Hands-On With Google Cloud AutoML

Building High-Quality Custom ML Models With Minimal Effort / **Nitya Narasimhan**

Who am I?

- PhD (distributed systems) & Polyglot (mobile web)
- Google Developer Expert (flutter) & Google Developer Group Organizer (New York City & Hudson Valley)
- Cloud DevRel PM @Microsoft (since Oct)
- Machine learning enthusiast & beginner passionate about making complex ideas accessible to all.

Introduction to Auto ML

What is Cloud AutoML
Learning2Learn
Transfer Learning
Google’s Machine Learning Ecosystem

Getting Setup for Cloud AutoML

25 mins

What is AutoML ?

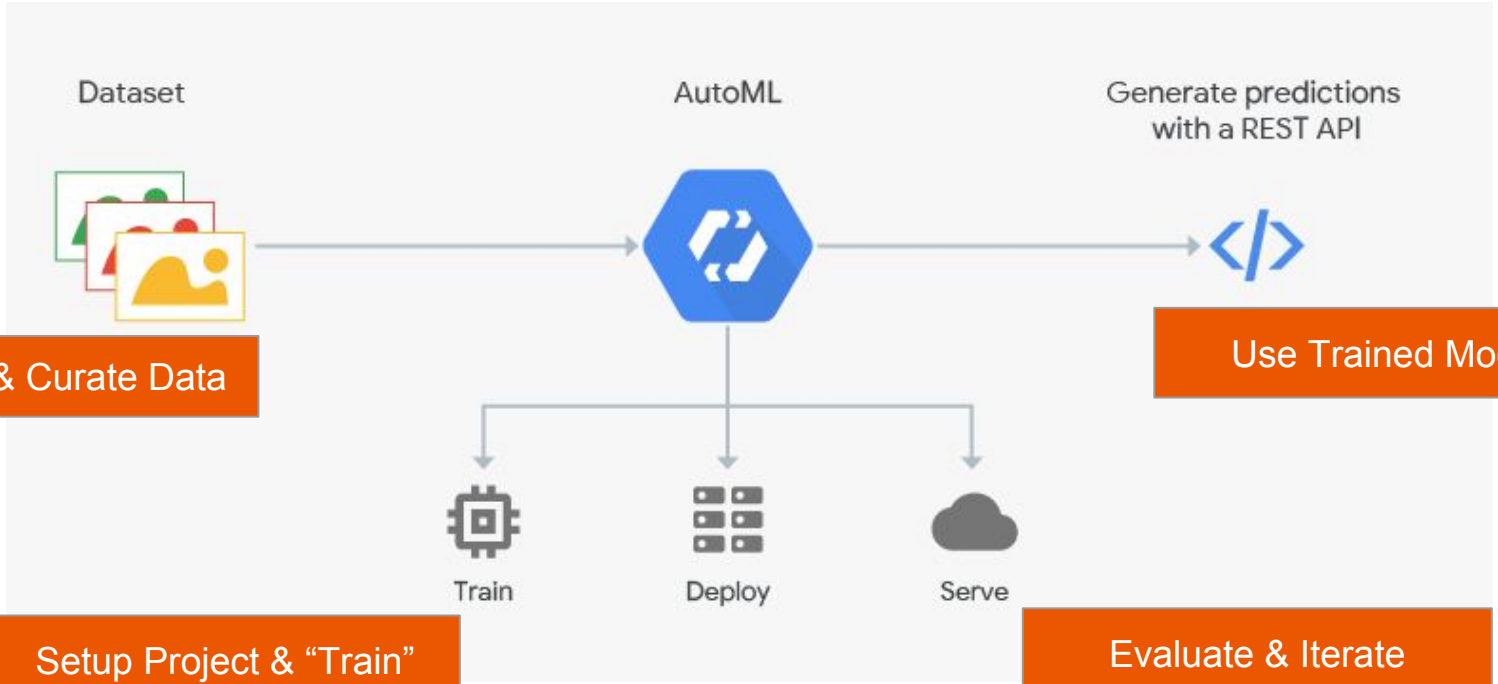
- (fast.ai) describes automated methods for *model selection* and/or *hyperparameter optimization*
- (developer) an auto-magical process that helps me build custom models from domain-specific data *without requiring machine learning expertise*
- many AutoML libraries exist; more being added every day (e.g, H20 AutoML, Azure etc.)

What does Google AutoML Provide?

- **Seamless integration** with Google Cloud products for effortless workflows (e.g., Storage, Cloud ML APIs)
- **Custom models** for your domain-specific data or requirements (beyond pre-trained “API” services)
- **Human labeling** support (only AutoML Vision) for high-quality privacy-preserving training datasets

Image Credit: https://cloud.google.com/automl/

How AutoML Works



Collect & Curate Data

Use Trained Model

Setup Project & “Train”

Evaluate & Iterate

Challenges for Deep Learning Today

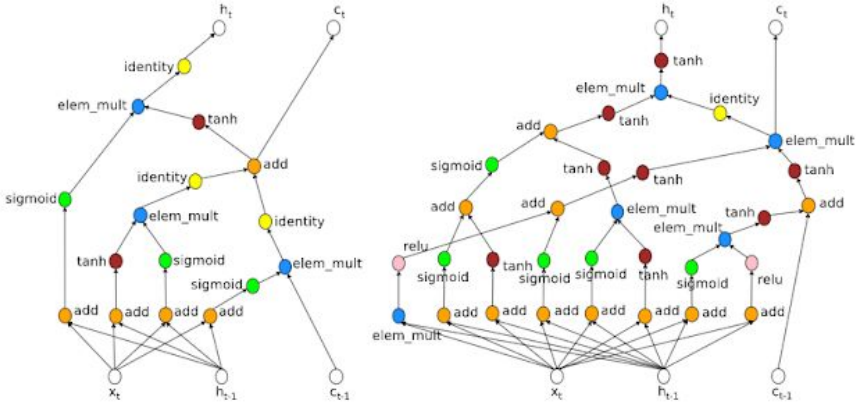
- **Large-scale Datasets.** How can I craft accurate models if I have insufficient data to start from?
- **Computational Resources.** Creating accurate models requires computing power & expertise
- **Art vs. Science.** Humans-in-the-loop succeed often by trial-and-error, which requires non-trivial time

The Transfer Learning Approach

- **Core idea.** Take advantage of pre-trained models built with larger datasets and resources, begin with those existing weights, and retrain key layers to customize it for your needs.
- **Underlying Idea.** Neural nets generalize for *similar* types of problems (that have common underlying features) and is ideal when datasets are marginally custom to faciliate “transfer” of learning.

The Neural Architecture Search Approach

Core idea. Assumes each dataset is unique and tries to compute weights from scratch. Use computational power to *search & discover* neural net architectures that are beyond human cognition.



Google's AutoML Products / currently in beta

- **AutoML Vision.** Classify your domain image dataset using custom labels. Get human labeling support. Register trained model for access via AutoML API.
- **AutoML Natural Language.** Classify *content* (text) into a set of custom categories (labels).
- **AutoML Translation.** Custom *language* translation (text) models for more domain-specific query results

Google's ML Ecosytem / big picture

- **TensorFlow.** Open source software machine learning framework. Used by data scientists. Custom models.
- **Cloud AI "API" Building Blocks.** Pre-trained models, REST API & AutoML. Focus on easy integration and quick wins for developers with limited ML expertise.
- **Firebase MLKit.** Mobile-focused SDK that brings cloud ML expertise into Android & iOS workflows. Plus ability to store and deploy custom TensorFlow Lite models.

Getting Ready:

Google Cloud Setup for AutoML

Google Cloud SDK Setup

- Quickstarts available for Linux, MacOS, Windows
- Quickstart Process
  - Install Python (v 2.7 recommended)
  - Install Google Cloud SDK
  - Run `./google-cloud-sdk/install.sh`
  - New terminal, verify `./gcloud --version`

```
Google Cloud SDK
221.0.0
bq 2.0.35
core 2018.10.12
gsutil 4.34
```

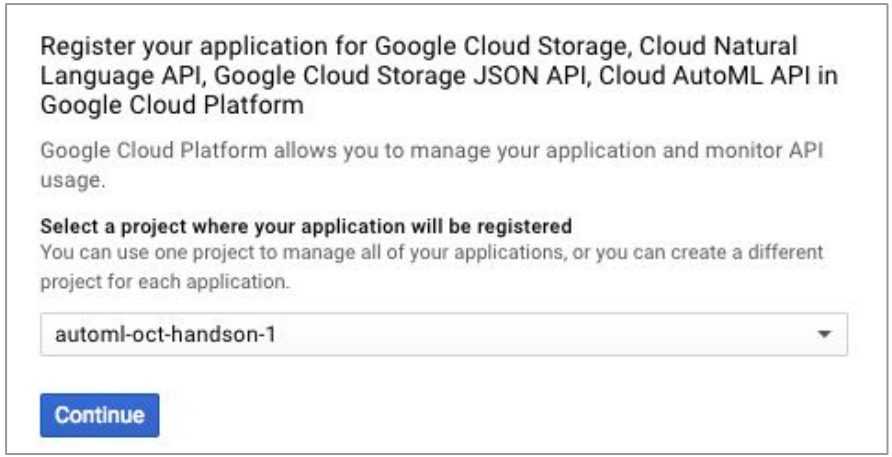
Google Cloud Init

- Run: `gcloud init`
  - Log into valid gmail account when prompted
  - Use default configuration if new
  - Create a new project if new

    (or pick one with Owner, Editor or View permissions)
- Verify init succeeded: `gcloud config list`
- Get help anytime: `gcloud help`

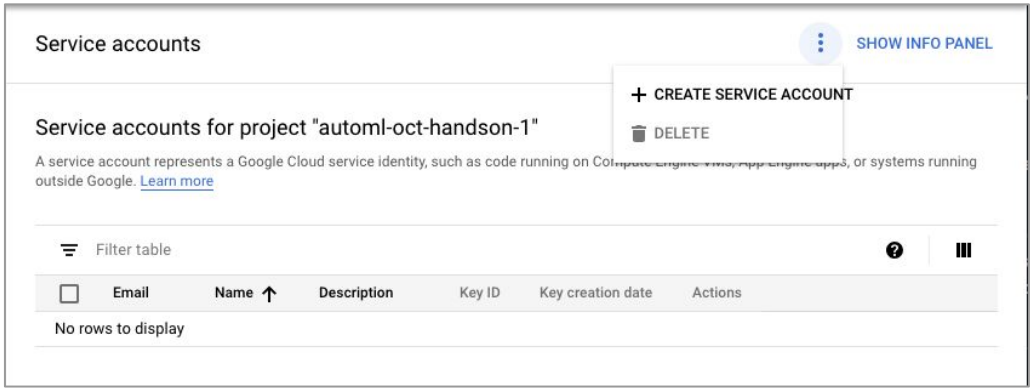
Google Cloud Console Setup

- GCP Console: Create or Select a Project
- Enable Billing: New users can use $300 credit
- Enable the AutoML APIs for Project
  - Natural Language
  - Translate
  - Vision



Google Cloud: Create Service Account & Key

- Like adding a user to your project, but representing the app
- Create key, store to filename in local filesystem
- `export GOOGLE_APPLICATION_CREDENTIALS=<filename>`



Install the Google Cloud Client Libraries

- Provides API-specific library implementations for programmatic access to Cloud Platform APIs (including AutoML)
- Client libraries available for Java, Python, Node.JS
- I used the Node.js client

  ```
  npm install --save @google-cloud/automl
  ```

- Example Usage: Crowdsourcing Training Data

Useful if you want to work with the AutoML services programmatically (vs. using the Dashboard)

Setup environment variables for convenience

- `export PROJECT_ID=”your-project-id”`
- `export REGION_NAME=”us-central1”`

Add IAM Policy Bindings

- IAM = Identity & Access Management
- Give service account **AutoML Editor** IAM role for project
- Then allow “AutoML” service accounts access to project as needed

Service Accounts: AutoML NL access

Allow the AutoML Natural Language service accounts to access your Google Cloud project resources:

```
gcloud projects add-iam-policy-binding project-id \
  --member="serviceAccount:custom-natural-language@appspot.gserviceaccount.com" \
  --role="roles/ml.admin"
```

```
gcloud projects add-iam-policy-binding project-id \
  --member="serviceAccount:custom-natural-language@appspot.gserviceaccount.com" \
  --role="roles/storage.admin"
```

Storage Bucket (lcm): NL datasets

```
gsutil mb -p project-id -c regional -l us-central1 gs://project-id-lcm/
```

Service Accounts: AutoML Translate access

3. Allow the AutoML Translation service accounts to access your Google Cloud project resources:

```
gcloud projects add-iam-policy-binding project-id \
  --member="serviceAccount:custom-translation@appspot.gserviceaccount.com" \
  --role="roles/ml.admin"
```

```
gcloud projects add-iam-policy-binding project-id \
  --member="serviceAccount:custom-translation@appspot.gserviceaccount.com" \
  --role="roles/storage.admin"
```

Storage Bucket (vcm): Translate datasets

```
gsutil mb -p project-id -c regional -l us-central1 gs://project-id-vcm/
```

Service Accounts: AutoML Vision access

. Allow the AutoML Vision service accounts to access your Google Cloud project resources:

```
gcloud projects add-iam-policy-binding project-id \
  --member="serviceAccount:custom-vision@appspot.gserviceaccount.com" \
  --role="roles/ml.admin"
```

```
gcloud projects add-iam-policy-binding project-id \
  --member="serviceAccount:custom-vision@appspot.gserviceaccount.com" \
  --role="roles/storage.admin"
```

Storage Bucket (vcm): Vision datasets

```
gsutil mb -p project-id -c regional -l us-central1 gs://project-id-vcm/
```

Desktop CLI vs. Google Cloud Shell

- **Google Cloud CLI** allows for desktop-driven access, configuration and execution of various Google Cloud interactions. It can be complex and vast in scope. *Install & Explore for experience.*
- **Google Cloud Shell** allows for browser-driven console-integrated experience that feels easier for beginners. *We'll explore this when doing AutoML walkthroughs later.*

Short Break

Questions?

