a'). Below these are 'Notebook properties' including environment ('TensorFlow Enterprise 2.3 (with LTS and Intel® MKL-DNN/MKL)'), machine type ('4 vCPUs, 15 GB RAM'), boot disk ('100 GB Standard persistent disk'), data disk ('100 GB Standard persistent disk'), and subnetwork ('default(10.128.0.0/20)'). There are also fields for external IP (Ephemeral(Automatic)), permission (Compute Engine default service account), and estimated cost (\$102.70 monthly, \$0.141 hourly). At the bottom right of the dialog are 'ADVANCED OPTIONS', 'CANCEL', and 'CREATE' buttons.

Train a BigQuery ML model

Step 1: Create a BigQuery dataset in your project

The screenshot shows the Google Cloud Platform BigQuery Editor interface. On the left, there's an 'Explorer' sidebar showing pinned projects, including 'cmpe260-334300' which is expanded to show datasets like 'dataset_57160442934586245...', 'export_evaluated_data_items...', and 'bigquery-public-data'. The main area is titled 'EDITOR' and has tabs for 'RUN', 'SAVE', 'SCHEDULE', and 'MORE'. To the right, a 'Create dataset' dialog box is open. It requires 'Project ID' (set to 'cmpe260-334300') and 'Dataset ID' (set to 'cc_default'). It also includes fields for 'Data location' (set to 'us-central1 (Iowa)'), 'Default table expiration' (unchecked), and 'Encryption' (radio button selected for 'Google-managed encryption key'). At the bottom right of the dialog are 'CREATE DATASET', 'CHANGE', and 'CANCEL' buttons.

