

Classify Images with Pre-built ML Models using Cloud Vision API and AutoML

Learning Objectives:

- Setup API key for ML Vision API
- Invoke the pretrained ML Vision API to classify images
- Review label predictions from Vision API
- Train and evaluate custom AutoML Vision image classification model
- Predict with AutoML on new image

AutoML Vision provides an interface for all the steps in training an image classification model and generating predictions on it. Start by enabling the Cloud AutoML API.

Open the navigation menu and select **APIs & Services > Library**. In the search bar type in "Cloud AutoML API". Click on the **Cloud AutoML API** result and then click **ENABLE**.

This may take a minute. You should now be on the following page (ensure that the **MANAGE** button appears and **API enabled** is also displayed):

Next you will need to issue each of the commands that below appear in the GCP Console. In Cloud Shell paste these commands to create environment variables for your Project ID and Qwiklabs Username,

replacing <QWIKLABS_USERNAME> with the user name you logged into the lab with:

```
export PROJECT_ID=$DEVSHELL_PROJECT_ID  
export QWIKLABS_USERNAME=<QWIKLABS_USERNAME>
```

Now, create a Storage Bucket for the images you will use in testing. Create one by running the following command:

```
gsutil mb -p $PROJECT_ID \  
  -c regional \  
  -l us-central1 \  
  gs://$PROJECT_ID-vcm/
```

Leave your Cloud Shell window open for additional steps to follow.

Now, create a Storage Bucket for the images you will use in testing. Create one by running the following command:

```
gsutil mb -p $PROJECT_ID \
    -c regional \
    -l us-central1 \
    gs://$PROJECT_ID-vcm/
```

Leave your Cloud Shell window open for additional steps to follow.

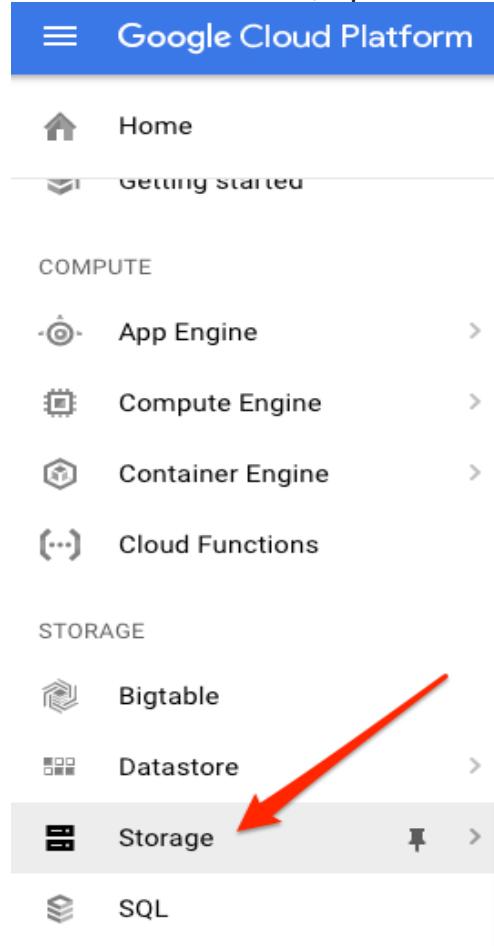
Now open a new browser tab and navigate to the [AutoML UI](#). You will be taken to the AutoML Vision Datasets page once the APIs are verified.

The screenshot shows the Google Cloud Platform interface with the AutoML Vision datasets page open. The URL in the address bar is `console.cloud.google.com/vision/datasets?project=qwiklabs-gcp-04-d0e433d98e62&folder=&organizationId=`. The page title is "Datasets - Vision - qwiklabs-gcp-04-d0e433d98e62". The left sidebar has "Vision" selected, showing "Dashboard", "Datasets" (which is highlighted in blue), and "Models". The main content area is titled "Datasets" and includes a "NEW DATASET" button. A table header is shown with columns: Name, Type, Total images, Labeled images, Last updated, and Status. Below the header, it says "No rows to display".