5 mins

AutoML Vision

Label Images Accurately
Vision API
AutoML Vision
Data Preparation / Model Testing / Model Evaluation

45 mins

Cloud ML & Vision: Bringing sight

Cloud Vision API

Image Recognition & Classification using pre-trained models for easy integration

Cloud Video Intelligence API

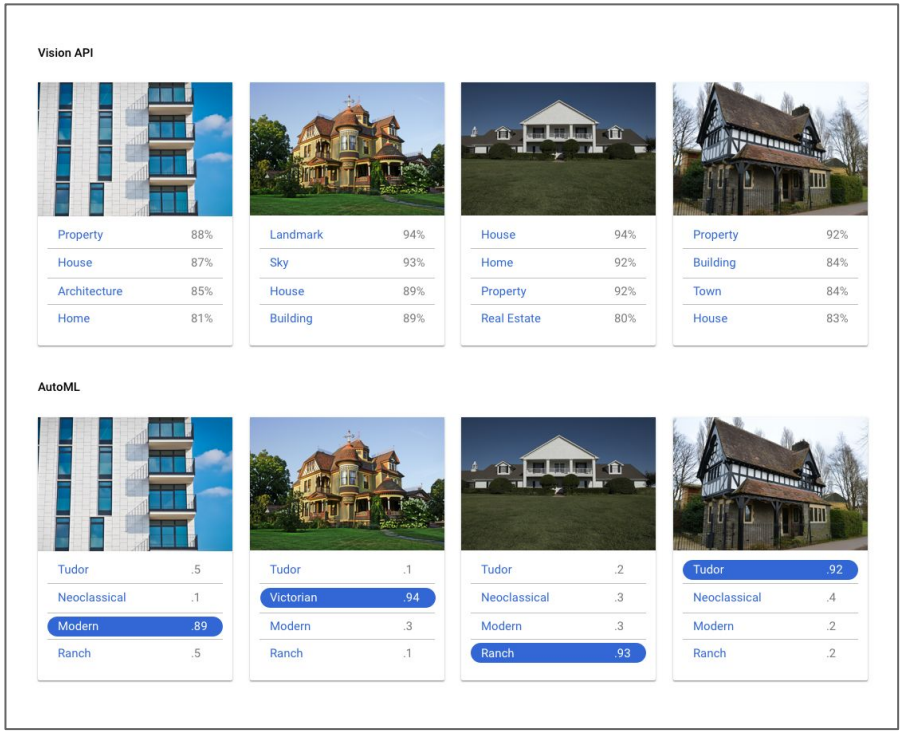
Scene-level Video Annotation to facilitate rich search & discovery

AutoML Vision

Custom image classification models

Image Credit: Google

What is the Vision API?



- Image labeling
- Face detection
- Landmark detection
- OCR recognition
- Explicit content
- + API & Libraries

Cloud Vision API Pricing

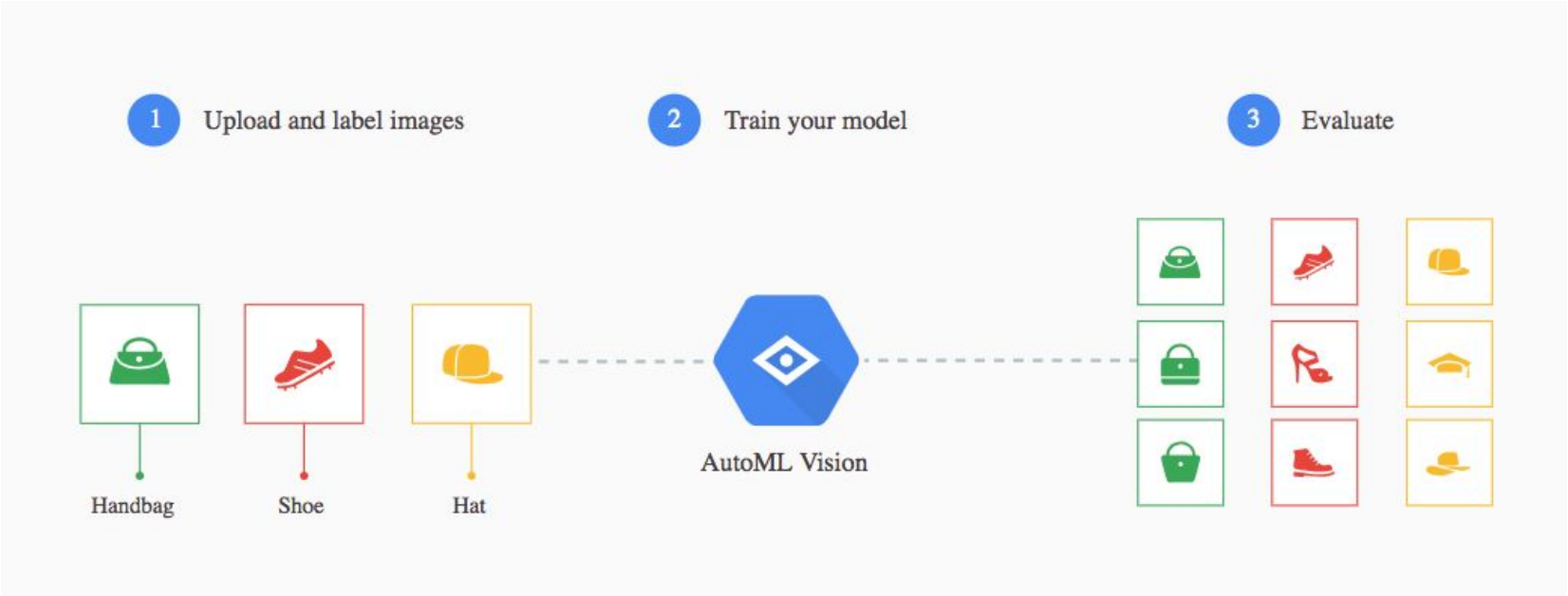
FEATURE	PRICE PER 1,000 UNITS, BY MONTHLY USAGE		
	1–1,000 UNITS/MONTH	1001–5,000,000 UNITS/MONTH	5,000,001–20,000,000 UNITS/MONTH
Label Detection	Free	$1.50	$1.00
Text Detection	Free	$1.50	$0.60
Safe Search (explicit content) Detection	Free	Free with Label Detection, or $1.50	Free with Label Detection, or $0.60
Facial Detection	Free	$1.50	$0.60
Landmark Detection	Free	$1.50	$0.60
Logo Detection	Free	$1.50	$0.60
Image Properties	Free	$1.50	$0.60
Crop Hints	Free	Free with Image Properties, or $1.50	Free with Image Properties, or $0.60
Web Detection	Free	$3.50	Contact Google for more information
Document Text Detection	Free	$1.50	$0.60
Object Localizer	Free	$2.25	$1.50

- Generous free tier for exploration
- Use Cases
  - Image Search
  - Document classification
  - Product Search

Cloud Vision API Interactive Demo

https://cloud.google.com/vision/

What is AutoML Vision?



AutoML Vision Pricing

AutoML Vision pricing is based on Training and Prediction. The accuracy of your model generally depends on how long you allow it to train and the quality of your training dataset. You will pay only for the compute hours used.

TRAINING
For training, you get one hour of free training per model for the first 10 models each month. Subsequent training hours are USD$20 per hour. Many customers find that one hour is sufficient to build an experimental model and use additional training hours to increase accuracy to production level.

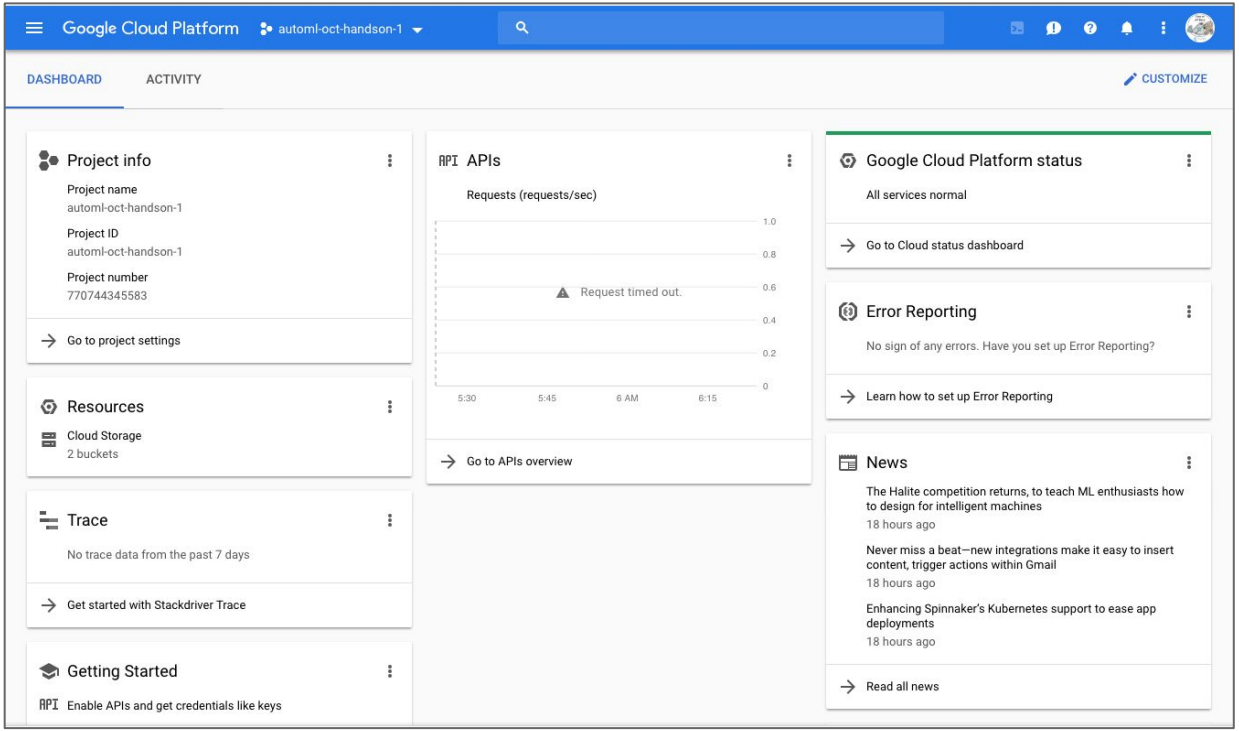
PREDICTION	
1–1,000 images	Free
1,001–5,000,000 images*	$3 per 1,000 images

AutoML Vision Training Example

A. AutoML Vision Codelab

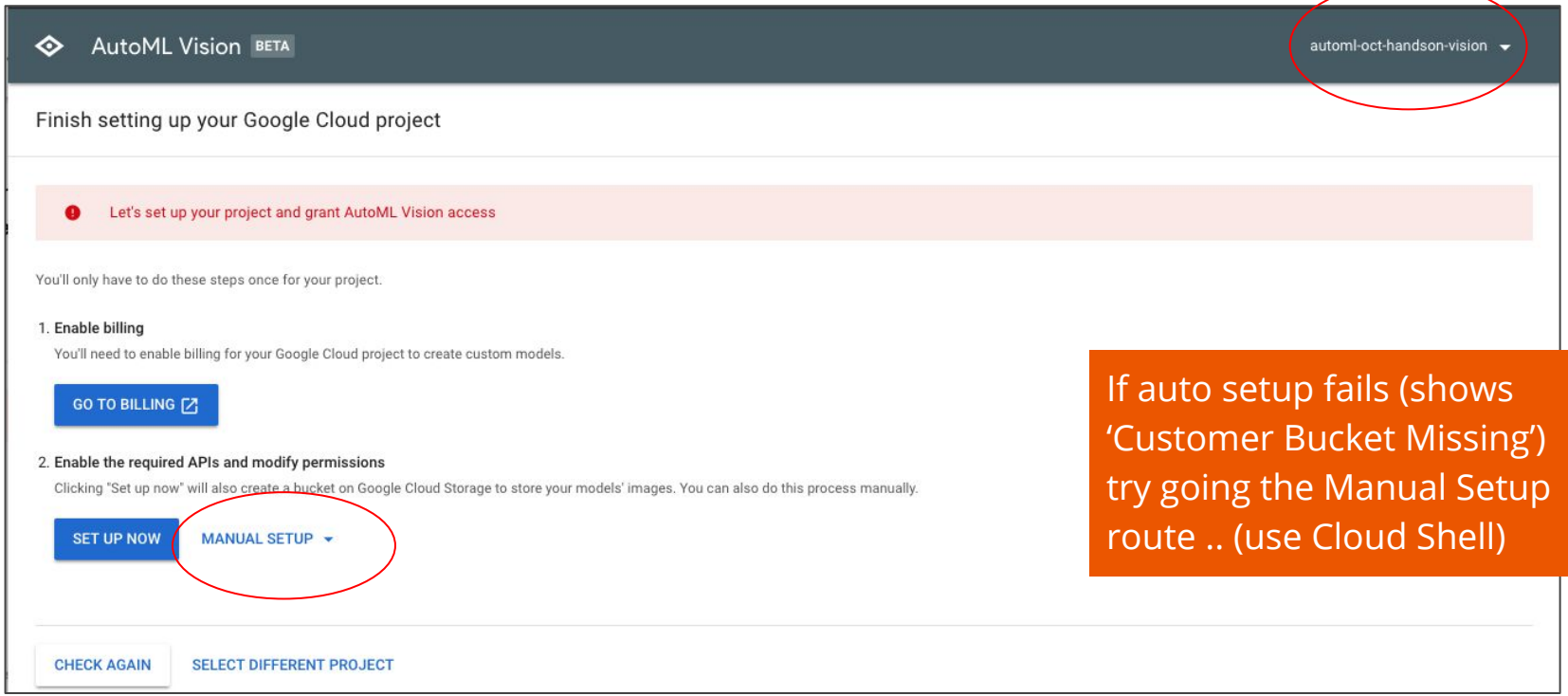
- Follow Clouds Codelab from Google Developers
- Step 1: Create Google account, sign into Cloud Console
- Step 2: Activate Cloud Shell for Project
- Step 3: Train model using dataset
- Step 4: Evaluate trained model (get precision, recall)
- Step 5: Use trained model to make predictions
- (Delete model if not in use)