Step 2: Run a CREATE MODEL query

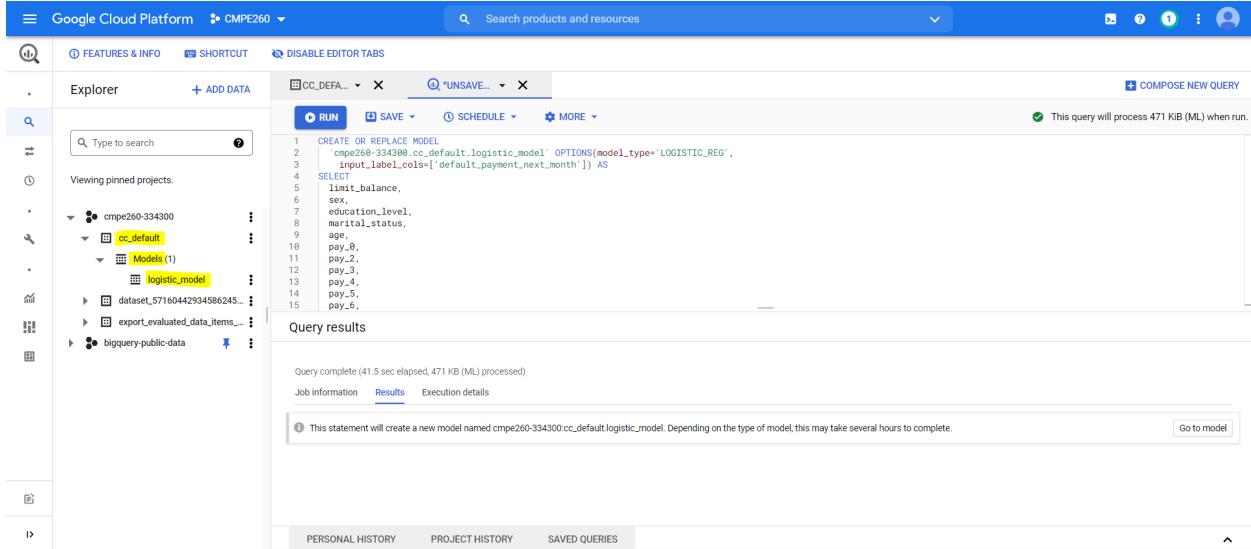


The screenshot shows the Google Cloud Platform BigQuery interface. In the top navigation bar, it says "Google Cloud Platform CMPE260". The search bar contains "Search products and resources". On the left, there's a sidebar with "FEATURES & INFO" and "SHORTCUT". Below that is an "Explorer" section with a search bar and a pinned projects list. One project, "cmpe260-334300", is expanded, showing datasets like "cc_default", "dataset_57160442934586245...", "export_evaluated_data_items...", and "bigquery-public-data". The main area is titled "cc_DEFAL... X" and has tabs for "CREATE OR REPLACE MODEL", "CANCEL", "SAVE", "SCHEDULE", and "MORE". A code editor window displays the following SQL query:

```
1 CREATE OR REPLACE MODEL
2   `cmpe260-334300.cc_default.logistic_model` OPTIONS(model_type='LOGISTIC_REG',
3     input_label_cols=['default_payment_next_month']) AS
4   SELECT
5     limit_balance,
6     sex,
7     education_level,
8     marital_status,
9     age,
10    pay_0,
11    pay_2,
12    pay_3,
13    pay_4,
14    pay_5,
15    pay_6,
```

Below the code editor is a "Query results" section with tabs for "Job information", "Results", and "Execution details". The "Elapsed time" is listed as 3.9 sec. The "Execution details" tab shows a "Slot time consumed" of 0 sec and a "Stages" status of "Preprocess". At the bottom of the results section, a message says "'cc_default' created." and "GO TO DATASET".

Check for the model created in `logistic_model`



This screenshot shows the same Google Cloud Platform BigQuery interface after the model has been created. The "cc_default" dataset is now expanded to show a "Models(1)" folder containing a single "logistic_model". The main query editor window remains the same, displaying the CREATE MODEL query. A message at the bottom of the results section states "Query complete (41.5 sec elapsed, 471 KB (ML) processed)". A tooltip over the "logistic_model" entry in the "Models(1)" folder says: "This statement will create a new model named cmpe260-334300.cc_default.logistic_model. Depending on the type of model, this may take several hours to complete." There is also a "Go to model" link.

Export your BigQuery ML model

Step 1: Create a Cloud Storage Bucket for your model

The screenshot shows the Google Cloud Platform interface for a project named 'CMPE260'. In the left sidebar, under 'Explorer', there is a 'Models' section containing one item: 'logistic_model'. The main content area displays the 'logistic_model' details. The 'Model Details' section includes fields like Model ID (cmpe260-334300:cc_default.logistic_model), Model type (LOGISTIC_REGRESSION), Data location (US), and various metadata such as Date created, Model expiration, and Loss type. Below this is the 'Training Options' section.

The screenshot shows the same Google Cloud Platform interface as above, but with a modal dialog titled 'Export model to Google Cloud Storage' overlaid. The dialog contains a single input field labeled 'Select GCS location *' with a 'BROWSE' button next to it. At the bottom of the dialog are 'SUBMIT' and 'CANCEL' buttons.

Create a Bucket:

The screenshot shows the Google Cloud Platform interface. On the left, the 'Explorer' sidebar lists projects and datasets. In the center, a model named 'logistic_model' is selected in the 'LOGISTI...' tab. On the right, a modal window titled 'Select object' is open, displaying a list of buckets. The 'Create new bucket' button is circled in red.

The screenshot shows the 'Create a bucket' wizard. Step 1: 'Name your bucket' - The input field contains 'oc_logistics'. Step 2: 'Choose where to store your data' - The 'Region' radio button is selected, with 'us-central1 (Iowa)' chosen from the dropdown. Step 3: 'Choose a default storage class for your data' - The 'Standard' storage class is selected. The 'CONTINUE' button is visible at the bottom of each step.

Upgrade your account to avoid a break in service (\$151.76 credit and 5 days left in your trial).

Google Cloud Platform CMPE260

Search products and resources

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

Explorer + ADD DATA LOGISTI... X

Type to search

Viewing pinned projects.

- cmpe260-334300
- cc_default
- Models (1)
 - logistic_model
- dataset_57160442934586245...
- export_evaluated_data_items...
- bigrquery-public-data

logistic_model

DETAILS TRAINING EVALUATION SCHEMA

Model type LOGISTIC_REGRESSION Data location US

Model Details EDIT

Model ID	cmpe260-334300:cc_default.logistic_model
Description	
Labels	
Date created	Thursday, December 9, 2021 at 7:17:35 PM GMT-08:00
Model expiration	Never
Date modified	Thursday, December 9, 2021 at 7:17:35 PM GMT-08:00
Data location	US
Model type	LOGISTIC_REGRESSION
Loss type	Mean log loss
Training data	TEMPORARY TRAINING DATA TABLE
Evaluation data	TEMPORARY EVALUATION DATA TABLE