Upload training images to Google Cloud Storage

In order to train a model to classify images of clouds, you need to provide labeled training data so the model can develop an understanding of the image features associated with different types of clouds. In this example your model will learn to classify three different types of clouds: cirrus, cumulus, and cumulonimbus. To use AutoML Vision you need to put your training images in Google Cloud Storage.

In the GCP console, open the **Navigation menu** and select **Storage > Browser**:



Once there, you should see the bucket from the last step.

Before you add the cloud images, create an environment variable with the name of your bucket by running the following command in Cloud Shell,

replacing `YOUR_BUCKET_NAME` in the command below with the name of your bucket:

```
export BUCKET=YOUR BUCKET NAME
```

The training images are publicly available in a Cloud Storage bucket. Use the `gsutil` command line utility for Cloud Storage to copy the training images into your bucket:

```
gsutil -m cp -r gs://automl-codelab-clouds/* gs://${BUCKET}
```

When the images finish copying, click the **Refresh** button at the top of the Cloud Storage browser. Then click on your bucket name. You should see 3 folders of photos for each of the 3 different cloud types to be classified:

Hit refresh when the copy is complete, you will see something like this:

The screenshot shows the Google Cloud Platform Storage interface. On the left, there's a sidebar with options like Browser, Transfer, Transfer for on-premises, Transfer Appliance, and Settings. The main area shows 'Bucket details' for 'qwiklabs-gcp-04-d0e433d98e62-vcm'. Under the 'Objects' tab, there are three folders listed: 'cirrus/' (Subject to object ACLs), 'cumulonimbus/' (Subject to object ACLs), and 'cumulus/' (Subject to object ACLs). Below the table is a terminal window titled 'CLOUD SHELL Terminal' showing the output of a gsutil command. The command is copying files from 'gs://automl-codelab-clouds/cumulus' to 'gs://\${BUCKET}'. The terminal shows the progress of the copy operation for multiple files.

```
Copying gs://automl-codelab-clouds/cumulus/16.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/17.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/18.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/19.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/2.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/20.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/3.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/4.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/5.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/6.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/7.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/8.jpg [Content-Type=image/jpeg]...
Copying gs://automl-codelab-clouds/cumulus/9.jpg [Content-Type=image/jpeg]...
student_04_0'd748727fcf@cloudshell:~ (qwiklabs-gcp-04-d0e433d98e62)$ [1]
```

Preview any file/image by going into the file and pressing preview button.

Create a dataset

Now that your training data is in Cloud Storage, you need a way for AutoML Vision to access it. You'll create a CSV file where each row contains a URL to a training image and the associated label for that image. This CSV file has been created for you; you just need to update it with your bucket name.

Run the following command to copy the file to your Cloud Shell instance:

```
gsutil cp gs://automl-codelab-metadata/data.csv .
```

Then update the CSV with the files in your project:

```
sed -i -e "s/placeholder/${BUCKET}/g" ./data.csv
```

Now you're ready to upload this file to your Cloud Storage bucket:

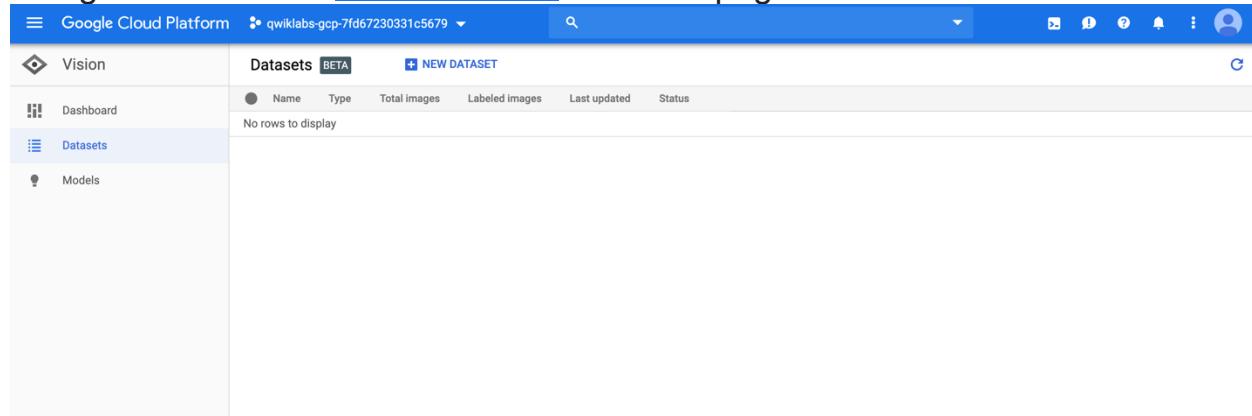
```
gsutil cp ./data.csv gs://${BUCKET}
```

Once that command completes, click the **Refresh bucket** button. Confirm that you see the `data.csv` file in your bucket.