A.1 Codelab: Create account & project



A.2 Codelab: Activate Shell

A.3 Codelab: Enable AutoML Vision



If auto setup fails (shows 'Customer Bucket Missing') try going the Manual Setup route .. (use Cloud Shell)

A.3 Codelab: Manual Setup

a. **Manually enable the APIs by clicking this link:**

```
https://console.cloud.google.com/flows/enableapi?project=automl-oct-handson-vision&apiid=st
```

A.3 Codelab: Manual Setup

b. Launch Google Cloud Shell and run the following commands

Or, if you installed the Cloud SDK, you can paste them into your command line

```
gcloud projects add-iam-policy-binding automl-oct-handson-vision \
    --member="user:flutterfyi@gmail.com" \
    --role="roles/automl.admin"
```

```
gcloud projects add-iam-policy-binding automl-oct-handson-vision \
    --member="serviceAccount:custom-vision@appspot.gserviceaccount.com" \
    --role="roles/ml.admin"
```

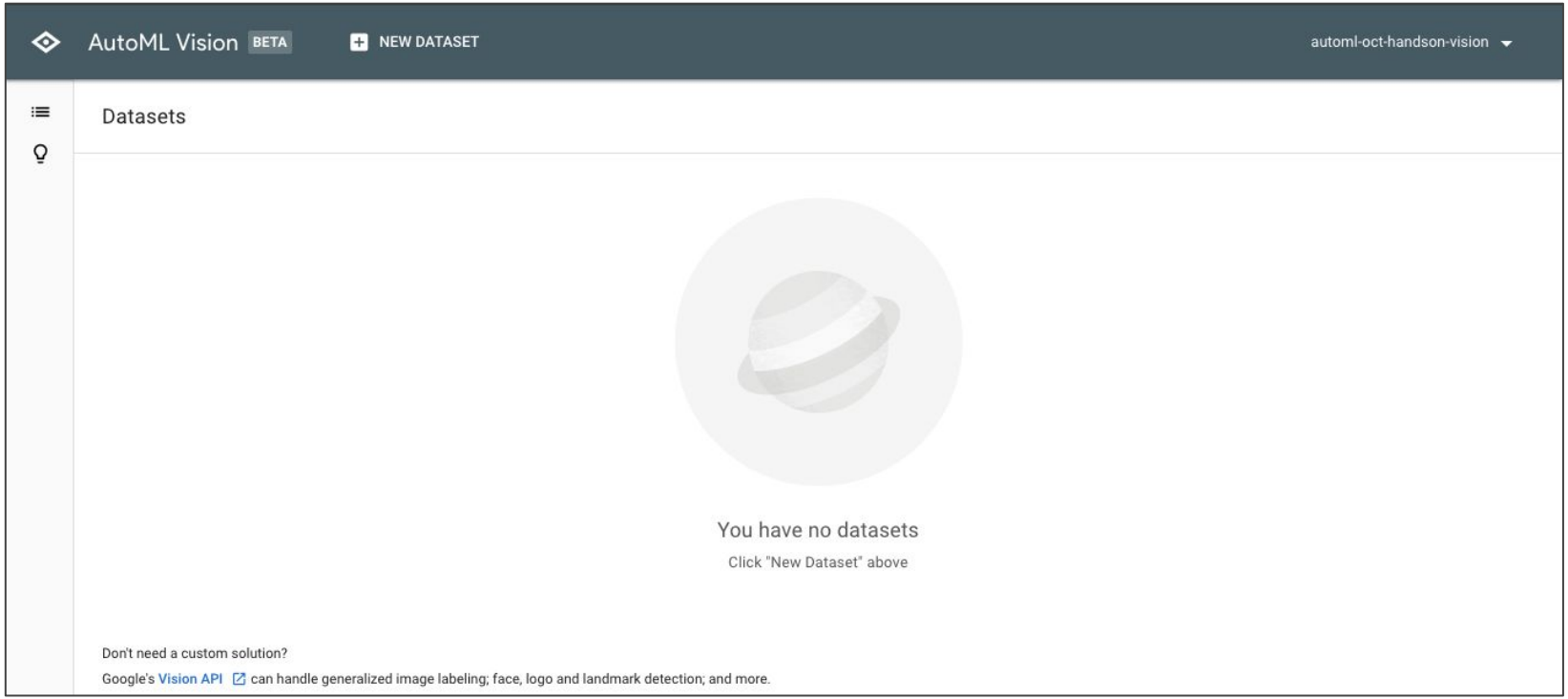
```
gcloud projects add-iam-policy-binding automl-oct-handson-vision \
    --member="serviceAccount:custom-vision@appspot.gserviceaccount.com" \
    --role="roles/storage.admin"
```

A.3 Codelab: Manual Setup

c. **Create a Google Cloud Storage bucket for storing your images**

```
gsutil mb -p automl-oct-handson-vision \
    -c regional    \
    -l us-central1 \
    gs://automl-oct-handson-vision-vcm/
```

A.3 Codelab: Manual Setup Success



A.4 Codelab: Upload training images to Storage

In Cloud Shell:

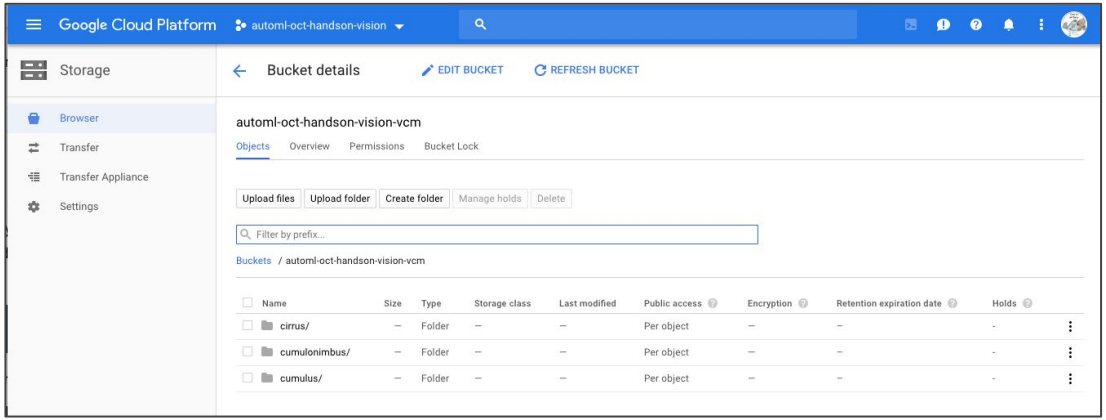
`export BUCKET=<created-bucket-name>`

Copies images dataset directly from source (storage) to project storage

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

In Console:

Hit Refresh



A.5 Codelab: Create dataset (image+labels CSV)

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

```
gs://placeholder/cirrus/8.jpg,cirrus
gs://placeholder/cirrus/9.jpg,cirrus
gs://placeholder/cumulonimbus/1.jpg,cumulonimbus
gs://placeholder/cumulonimbus/10.jpg,cumulonimbus
```

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

```
gs://<mybucketname>/cirrus/8.jpg,cirrus
gs://<mybucketname>/cirrus/9.jpg,cirrus
gs://<mybucketname>/cumulonimbus/1.jpg,cumulonimbus
gs://<mybucketname>/cumulonimbus/10.jpg,cumulonimbus
```

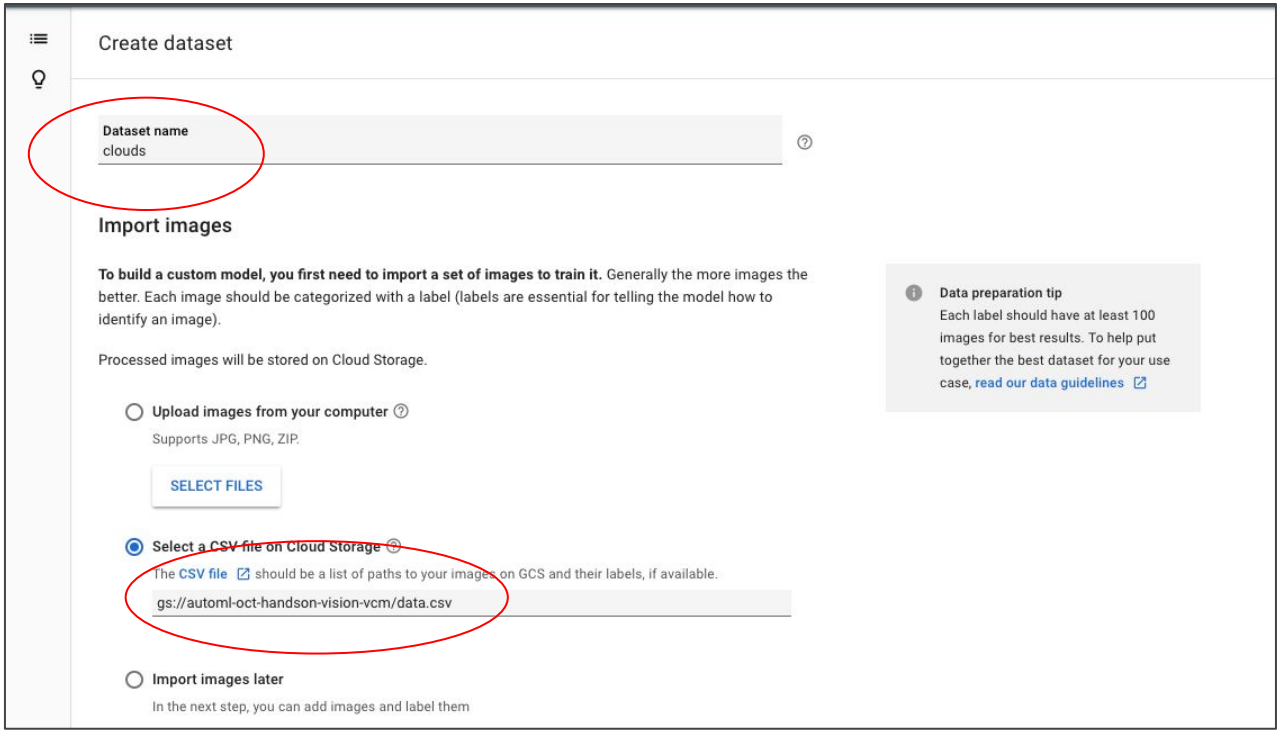
A.5 Codelab: Upload updated CSV to Storage.