Training Options

PERSONAL HISTORY PROJECT HISTORY SAVED QUERIES

Location us-central1 (Iowa)

CONTINUE

Choose a default storage class for your data
Default storage class: Standard

Choose how to control access to objects

Prevent public access
Restrict data from being publicly accessible via the internet. Will prevent this bucket from being used for web hosting. [Learn more](#)

Enforce public access prevention on this bucket

Access control

Uniform Ensure uniform access to all objects in the bucket by using only bucket-level permissions (IAM). This option becomes permanent after 90 days. [Learn more](#)

Fine-grained Specify access to individual objects by using object-level permissions (ACLs) in addition to your bucket-level permissions (IAM). [Learn more](#)

CONTINUE

Choose how to protect object data
Protection tools: None
Data encryption: Google-managed key

CREATE CANCEL

Upgrade your account to avoid a break in service (\$151.76 credit and 5 days left in your trial).

Google Cloud Platform CMPE260

Search products and resources

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

Explorer + ADD DATA LOGISTI... X

Type to search

Viewing pinned projects.

- cmpe260-334300
- cc_default
- Models (1)
 - logistic_model
- dataset_57160442934586245...
- export_evaluated_data_items...
- bigrquery-public-data

logistic_model

DETAILS TRAINING EVALUATION SCHEMA

Model type LOGISTIC_REGRESSION Data location US

Model Details EDIT

Model ID	cmpe260-334300:cc_default.logistic_model
Description	
Labels	
Date created	Thursday, December 9, 2021 at 7:17:35 PM GMT-08:00
Model expiration	Never
Date modified	Thursday, December 9, 2021 at 7:17:35 PM GMT-08:00
Data location	US
Model type	LOGISTIC_REGRESSION
Loss type	Mean log loss
Training data	TEMPORARY TRAINING DATA TABLE
Evaluation data	TEMPORARY EVALUATION DATA TABLE

Training Options

PERSONAL HISTORY PROJECT HISTORY SAVED QUERIES

Export model to Google Cloud Storage

Exporting a model allows you to use this model in other systems/applications. Please select Cloud Storage path to export this model along with all asset files. [Learn more](#)

Select GCS location * cc_logistics/model-assets BROWSE

SUBMIT CANCEL

When the above job completes, we will see your model assets exported to the bucket you just created under a `model-assets` subdirectory

The screenshot shows the Google Cloud Platform Storage interface for the bucket `cc_logistics`. The bucket details page is displayed, showing basic information like location (us-central1 (Iowa)), storage class (Standard), public access (Not public), and protection (None). The `OBJECTS` tab is selected, showing a list of objects in the `model-assets` directory. The list includes:

Name	Type	Size	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiration date	Holds
assets/	Folder	—	—	—	—	—	—	—	—	—
saved_model.pb	application/octet-stream	74.2 KB	Dec 9, 2021...	Standard	Dec 9, 2021, ...	Not public	—	Google-managed key	—	None
variables/	Folder	—	—	—	—	—	—	—	—	—

Import the model to Vertex AI

Step 1: Import the model

Navigate to Vertex AI → Models

The screenshot shows the Google Cloud Platform Vertex AI interface. The sidebar navigation bar has several categories: Looker, Data Catalog, Data Fusion, Financial Services, Healthcare, Life Sciences, Dataprep, ARTIFICIAL INTELLIGENCE, and Vertex AI. The Vertex AI category is expanded, showing sub-options: AI Platform, Data Labeling, Document AI, Natural Language, Recommendation..., and Retail. The main content area displays the Vertex AI Models page, which includes tabs for PROTECTION, ACTIONS, and LIFECYCLE. The `Models` tab is currently selected. Below the tabs, there is a table showing a single entry:

Name	Type	Size	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiration date	Holds
saved_model.pb	application/octet-stream	74.2 KB	Dec 9, 2021...	Standard	Dec 9, 2021, ...	Not public	—	Google-managed key	—	None

Click on 'Import'