The screenshot shows the Google Cloud Platform interface. On the left, the 'Storage' sidebar is open, showing options like 'Browser', 'Transfer', 'Transfer for on-premises', 'Transfer Appliance', and 'Settings'. The main area is titled 'Bucket details' for 'qwiklabs-gcp-04-d0e433d98e62-vcm'. It shows an 'Objects' section with a table containing one item: 'data.csv' (1.83 KB, text/csv, Regional, last modified 6/12/20, 12:43:31 AM UTC-4). Below the table are buttons for 'Upload files', 'Upload folder', 'Create folder', 'Manage holds', and 'Delete'. A 'Filter by prefix...' input field is also present. At the bottom of the objects list, there's a note: 'Buckets / qwiklabs-gcp-04-d0e433d98e62-vcm'. At the very bottom of the page, there's a 'CLOUD SHELL' tab with a terminal window. The terminal window shows the following session:

```
Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to qwiklabs-gcp-04-d0e433d98e62.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
student_04_07d748727fcf@cloudshell:~ (qwiklabs-gcp-04-d0e433d98e62)$ ls  
data.csv README-cloudshell.txt  
student_04_07d748727fcf@cloudshell:~ (qwiklabs-gcp-04-d0e433d98e62)$ sed -i -e "s/placeholder/${BUCKET}/g" ./data.csv  
student_04_07d748727fcf@cloudshell:~ (qwiklabs-gcp-04-d0e433d98e62)$ gsutil cp ./data.csv gs://${BUCKET}  
CommandException: "gsutil cp" command does not support provider-only URLs.  
student_04_07d748727fcf@cloudshell:~ (qwiklabs-gcp-04-d0e433d98e62)$ export BUCKET=qwiklabs-gcp-04-d0e433d98e62-vcm  
student_04_07d748727fcf@cloudshell:~ (qwiklabs-gcp-04-d0e433d98e62)$ gsutil cp ./data.csv gs://${BUCKET}  
Copying file://./data.csv [Content-type:text/csv]...  
[1 files] 1.8 KiB/ 1.8 KiB  
Operation completed over 1 objects/1.8 KiB.  
student_04_07d748727fcf@cloudshell:~ (qwiklabs-gcp-04-d0e433d98e62)$
```

Navigate back to the [AutoML Vision](#) Datasets page.



At the top of the console, click **+ NEW DATASET**.

Type "clouds" for the Dataset name.

Leave "Single-label Classification" checked.

Click **CREATE DATASET** to continue

Create new dataset

Dataset name *

Use letters, numbers and underscores up to 32 characters.

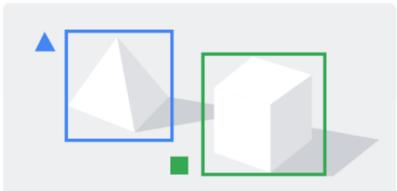
Select your model objective



Single-Label Classification
Predict the one correct label that you want assigned to an image.



Multi-Label Classification
Predict all the correct labels that you want assigned to an image.



Object detection
Predict all the locations of objects that you're interested in.

[CANCEL](#) [CREATE DATASET](#)

On the next screen you will choose the location of your training images (the ones you uploaded in the previous step)

Choose **Select a CSV file on Cloud Storage** and add the file name to the URL for the file you just uploaded - `gs://your-project-name-vcm/data.csv`. You may also use the browse function to find the csv file. Once you see the white in green checkbox you may select **CONTINUE** to proceed.