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

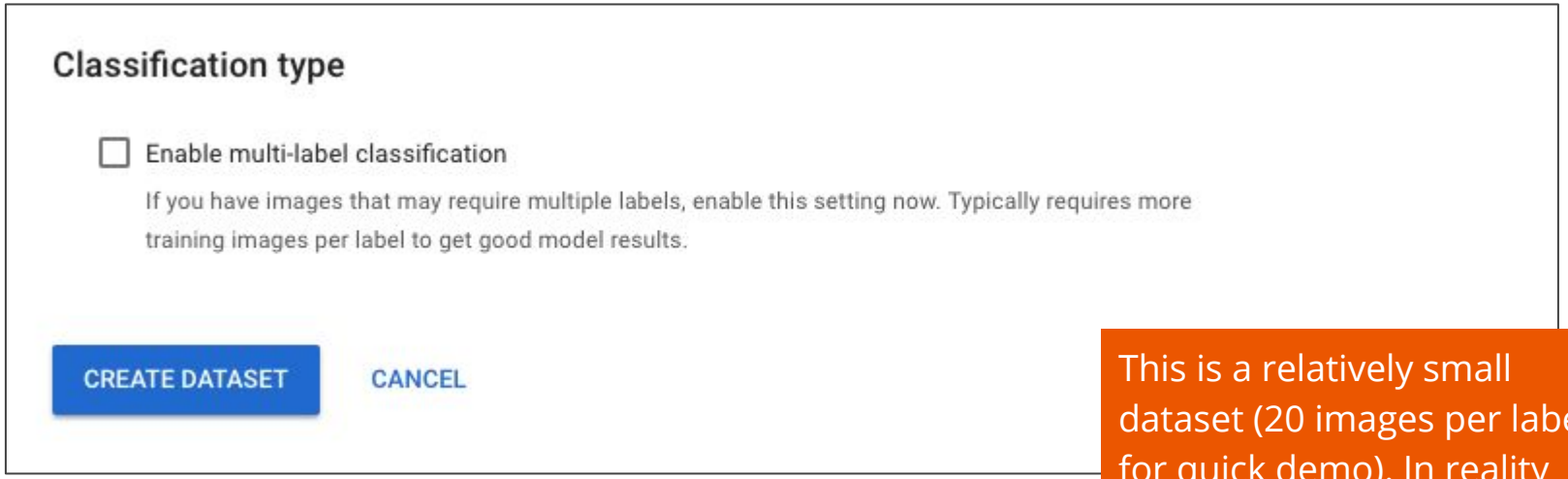
Buckets / automl-oct-handson-vision-vcm

Name	Size	Type	Storage class	Last modified	Public access	Encryption	Retention expiration date	Holds
cirrus/	–	Folder	–	–	Per object	–	–	-
cumulonimbus/	–	Folder	–	–	Per object	–	–	-
cumulus/	–	Folder	–	–	Per object	–	–	-
data.csv	3.53 KB	text/csv	Regional	10/19/18, 7:26 AM	Not public	Google-managed key	–	None

A.6 Codelab: Create dataset

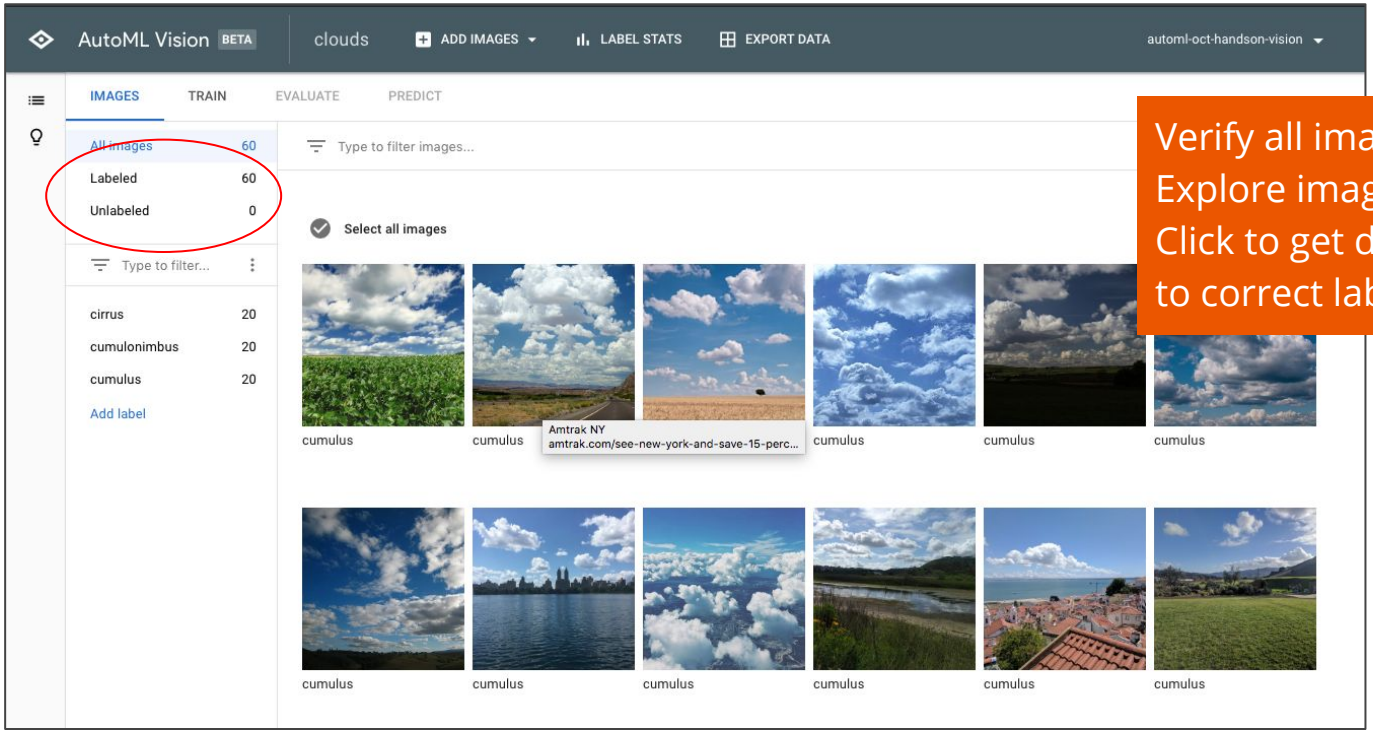


A.6 Codelab: Create dataset



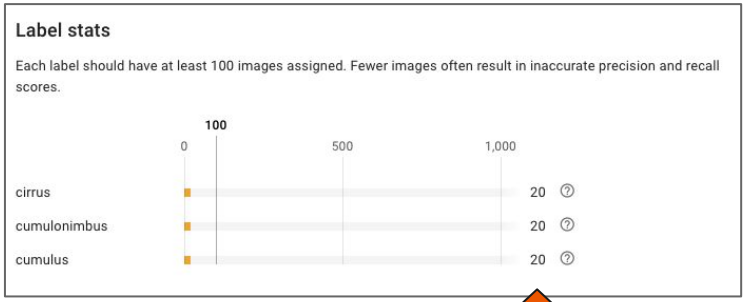
This is a relatively small dataset (20 images per label for quick demo). In reality you need 100s and 1000s.

A.7 Codelab: Inspect dataset (images)



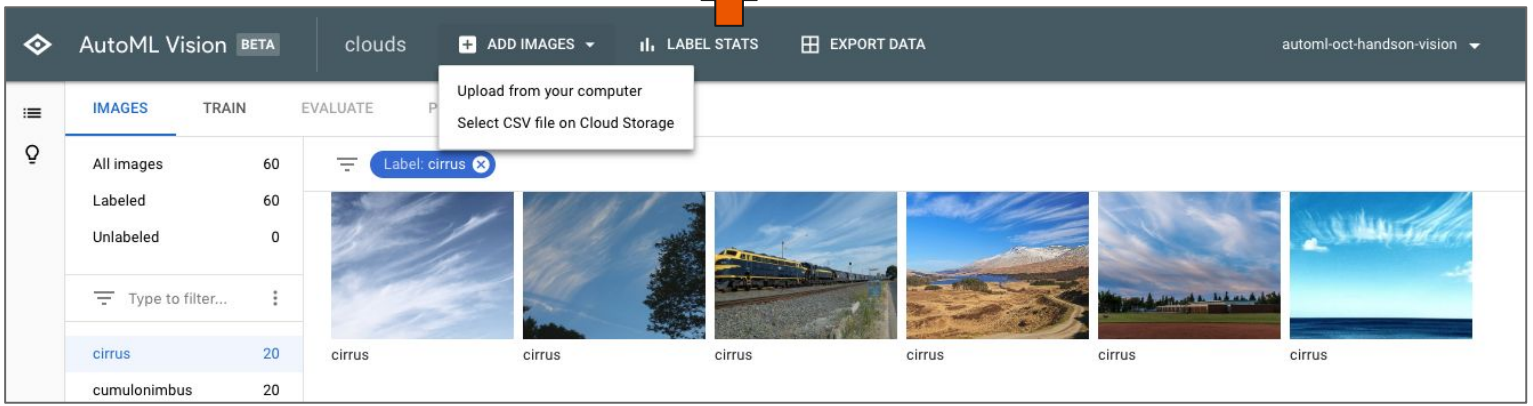
Verify all images have labels.
Explore images using filters.
Click to get details view (e.g., to correct labels)

A.7 Codelab: Inspect stats, add images/labels

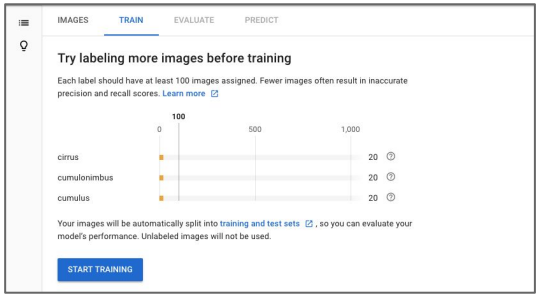


Correct the dataset as needed. You will need to retrain the model ..

Free* **Human Labeling** support (iff 2-20 labels & 100+ unlabeled images)

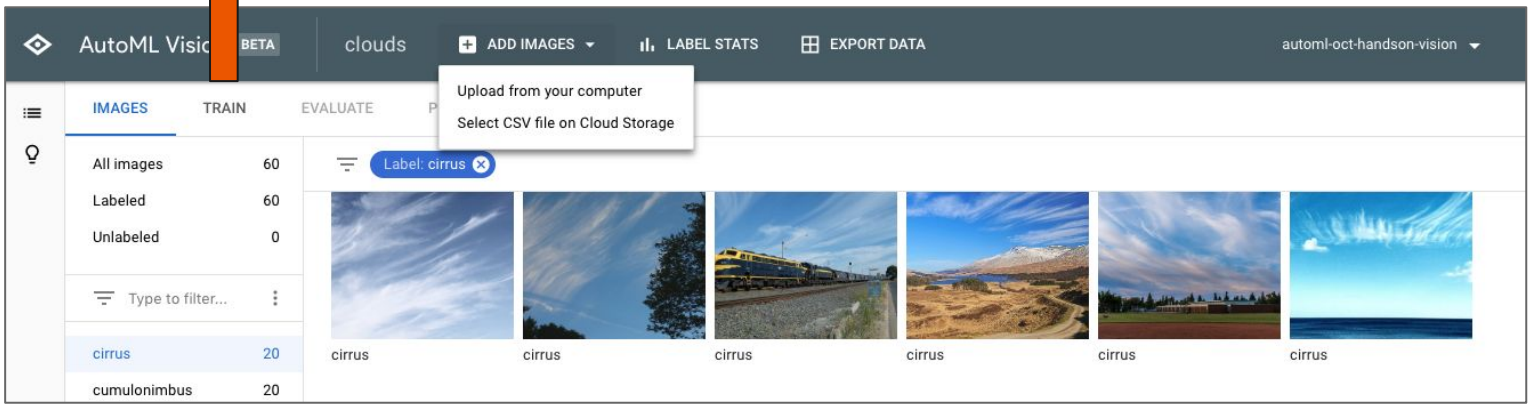


A.8 Codelab: Train your model

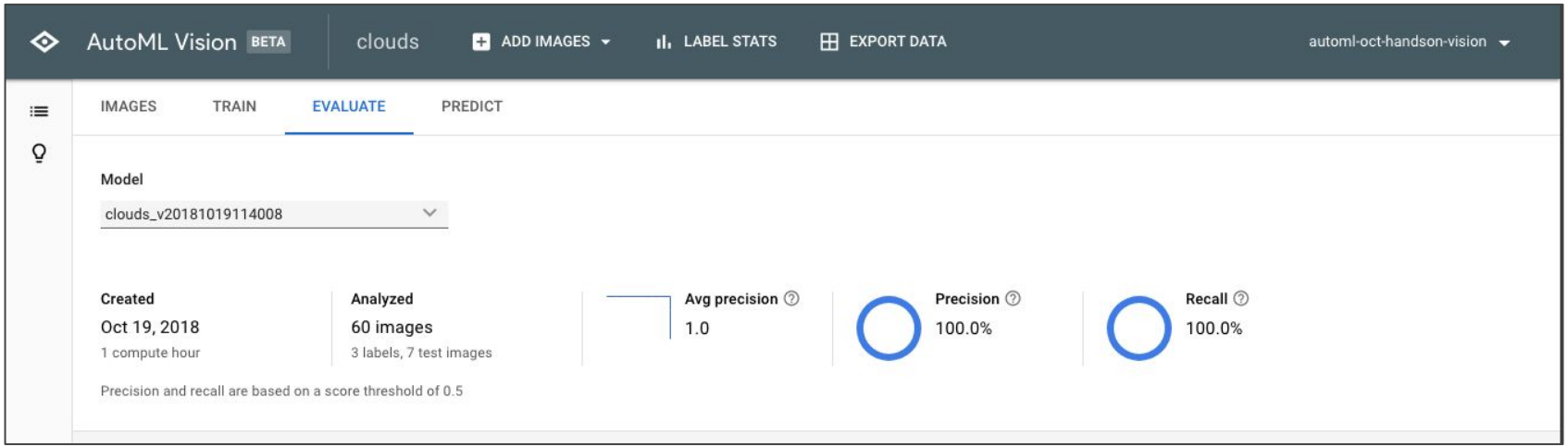


Getting more data can be hard. Some options:

explore public/free resources,
use crowdsourcing,
capture video and extract frames,
create variations of image (light, occlusion)