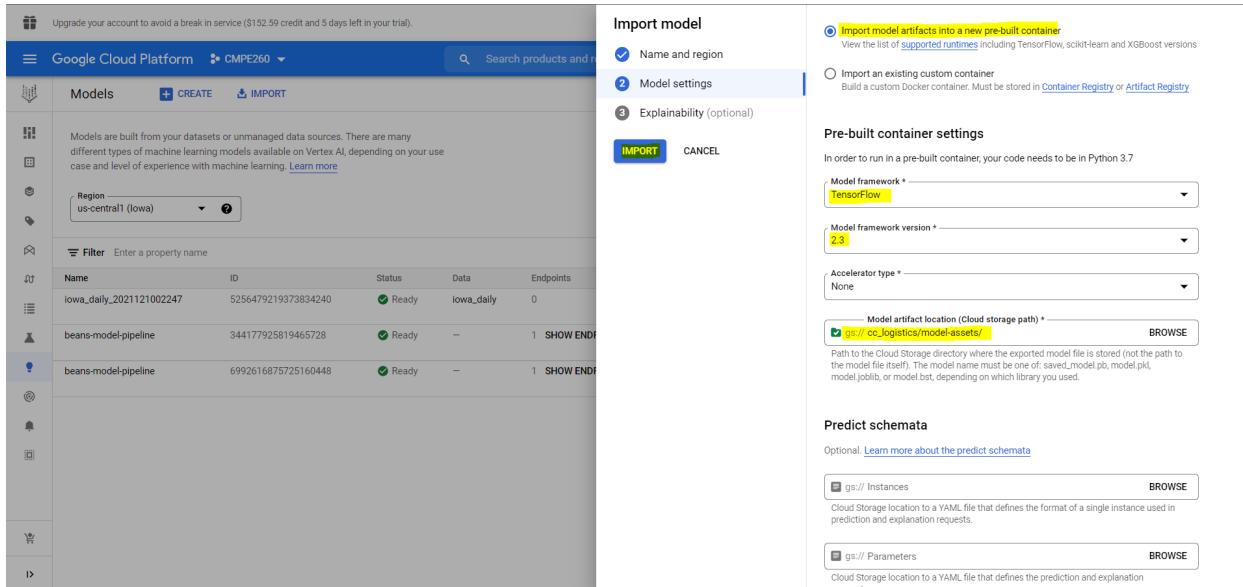
The screenshot shows the Google Cloud Platform Models page. At the top, there is a search bar and a 'REFRESH' button. Below the search bar, there are buttons for 'CREATE' and 'IMPORT'. A message states: 'Models are built from your datasets or unmanaged data sources. There are many different types of machine learning models available on Vertex AI, depending on your use case and level of experience with machine learning. [Learn more](#)'. A dropdown menu for 'Region' is set to 'us-central1 (Iowa)'. A 'Filter' input field is present. The main table lists three models:

Name	ID	Status	Data	Endpoints	Region	Type	Created	Notifications	Labels
iowa_daily_2021121002247	5256479219373834240	Ready	iowa_daily	0	us-central1	Tabular Forecasting AutoML	Dec 9, 2021, 4:28:37 PM		
beans-model-pipeline	344177925819465728	Ready	—	1 SHOW ENDPOINTS	us-central1	Imported Custom training	Dec 9, 2021, 1:42:22 PM		
beans-model-pipeline	6992616875725160448	Ready	—	1 SHOW ENDPOINTS	us-central1	Imported Custom training	Dec 9, 2021, 1:40:14 PM		

Enter the ModelName and Region:

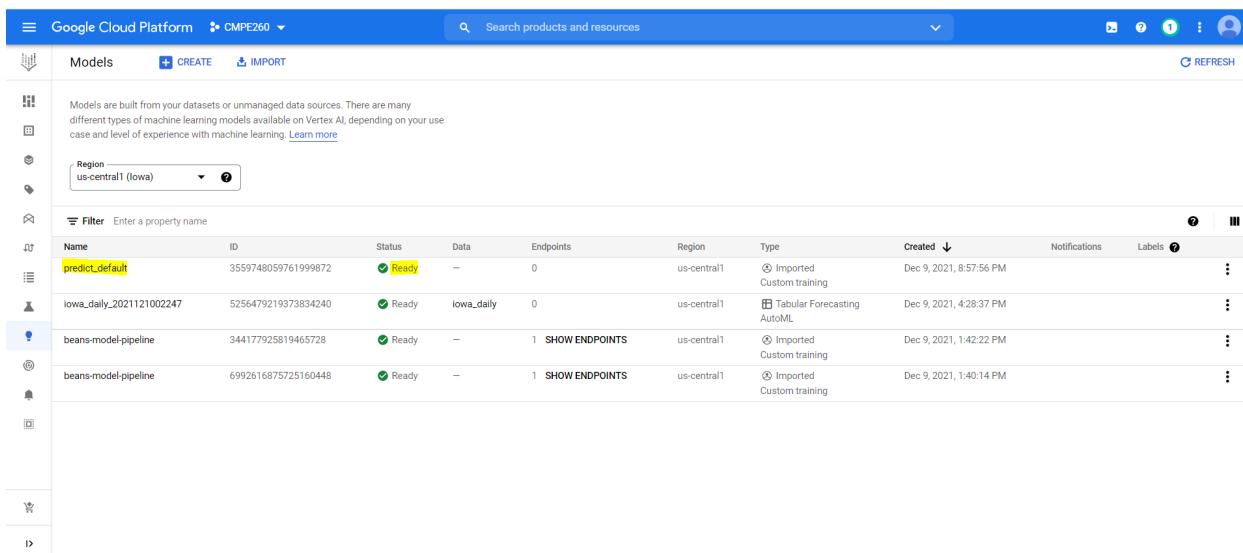
The screenshot shows the 'Import model' dialog box. It has three steps: 1. Name and region (selected), 2. Model settings, 3. Explainability (optional). Step 1 is titled 'Name and region' with sub-instructions: 'You can import model artifacts that have been trained outside of Google Cloud. Once your model has been imported, you can serve it for online or batch predictions and compare it against your other Cloud AI models. [More info](#)'. It contains fields for 'Model name *' (set to 'predict_default') and 'Region' (set to 'us-central1 (Iowa)'). Step 2 is titled 'Model settings' and Step 3 is titled 'Explainability (optional)'. At the bottom are 'IMPORT' and 'CANCEL' buttons, and a 'CONTINUE' button under 'ADVANCED OPTIONS'.

In **Model settings**, keep "Import model artifacts into a new pre-built container" selected.
In the **Model framework** dropdown, select **TensorFlow**. Then select **2.3** as the framework version.
In the Model artifact location field, click **Browse**, click into the GCS bucket you just created, and click on the **model-assets** directory:



The screenshot shows the Google Cloud Platform Models page. On the left, there's a sidebar with various icons. The main area shows a table of existing models with columns for Name, ID, Status, Data, and Endpoints. A modal window titled "Import model" is open on the right. It has three tabs: "Name and region" (selected), "Model settings" (highlighted in blue), and "Explainability (optional)". Under "Model settings", the "Model framework" dropdown is set to "TensorFlow" and the "Model framework version" dropdown is set to "2.3". The "Model artifact location (Cloud storage path)" input field contains the path "gs://cc_logistics/model-assets/" and has a "BROWSE" button. Below this, under "Predict schemata", there are two input fields: "Instances" (with a "BROWSE" button) and "Parameters" (with a "BROWSE" button).

Model is Imported



The screenshot shows the Google Cloud Platform Models page after the model has been imported. The table now includes a new row for "predict_default" located at "gs://cc_logistics/model-assets/". The columns include Name, ID, Status, Data, Endpoints, Region, Type, Created, Notifications, and Labels. The "predict_default" entry shows a status of "Ready", was created on Dec 9, 2021, at 8:57:56 PM, and is of type "Imported Custom training". The other four rows remain the same as in the previous screenshot.

Deploy the model to an endpoint

Step 1: Creating an endpoint

Navigate to the Model → Deploy & Test

The screenshot shows the Google Cloud Platform interface for a machine learning model named "predict_default". The "DEPLOY & TEST" tab is active. The "Deploy your model" section contains a brief description of what endpoints are and a prominent blue "DEPLOY TO ENDPOINT" button. The "Test your model" section has a note that deployment is required first, with a "Pricing guide" link. A sidebar on the left lists various project components.

The screenshot shows the "Deploy to endpoint" dialog box. It's step 1 of 3, titled "Define your endpoint". The "Create new endpoint" radio button is selected, and the endpoint name "default_pred_y1" is entered. The "Region" dropdown is set to "us-central1 (Iowa)". Under the "Access" section, the "Standard" radio button is selected, with a note about REST API access. The "Private" option is also shown with a note about VPC network access. A "CONTINUE" button is at the bottom right.