The screenshot shows the Google Cloud Platform Vision API interface. The left sidebar has 'Vision' selected. The main area is titled 'Select files to import'. It explains that to build a custom model, images must be imported and categorized with labels. It lists two options: 'Upload images from your computer' (radio button) and 'Select a CSV file on Cloud Storage' (radio button, which is selected). Below this, there's a section titled 'Select a CSV file on Cloud Storage' with instructions to upload GCS paths to images in CSV format. A sample CSV format is shown:

```
[set,]image_path[,label]  
TRAIN,gs://My_Bucket/sample1.jpg,cat  
TEST,gs://My_Bucket/sample2.jpg,dog
```

The input field contains the URL `gs://* <your-project-name>-vcm/data.csv`, with a checked checkbox preceding it. There is also a 'BROWSE' button. At the bottom, there are 'CONTINUE' and 'BACK' buttons.

After you are returned to the **IMPORT** tab, navigate to the **IMAGES** tab. It will take **8 to 12 minutes** while the image metadata is processed ("Running: Importing Images" will appear on the screen). Once complete, the images will appear by category.

Inspect images

Next proceed with a brief examination of the images.

Screenshot of the Google Cloud Platform Vision interface showing a dataset for clouds. The interface includes a sidebar with 'Vision', 'Dashboard', 'Datasets' (selected), and 'Models'. The main area shows 'IMPORT' and 'IMAGES' tabs, with 'IMAGES' selected. It displays statistics: All images (60), Labeled (60), Unlabeled (0). A 'Filter labels' section lists 'cirrus' (20), 'cumulonimbus' (20), and 'cumulus' (20). An 'ADD NEW LABEL' button is also present. To the right, there are sections for 'LABEL STATS' and 'EXPORT DATA'. Below these are buttons for 'TRAIN', 'EVALUATE', and 'TEST & USE'. A 'Filter images' section allows filtering by labels. On the right, a grid of six images is shown, each labeled with its category and count: 'cirrus(1)', 'cumulus(1)', 'cirrus(1)', 'cumulus(1)', 'cumulonimbus(1)', and 'cumulus(1)'.

Try filtering by different labels in the left menu (i.e. click cumulus) to review the training images:

≡ Type to filter... ⋮

cirrus 20

cumulonimbus 20

cumulus 20

[Add label](#)

Note: If you were building a production model, you'd want *at least* 100 images per label to ensure high accuracy. This is just a demo so only 20 images of each type were used so the model could train quickly.

If any images are labeled incorrectly you can click on them to switch the label or delete the image from your training set:

✖ Image 2 of 50 🔍 🔎

≡ |Filter labels

cirrus
 cumulonimbus
 cumulus

 Unlabeled



To see a summary of how many images you have for each label, click on **LABEL STATS**. You should see the following pop-out box show up on the right side of your browser. Press **DONE** after reviewing the list.

Label Stats

Unlabeled images aren't used. Your dataset will be automatically split into [Train, Validation and Test sets](#).

Ideally, each label should have at least **10 images**. Fewer images often result in inaccurate precision and recall. You must also have at least **8, 1, 1 images** each assigned to your Train, Validation and Test sets.

Labels	Images	Train	Validation	Test
cirrus	 20	16	2	2
cumulonimbus	 20	16	2	2
cumulus	 20	16	2	2

[DONE](#)

Note: If you are working with a dataset that isn't already labeled, AutoML Vision provides an in-house human labeling service.

Train your model

You're ready to start training your model! AutoML Vision handles this for you automatically, without requiring you to write any of the model code.

To train your clouds model, go to the **TRAIN** tab and click **START TRAINING**.

← clouds **BETA**

LABEL STATS

EXPORT DATA

IMPORT

IMAGES

TRAIN

EVALUATE

TEST & USE

Try labeling more images before training

Unlabeled images aren't used. Your dataset will be automatically split into [Train, Validation, and Test sets](#).

Ideally, each label should have at least **10 images**. Fewer images often result in inaccurate precision and recall. You must have at least **8, 1, 1 images** each assigned to your Train, Validation and Test sets.

Labels	Images	Train	Validation	Test
cirrus	<div style="width: 16px;"></div> 20	16	2	2
cumulonimbus	<div style="width: 16px;"></div> 20	16	2	2
cumulus	<div style="width: 16px;"></div> 20	16	2	2

START TRAINING

Enter a name for your model, or use the default auto-generated name.