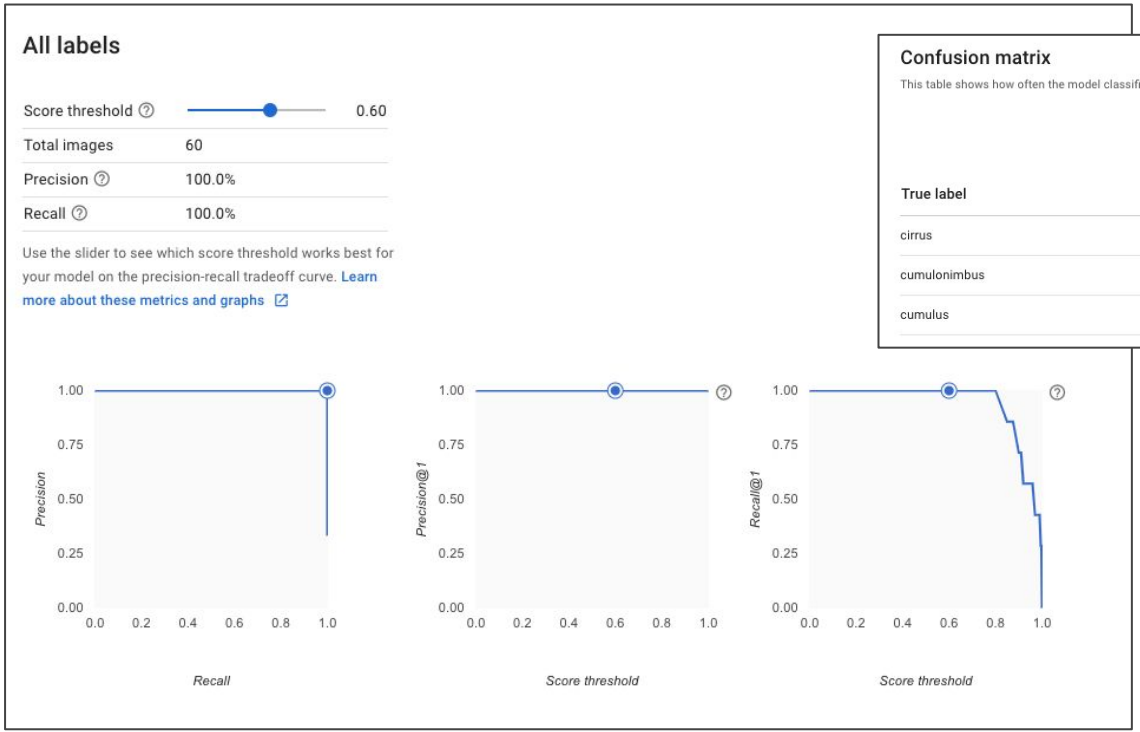


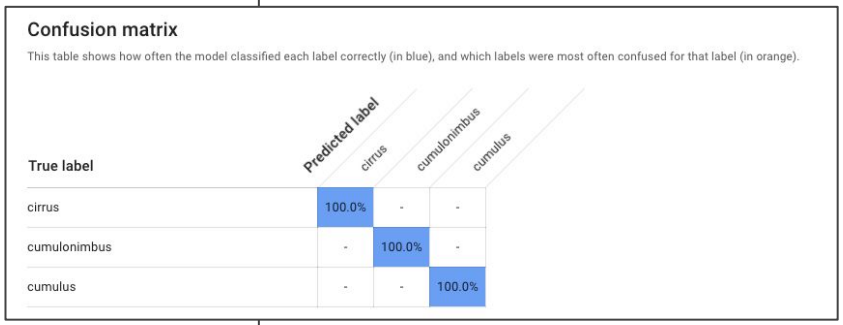
A.8 Codelab: Evaluate the model



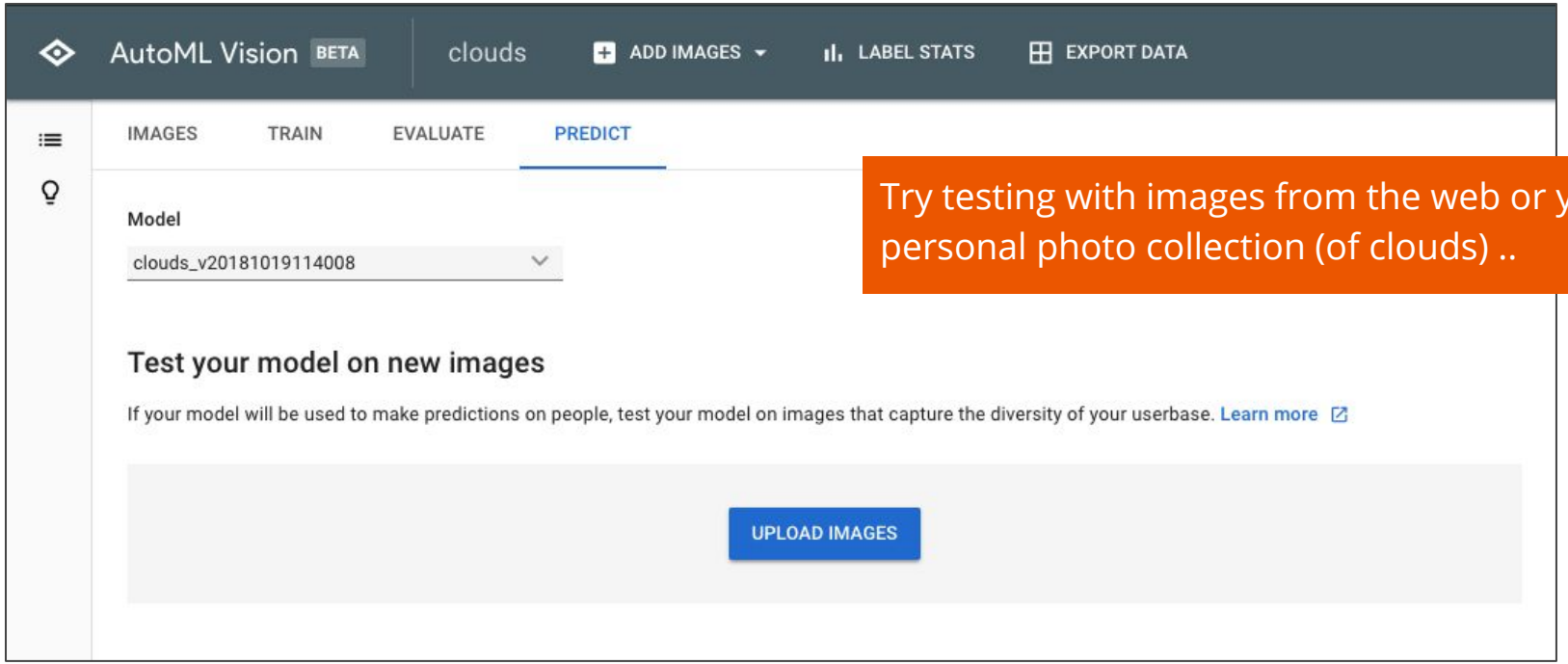
This level of precision/recall is not typical.
(reflects demo usage & small dataset)

A.8 Codelab: Evaluate the model



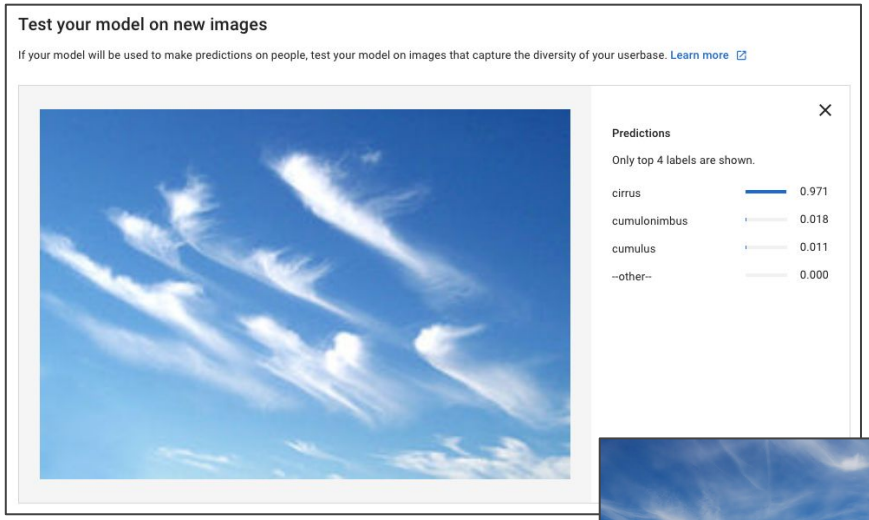


A.9 Codelab: Generate Predictions with model



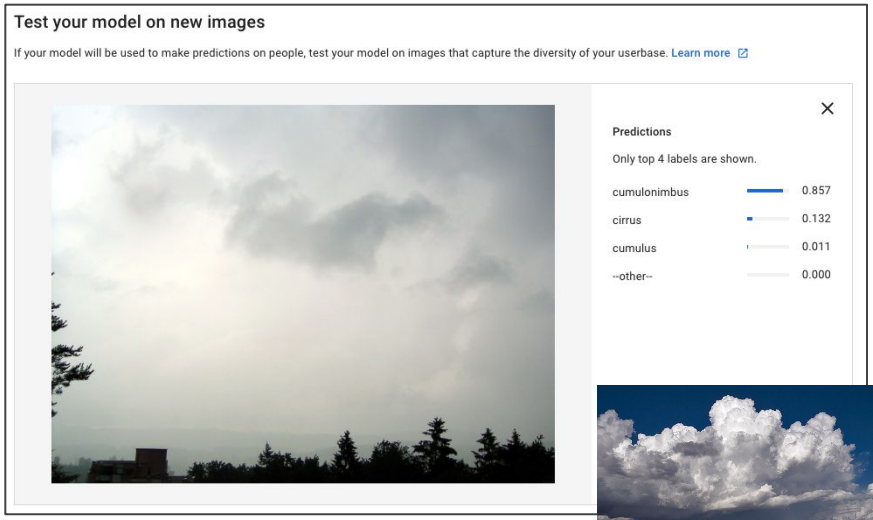
Try testing with images from the web or your personal photo collection (of clouds) ..

A.9 Codelab: Test examples



Wikipedia / cirrus uncinus

training



Wikipedia / nimbostratus

training

B. AutoML Vision Quickstart / 3600 images

- Uses sample from TensoFlow Blog Post flower dataset
- Step 1: Install and configure Google Cloud SDK
- Step 2: Setup project and create dataset
- Step 3: Train model using dataset
- Step 4: Evaluate trained model
- Step 5: Use trained model to make predictions
- (Delete model if not in use)

Short Break

Be right back

5 mins

AutoML Natural Language

Detect Entities & Sentiment in Converation
NL API
AutoML Natural Language
Data Preparation / Model Testing / Model Evaluation

35 mins

Cloud ML & Language: Natural Language

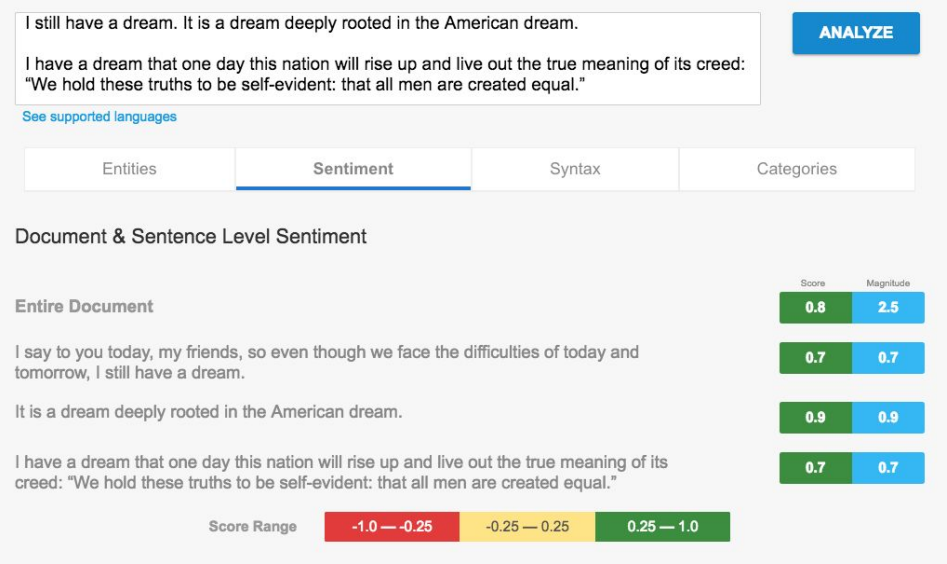
Cloud Natural Language API

Text Parsing & Analysis using pre-trained models for easy integration

AutoML Natural Language

Custom text classification models

What is the Natural Language API?