Give your endpoint a name, like `default_pred_v1`, leave the traffic splitting settings as is, and then select a machine type for your model deployment. We used an `n1-highcpu-2` here, but you can choose whichever machine type you'd like.

The screenshot shows two main panels. On the left, the 'predict_default' endpoint details page is visible, showing sections for 'Deploy & Test', 'Batch Predictions', and 'Model Properties'. A 'DEPLOY TO ENDPOINT' button is prominent. On the right, a detailed 'Deploy to endpoint' dialog is open, divided into three steps: 1. Define your endpoint (with 'predict_default' selected), 2. Model settings (set to 'n1-highcpu-2'), and 3. Model monitoring (disabled). The 'Compute resources' section shows 'Traffic split' at 100%, 'Minimum number of compute nodes' set to 1, and 'Machine type' set to 'n1-highcpu-2, 2 vCPUs, 1.8 GB memory'. Advanced scaling options are also shown.

The screenshot shows the same 'predict_default' endpoint details page. The 'Status' column for the endpoint row now shows 'Deploying model'. The 'JSON request' and 'Response' sections at the bottom are empty, indicating no requests have been made yet.

The Endpoint is created.

This screenshot shows the 'predict_default' endpoint in the Google Cloud Platform Predictions interface. The 'DEPLOY & TEST' tab is selected. In the 'Deploy your model' section, a table lists a single endpoint named 'default_pred_v1' with status 'Active'. The 'Test your model' section contains a 'PREVIEW' button and a JSON request builder. The 'JSON request' field contains the following JSON:

```
{
  "instances": [
    {
      "sample_key": "sample_value"
    }
  ]
}
```

The 'Response' field is currently empty.

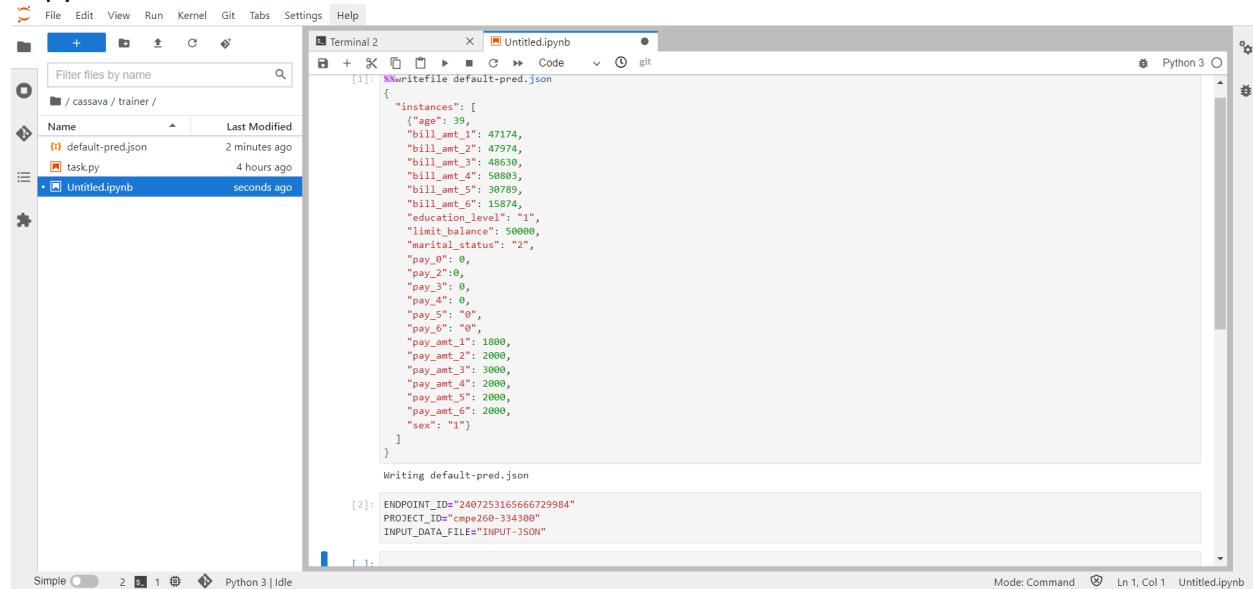
Click on 'Sample Request', Copy the lines in Step 4

This screenshot shows the same 'predict_default' endpoint interface. The 'Test your model' section now includes a 'Sample Request' button. The 'JSON request' field is identical to the one in the previous screenshot. To the right, a 'Sample Request' panel is open, showing steps for executing a command-line request. Step 4 shows the command:

```
$ curl \
-X POST \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
https://us-central1-aiplatform.googleapis.com/v1/projects/cmpe260-334300/endpoints/2407253165666729984:predict \
-d @"$INPUT_DATA_FILE"
```

A yellow box highlights the project ID 'cmpe260-334300' in the URL.