Leave **Cloud hosted** selected and click **CONTINUE**.

Train new model

1 Define your model

Model name *

clouds_20191017015319

Cloud hosted

Host your model on Google Cloud for online predictions

Edge

Download your model for offline/mobile use

[CONTINUE](#)

2 Set a node hour budget

[START TRAINING](#)

[CANCEL](#)

For the next step, type the value "8" into the **Set your budget** box and check "**Deploy model to 1 node after training.**" This process (auto-deploy) will make your model immediately available for predictions after testing is complete. Click **START TRAINING**.

Train new model

Define your model

2 Set a node hour budget

Specify the maximum number of node hours to spend training your model. If your model stops improving before then, AutoML Vision will stop training and you'll only be charged for the actual node hours used.

For cloud-hosted model. You can train for 40 node hours (per billing account) for free. Standard pricing applies afterwards. [Pricing guide](#)



Note to beta users: AutoML Vision has updated its pricing for node hours.

Budget *

8

maximum node hours



16 node hours is recommended for this dataset

Estimated completion time: 2 hours

Deploy model to 1 node after training

Make your model available for REST API requests immediately after training.

Deployment pricing applies.

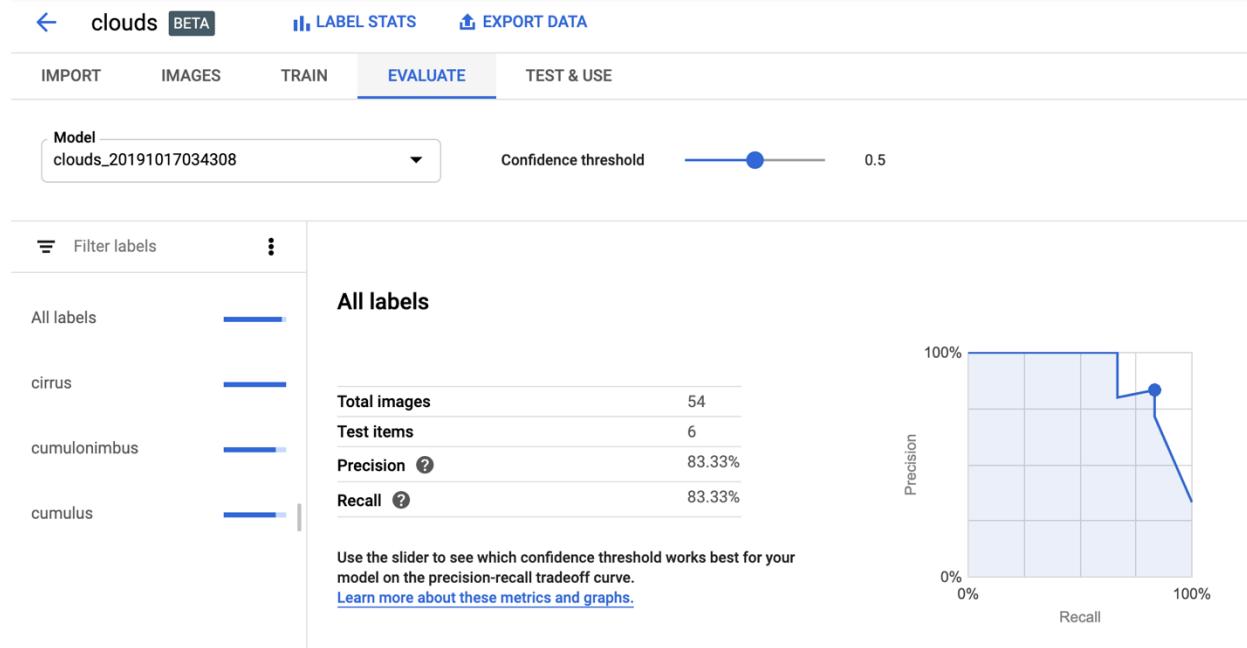
START TRAINING

CANCEL

Training this dataset will take around **55 to 90 minutes** to complete. The total training time includes node training time as well as infrastructure set up and tear down.

Evaluate your model

After training is complete, click on the **EVALUATE** tab. Here you'll see information about Precision and Recall of the model. It should resemble the following:



You can also adjust the **Confidence threshold** slider to see its impact. Finally, scroll down to take a look at the **Confusion matrix**.