- Syntax analysis
- Entity recognition
- Sentiment analysis
- Content classification
- Multilanguage analysis
- + REST API & libraries

Cloud Natural Language API Pricing

	PRICE PER 1,000 UNITS, BY MONTHLY USAGE			
FEATURE	0 - 5K UNITS/MONTH	5K+ - 1M UNITS/MONTH	1M+ - 5M UNITS/MONTH	5M+ - 20M UNITS/MONTH
Entity Analysis	FREE	$1.00	$0.50	$0.25
Sentiment Analysis	FREE	$1.00	$0.50	$0.25
Syntax Analysis	FREE	$0.50	$0.25	$0.125
Entity Sentiment Analysis	FREE	$2.00	$1.00	$0.50

	PRICE PER 1,000 UNITS, BY MONTHLY USAGE			
FEATURE	0 - 30K UNITS/MONTH	30K+ - 250K UNITS/MONTH	250K+ - 5M UNITS/MONTH	5M+ UNITS/MONTH
Content Classification	FREE	$2.00	$0.50	$0.10

Cloud Natural Language API Interactive Demo

https://cloud.google.com/natural-language/

How AutoML Natural Language Works

AutoML Natural Language Pricing

TRAINING
$3 per hour

PREDICTION	
1–30,000 text records*	Free
30,001–5,000,000 text records**	$5 per 1,000 text records

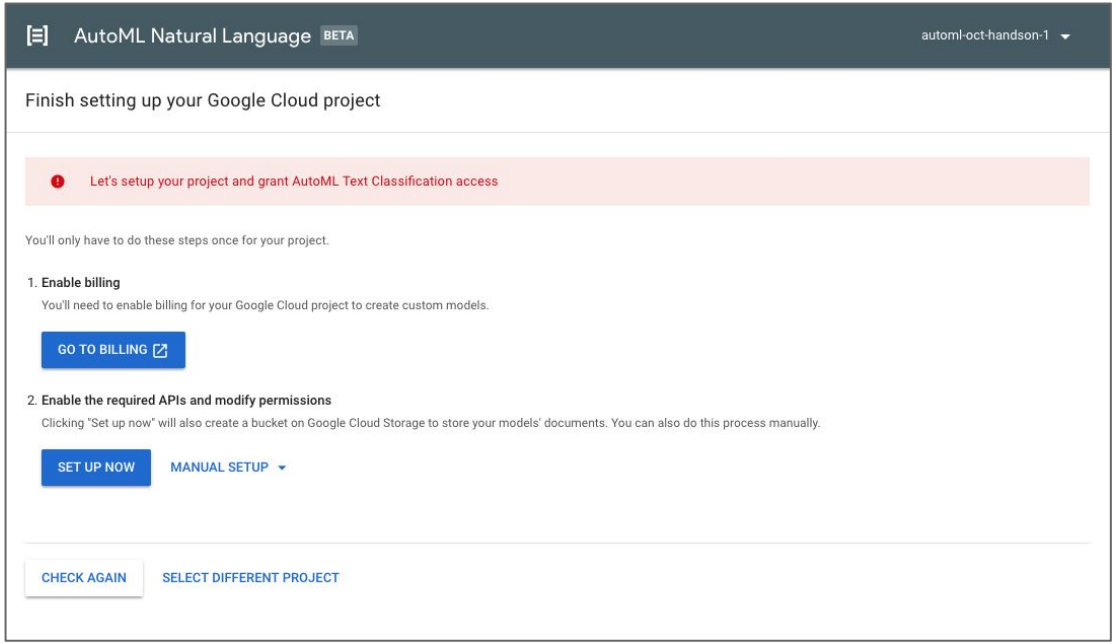
*A text record corresponds to a document provided as input to a Natural Language API request, see the pricing guide for more detail.

AutoML Natural Language Training Example

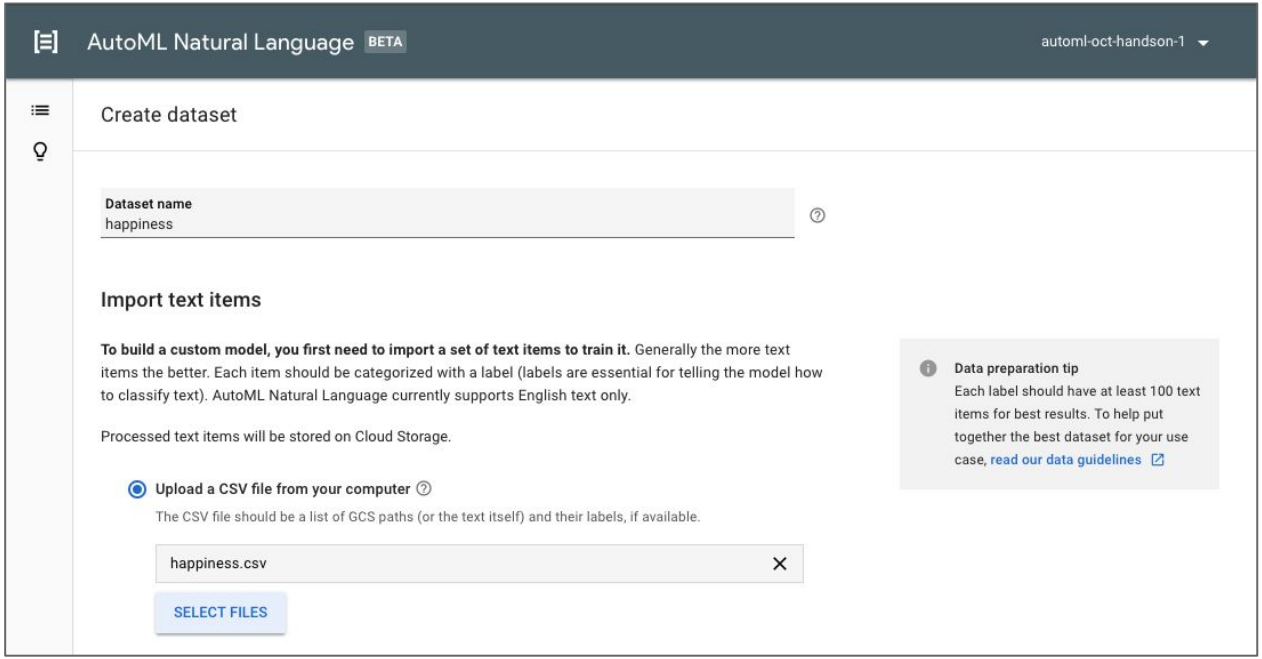
AutoML Natural Language / Quickstart

- Uses sample from Kaggle HappyDB dataset
- Step 1: Install and configure Google Cloud SDK
- Step 2: Setup project and create dataset
- Step 3: Train model using dataset
- Step 4: Evaluate trained model
- Step 5: Use trained model to make predictions
- (Delete model if not in use)

Step 2a / Setup AutoML NL Project

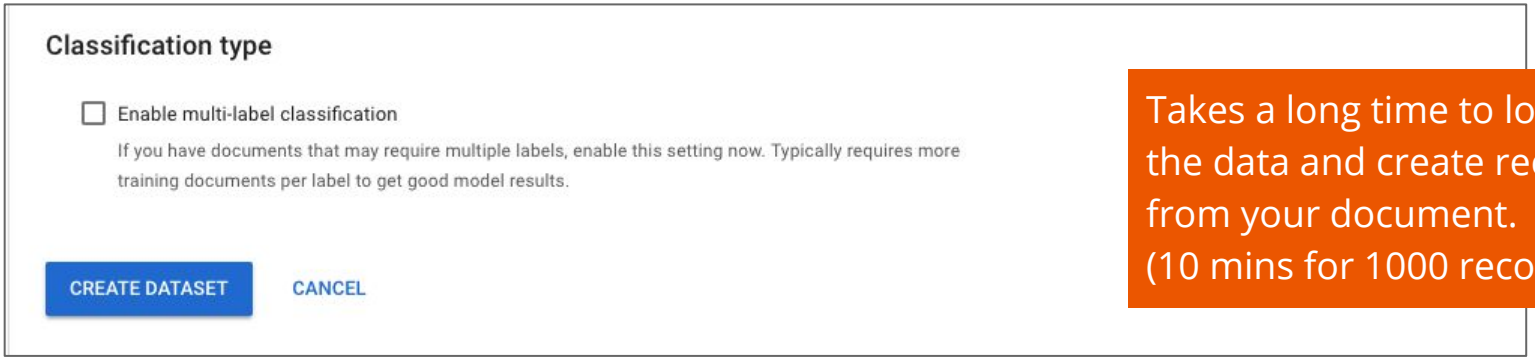


Step 2b / Add New Dataset



Step 2c / Pick classification type, create!

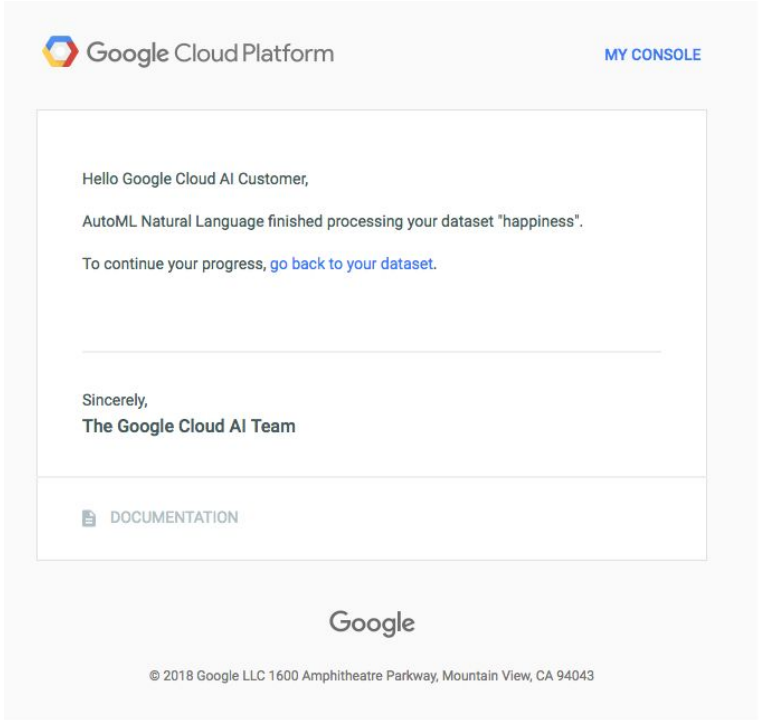
- MULTICLASS assigns ONE label per document
- MULTI-LABEL allows MANY labels per document



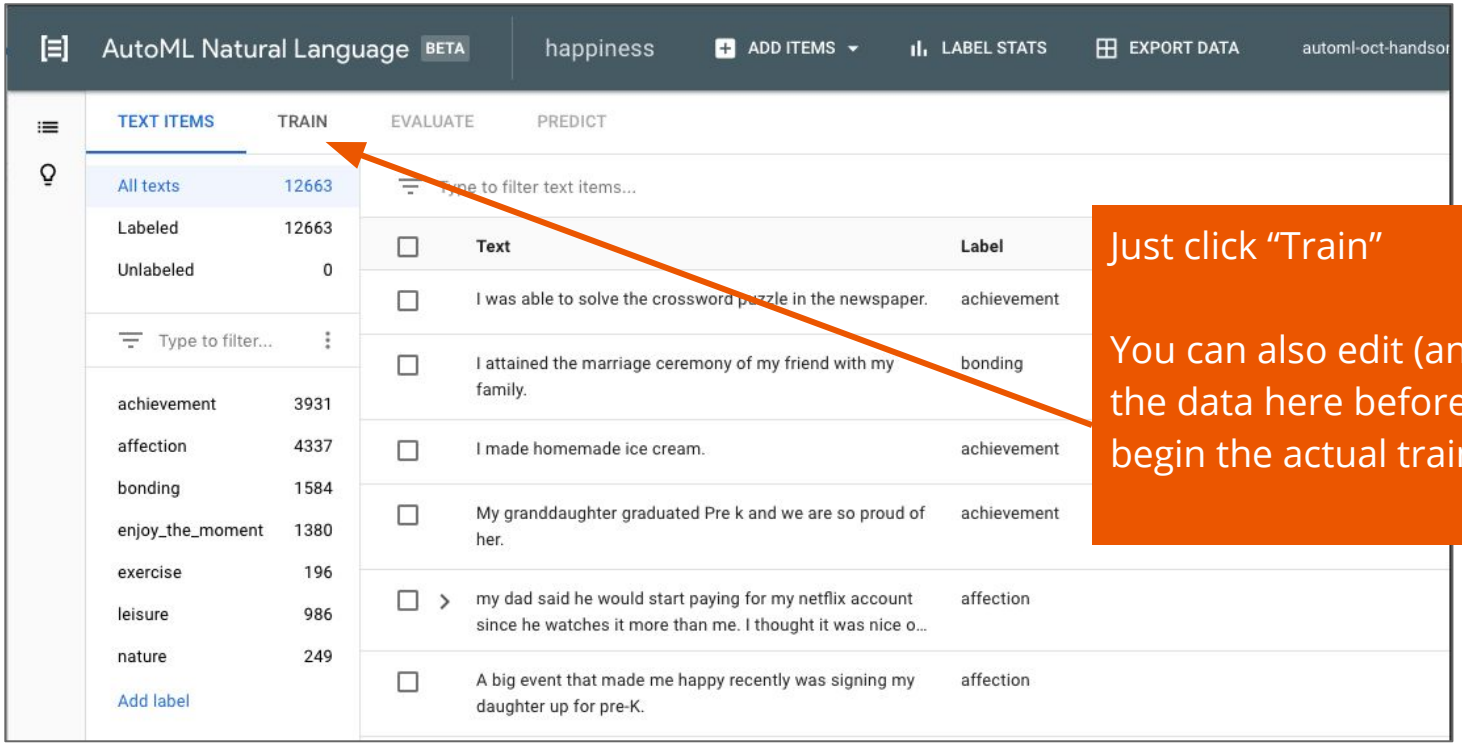
Takes a long time to load the data and create records from your document.
(10 mins for 1000 records)