Copy in Notebook Cell:



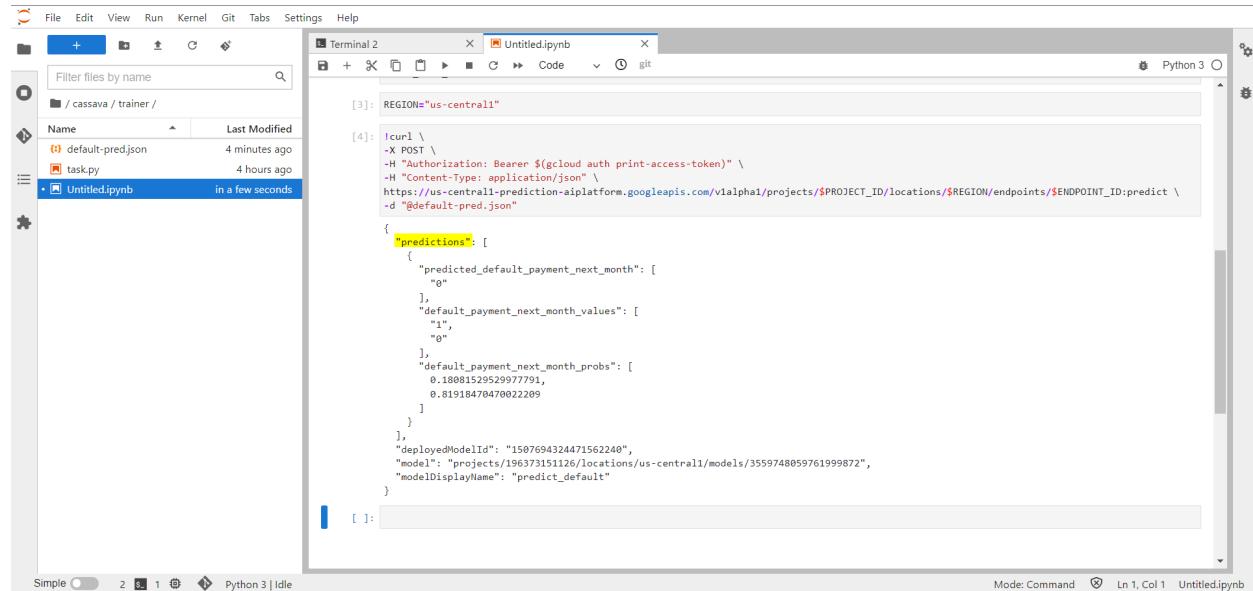
The screenshot shows a Jupyter Notebook interface with a file tree on the left containing 'cassava / trainer /', 'default-pred.json' (modified 2 minutes ago), 'task.py' (modified 4 hours ago), and 'Untitled.ipynb' (modified seconds ago). The main area has two terminal tabs. Terminal 2 is active and displays Python code writing a JSON file:

```
[1]: #!/usr/bin/env python3
with open('default-pred.json', 'w') as f:
    f.write(json.dumps({
        "instances": [
            {"age": 39,
             "bill_amt_1": 47174,
             "bill_amt_2": 47974,
             "bill_amt_3": 48630,
             "bill_amt_4": 50803,
             "bill_amt_5": 30789,
             "bill_amt_6": 15874,
             "education_level": "1",
             "limit_balance": 50000,
             "marital_status": "2",
             "pay_0": 0,
             "pay_2": 0,
             "pay_3": 0,
             "pay_4": 0,
             "pay_5": 0,
             "pay_6": 0,
             "pay_amt_1": 1800,
             "pay_amt_2": 2000,
             "pay_amt_3": 3000,
             "pay_amt_4": 2000,
             "pay_amt_5": 2000,
             "pay_amt_6": 2000,
             "sex": "1"
        ]
    }))
Writing default-pred.json
```