Confusion matrix

True Label	Predicted Label		
	cumulus	cumulonimbus	cirrus
cumulus	50%	50%	-
cumulonimbus	-	100%	-
cirrus	-	-	100%

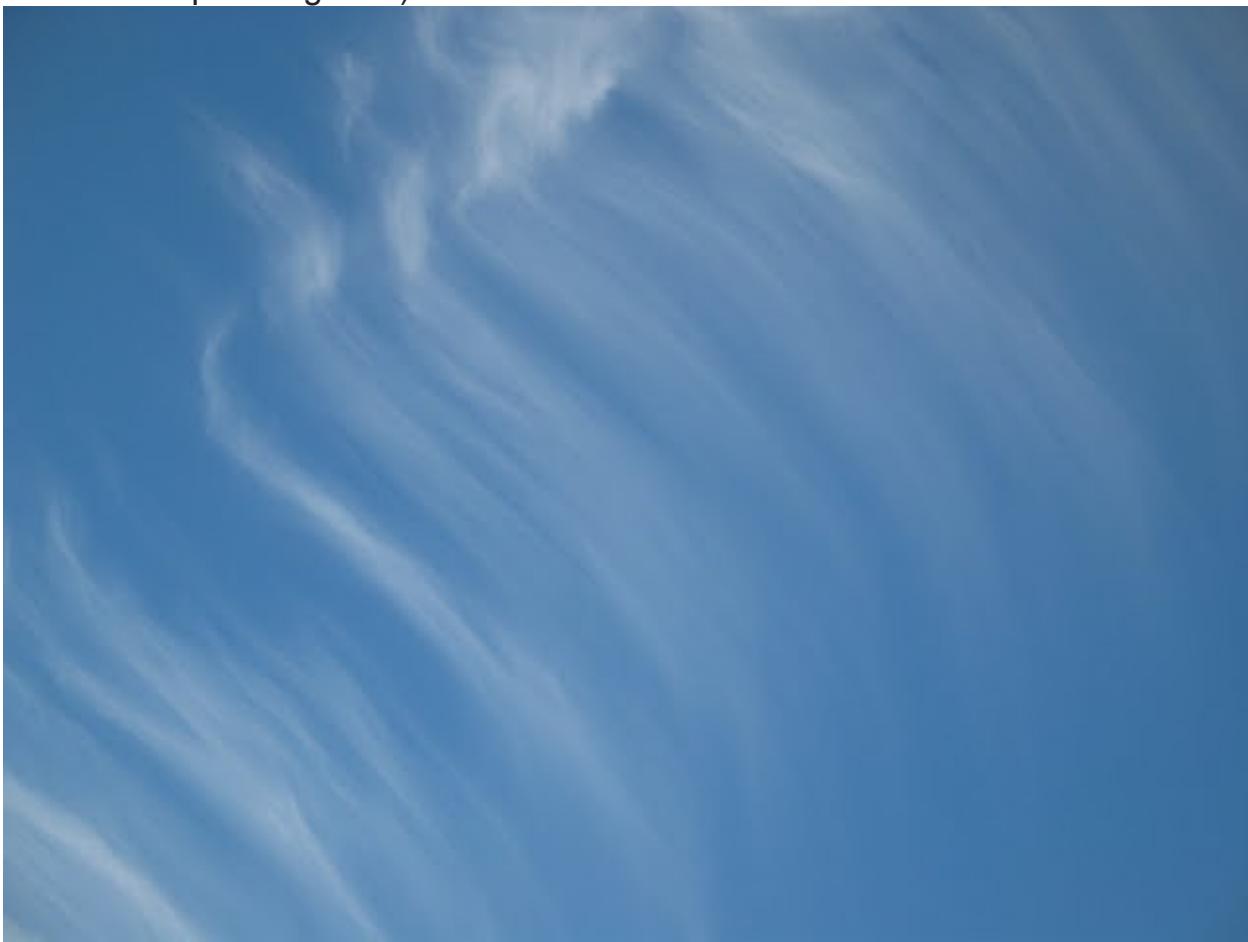
This tab provides some common machine learning metrics to evaluate your model accuracy and see where you can improve your training data. Since the focus for this lab was not on accuracy, move on to the next section about predictions section. Feel free to browse the accuracy metrics on your own.