Step 2d / Dataset created

- Example:
  `I finally mastered how to make crispy tofu.,`**`achievement`**
- Example:
  `Daughter committed to a college.,`**`affection`**



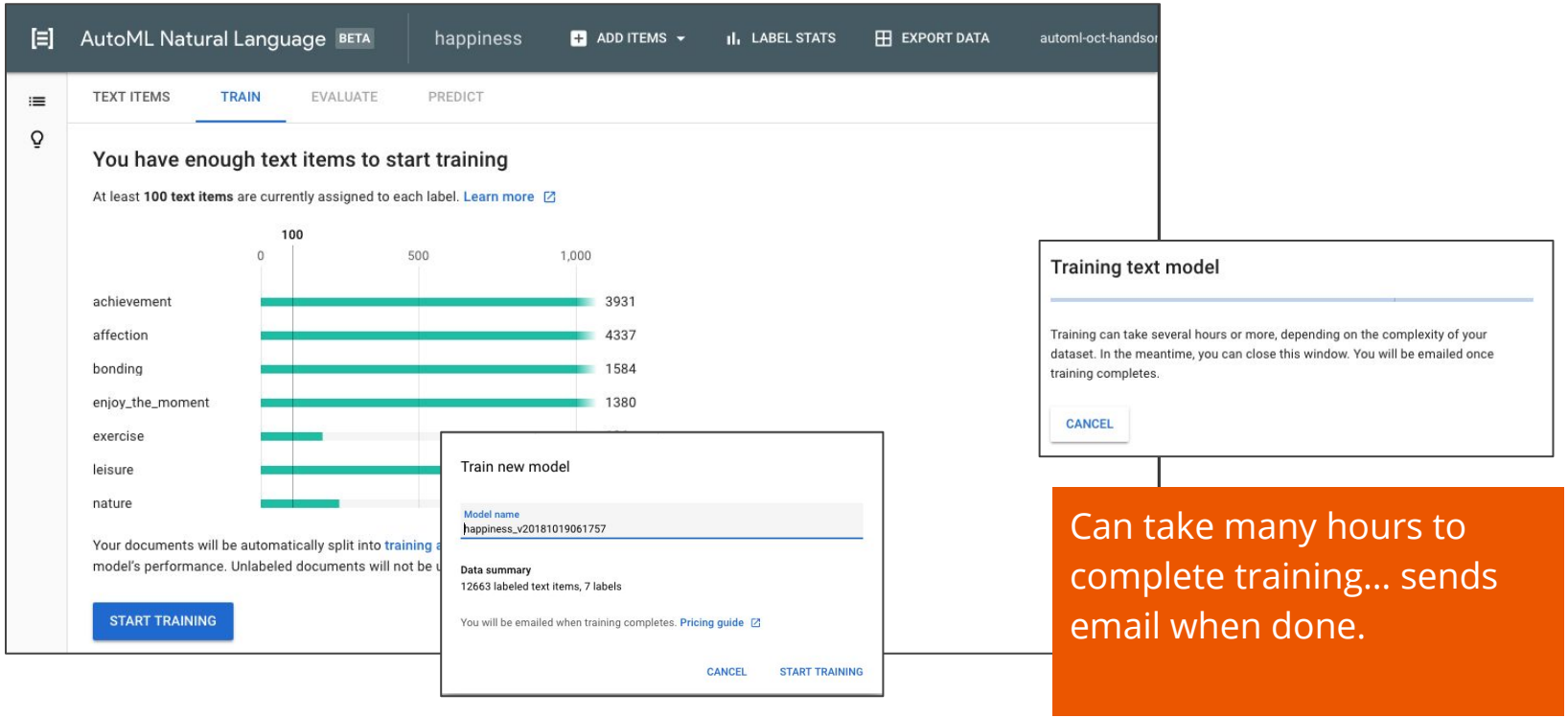
Step 3: Train Model Using Dataset



Just click “Train”

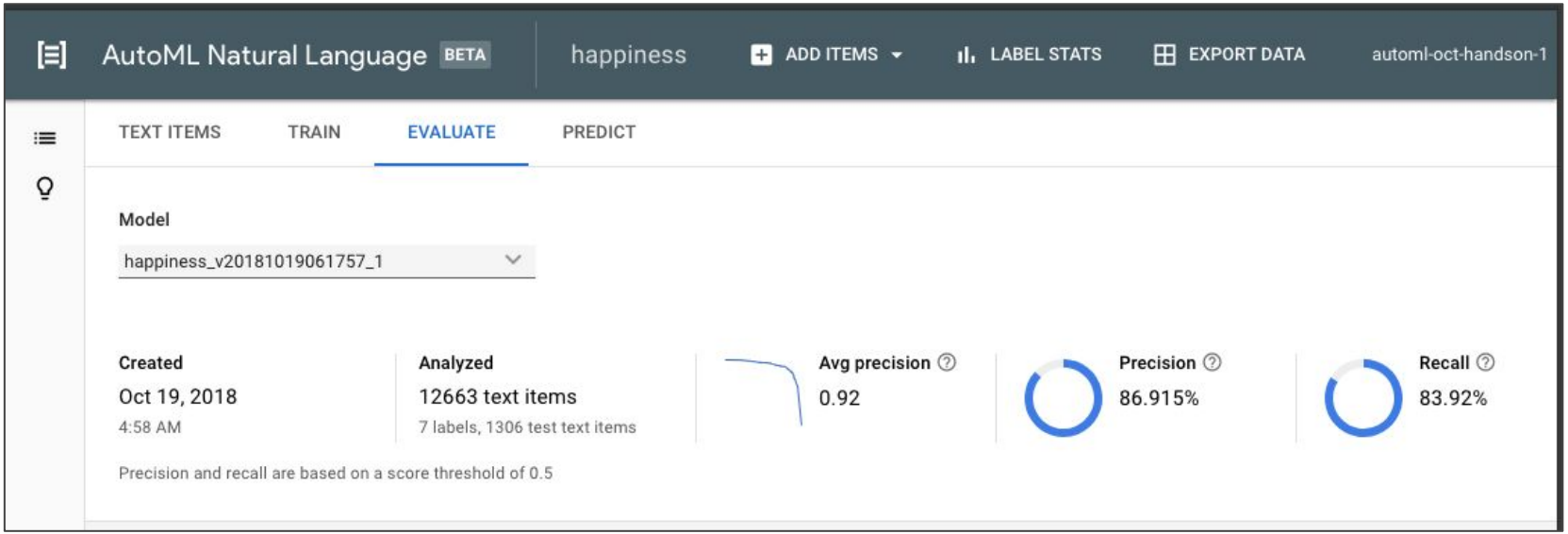
You can also edit (and clean) the data here before you begin the actual training..

Step 3: Train Model Using Dataset



Can take many hours to complete training… sends email when done.

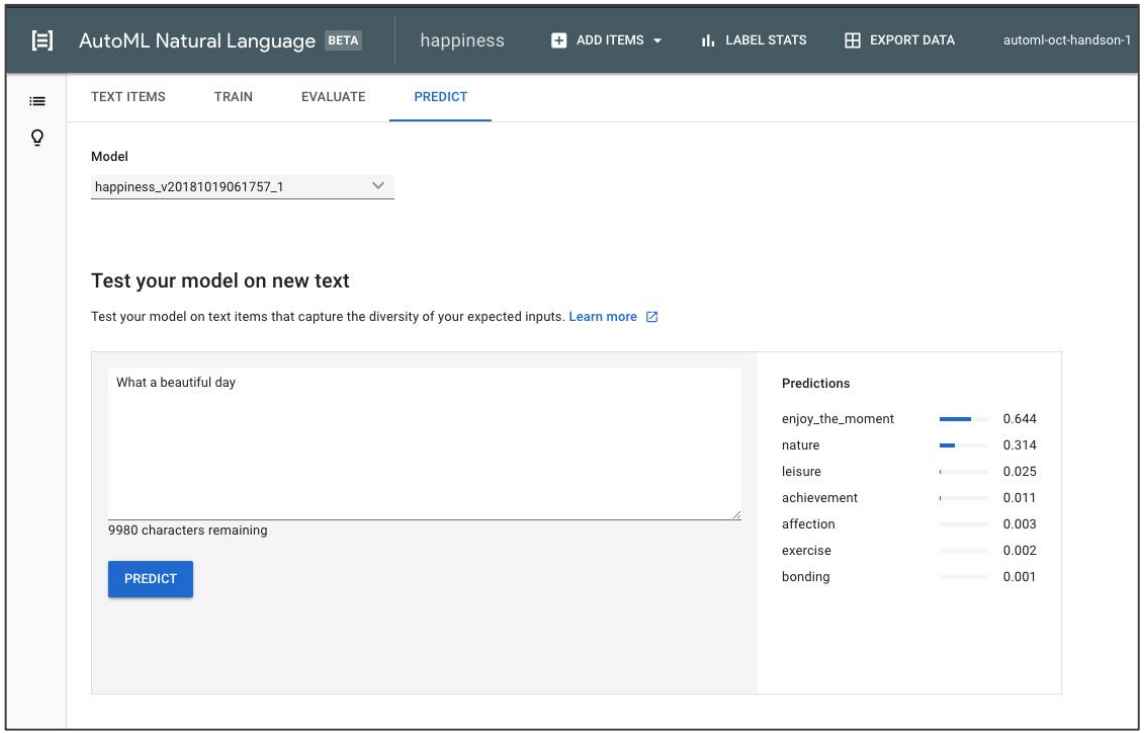
Step 4: Evaluate Trained Model



Precision. Of those labeled, how many were labeled correctly?

Recall. Of those that should have the label, how many were assigned it?

Step 5: Use Trained Model / In Dashboard



Step 5: Use Trained Model / Via API

Use your custom model

You can now run predictions on text items using your custom content classification model. (Note: You will need a service account)

REST API | PYTHON

Execute the request

```
export GOOGLE_APPLICATION_CREDENTIALS=key-file-path

curl -X POST \
  -H "Authorization: Bearer $(gcloud auth application-default print-access-token)" \
  -H "Content-Type: application/json" \
  https://automl.googleapis.com/v1beta1/projects/automl-oct-handson-1/locations/us-central1/models/TCN897237531609594639
  -d '{
        "payload" : {
          "textSnippet": {
               "content": "YOUR TEXT HERE",
                "mime_type": "text/plain"
           },
        }
      }'
```

Step 5: Use Trained Model / Via Client Library

predict.py

```
import sys

from google.cloud import automl_v1beta1
from google.cloud.automl_v1beta1.proto import service_pb2


def get_prediction(content, project_id, model_id):
  prediction_client = automl_v1beta1.PredictionServiceClient()

  name = 'projects/{}/locations/us-central1/models/{}'.format(project_id, model_id)
  payload = {'text_snippet': {'content': content, 'mime_type': 'text/plain' }}
  params = {}
  request = prediction_client.predict(name, payload, params)
  return request  # waits till request is returned

if __name__ == '__main__':
  content = sys.argv[1]
  project_id = sys.argv[2]
  model_id = sys.argv[3]

  print get_prediction(content, project_id,  model_id)
```

Execute the request

```
python predict.py "YOUR TEXT HERE" automl-oct-handson-1 TCN8972375316095946390
```

Short Break

Be right back

5 mins

AutoML Translation

Translate Query Language
Translation API
AutoML Translate
Data Preparation / Model Testing / Model Evaluation

25 mins

Cloud ML & Language: Translation

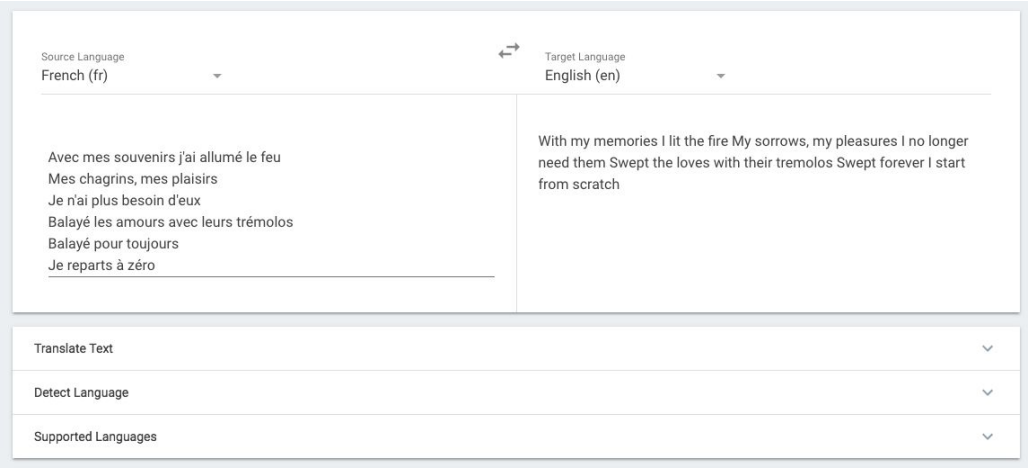
Cloud Translation API

Language detection & translation

AutoML Translation

Custom domain-specific translation

What is the Translation API?