Terminal 1 is visible below it, showing environment variables:

```
[2]: ENDPOINT_ID="2407253165666729984"
PROJECT_ID="cmpe260-334300"
INPUT_DATA_FILE="INPUT-JSON"
```

Prediction Result:



The screenshot shows a Jupyter Notebook interface with a file tree on the left containing 'cassava / trainer /', 'default-pred.json' (modified 4 minutes ago), 'task.py' (modified 4 hours ago), and 'Untitled.ipynb' (modified in a few seconds). The main area has two terminal tabs. Terminal 2 is active and displays a curl command for prediction:

```
[3]: REGION="us-central1"
[4]: curl \
-X POST \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
https://us-central1-prediction-apiplatform.googleapis.com/v1alpha1/projects/$PROJECT_ID/locations/$REGION/endpoints/$ENDPOINT_ID:predict \
-d "default-pred.json"

{
  "predictions": [
    {
      "predicted_default_payment_next_month": [
        "0"
      ],
      "default_payment_next_month_values": [
        "1",
        "0"
      ],
      "default_payment_next_month_probs": [
        0.1808152952997791,
        0.81918470470922209
      ]
    }
  ],
  "deployedModelId": "1507694324471562240",
  "model": "projects/196373151126/locations/us-central1/models/3559748059761999872",
  "modelDisplayName": "predict_default"
}
```

Cleanup

Stop the Notebook Instance:

The screenshot shows the Google Cloud Platform Notebooks interface. At the top, there are buttons for NEW NOTEBOOK, REFRESH, START, STOP (which is highlighted in yellow), RESET, and DELETE. Below these are tabs for MANAGED NOTEBOOKS (PREVIEW) and USER-MANAGED NOTEBOOKS. The main area displays a table of notebook instances with columns: Notebook name, Zone, Auto-upgrade, Environment, Machine type, GPUs, Permission, and Last modified. A tooltip message at the top states: "As of the M80 DLVM release, all environments will include JupyterLab 3.x by default. To continue using an existing environment's JupyterLab 1.x version, disable auto-upgrade (if enabled) and do not manually upgrade the environment to a new environment version. To create new Notebooks with JupyterLab 1.x installed, see creating specific versions of Notebooks." A note below says: "Notebooks have JupyterLab pre-installed and are configured with GPU-enabled machine learning frameworks. Learn more".

Notebook name	Zone	Auto-upgrade	Environment	Machine type	Gpus	Permission	Last modified
tensorflow-2-3-20211209-130458	OPEN JUPYTERLAB	us-central1-a	TensorFlow:2.3	4 vCPUs, 15 GB RAM	None	Service account	Dec 9, 2021, 3:35:42 PM
tensorflow-2-7-20211208-091022	OPEN JUPYTERLAB	us-central1-a	TensorFlow:2.7	4 vCPUs, 15 GB RAM	None	Service account	Dec 9, 2021, 3:39:09 PM

Delete the Endpoint

The screenshot shows the Google Cloud Platform Endpoints interface. At the top, there are buttons for CREATE ENDPOINT and REFRESH. Below these are tabs for ENDPOINTS (selected) and EDIT LABELS. The main area displays a table of endpoints with columns: Name, ID, Status, Models, Region, Monitoring, Most recent alerts, Last updated, API, Notification, and Labels. A tooltip message at the top states: "Endpoints are machine learning models made available for online prediction requests. Endpoints are useful for timely predictions from many users (for example, in response to an application request). You can also request batch predictions if you don't need immediate results." A note below says: "To create an endpoint, you need at least one machine learning model. Learn more". A dropdown menu for Region is set to us-central1 (Iowa). A context menu is open over the first endpoint (default_pred_v1), showing options: View logs, Edit labels, Rename endpoint, and Remove endpoint.

Name	ID	Status	Models	Region	Monitoring	Most recent alerts	Last updated	API	Notification	Labels
default_pred_v1	2407253165666729984	Active	1	us-central1	Disabled	—	Dec 9, 2021, 9:09:16 PM	Sample request		
beans-model-pipeline_endpoint	1416461247645220864	Active	1	us-central1	Disabled	—	Dec 9, 2021, 1:50:03 PM	Sample request		
beans-model-pipeline_endpoint	3282992186957758464	Active	1	us-central1	Disabled	—	Dec 9, 2021, 1:49:59 PM	Sample request		