Generate predictions

Now it's time for the most important part: generating predictions on your trained model using data it hasn't seen before.

There are a few ways to generate predictions. In this lab you'll use the UI to upload images. You'll see how your model does classifying these two images (the first is a cirrus cloud, the second is a cumulonimbus).

First, download these images to your local machine by right-clicking on each of them (**Note:** You may want to assign a simple name like 'Image1' and 'Image2' to assist with uploading later):





Navigate to the **TEST & USE** tab in the AutoML UI:

On this page you will see that the model you just trained and deployed is listed in the "Model" pick list.

Click **UPLOAD IMAGES** and upload the cloud sample images you just saved to your local disk (you may select both images at the same time).

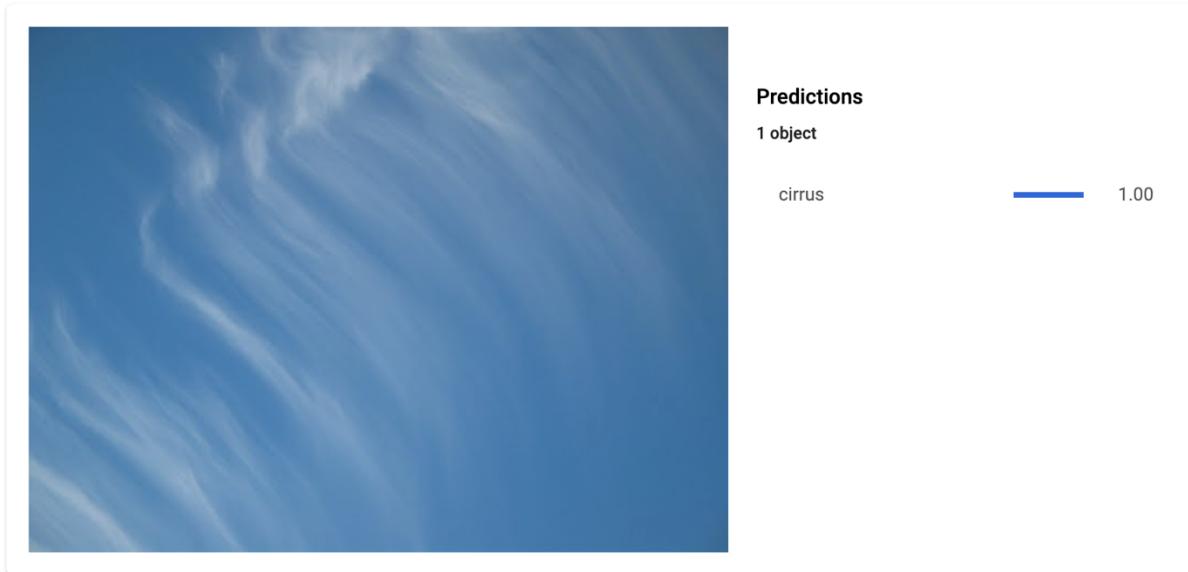
Screenshot of the 'TEST & USE' section of a machine learning model interface.

The interface includes:

- Cloud icon in the top-left corner.
- Back arrow and 'clouds' text in the top-left.
- 'BETA' button in the top-right.
- 'LABEL STATS' and 'EXPORT DATA' buttons in the top-right.
- Navigation tabs: IMPORT, IMAGES, TRAIN, EVALUATE, TEST & USE (highlighted).
- A sidebar with icons for IMPORT, EXPORT, and TEST & USE.
- A dropdown menu labeled 'Model' showing 'clouds_20191022095815'.
- A message box indicating the model is deployed: "Your model is deployed and is available for online prediction requests. [Learn more](#)".
- A section titled 'Test your model' with a blue 'UPLOAD IMAGES' button.
- A note below the button: "Up to 10 images can be uploaded at a time".

The 'Model' dropdown and the 'UPLOAD IMAGES' button are circled in red.

When the prediction request completes you should see something like the following:





Predictions

1 object

cumulonimbus

 1.00