- Text Translation
- Language Detection
- HTML or Text inputs
- 100+ languages (1000s of language-pairs)
- Continuous updates
- + REST API & libraries

Cloud Translate API Pricing

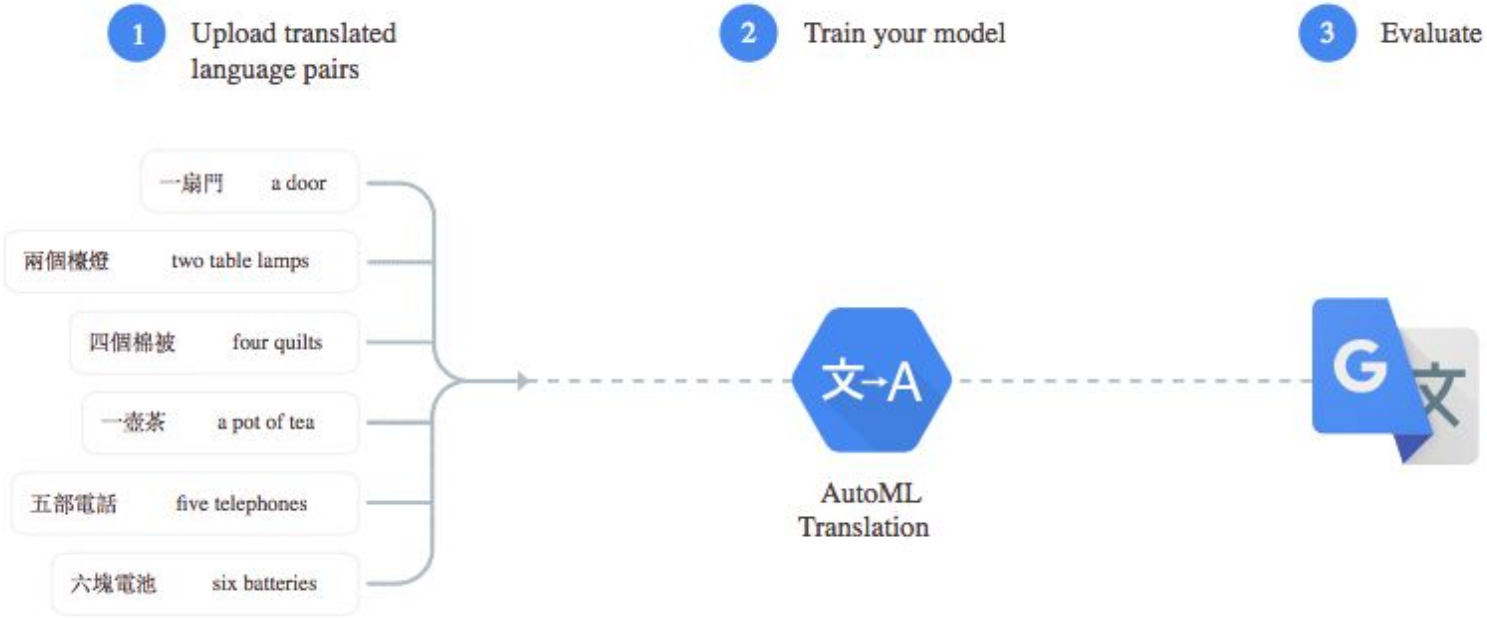
Translation API pricing is based on usage. Translation usage is calculated in millions of characters. You are billed per character, so you pay only for what you use.

FEATURE	COST (USD) UP TO 1,000 M CHARACTERS/MONTH
Text Translation	$20 per million characters
Language detection	$20 per million characters

Cloud Translate API Interactive Demo

https://cloud.google.com/translate/

How AutoML Translate Works



AutoML Translation Pricing

AutoML Translation pricing is based on Training and Prediction usage and storage.

TRAINING
$76 per hour

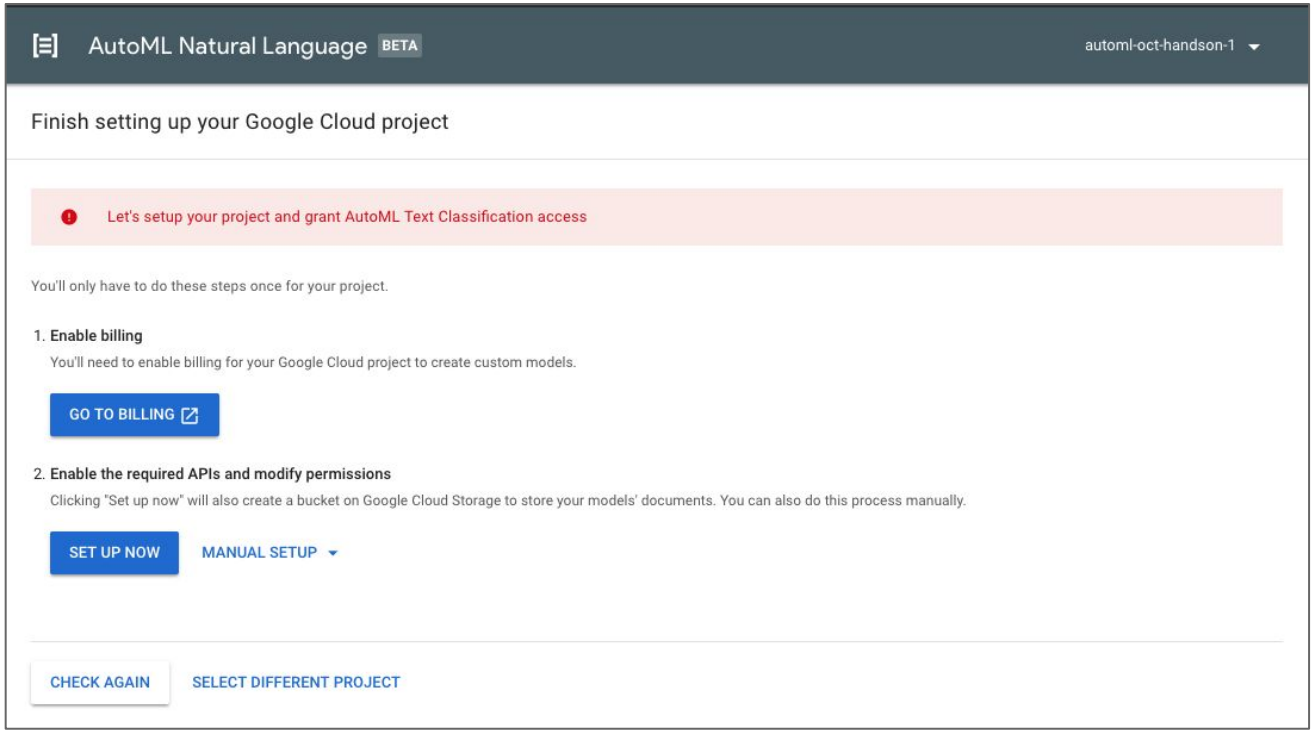
PREDICTION	
1–500,000 characters	Free
500,001–5,000,000 characters*	$80 per million characters

AutoML Translation Training Example

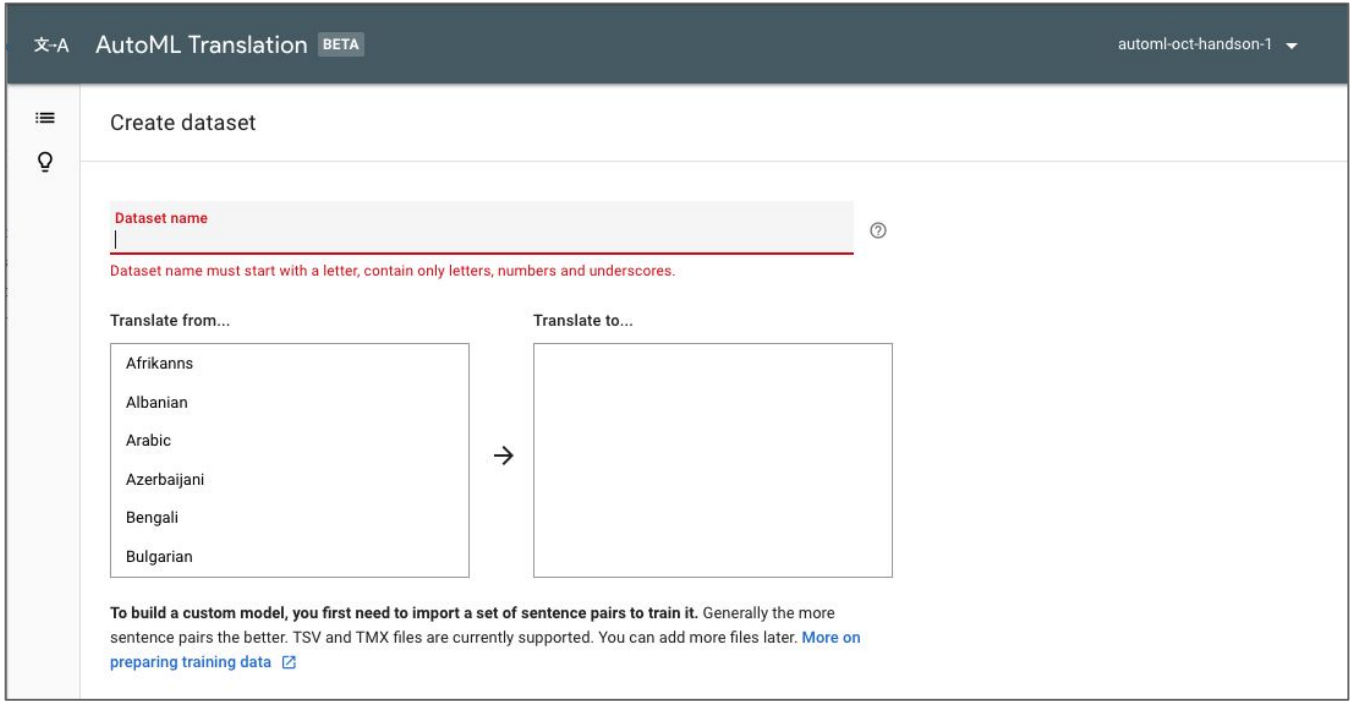
AutoML Translation / Quickstart

- Uses sample from Google Quickstart (en-es, en-de, en-ru, en-fr)
- Step 1: Install and configure Google Cloud SDK
- Step 2: Setup project and create dataset
- Step 3: Train model using dataset
- Step 4: Evaluate trained model
- Step 5: Use trained model to make predictions
- (Delete model if not in use)

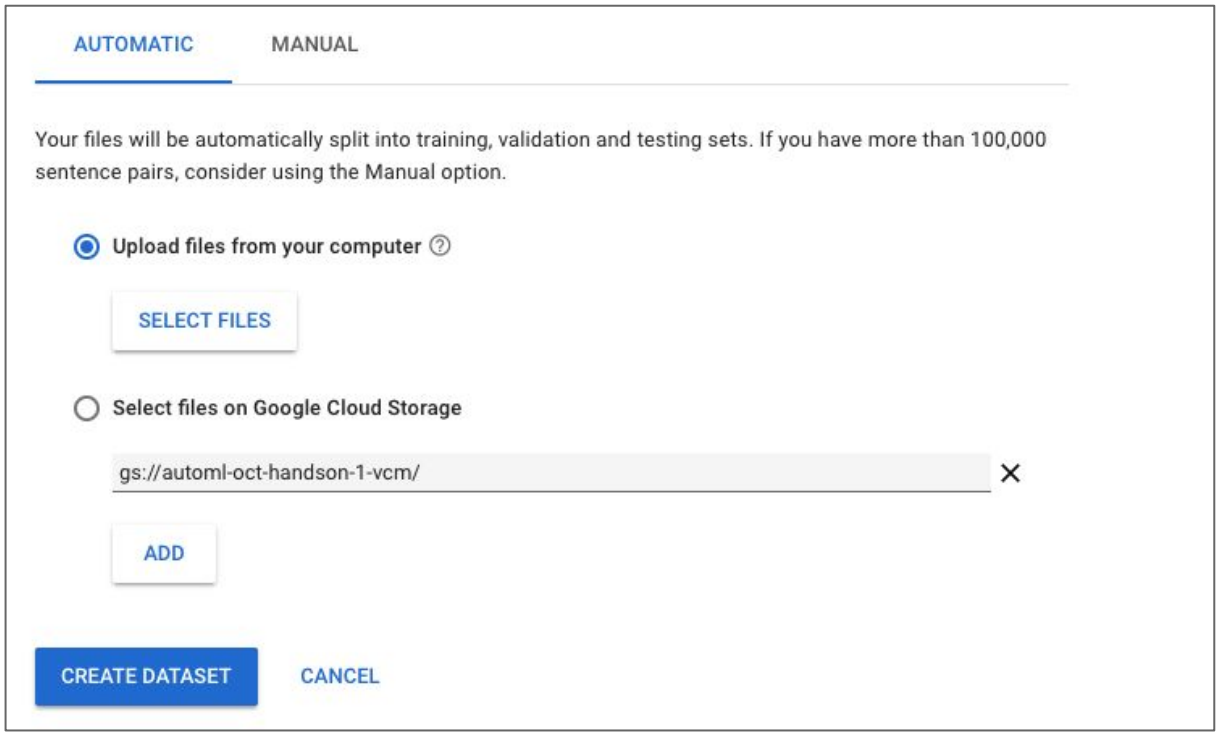
Step 2a / Setup AutoML Translate Project



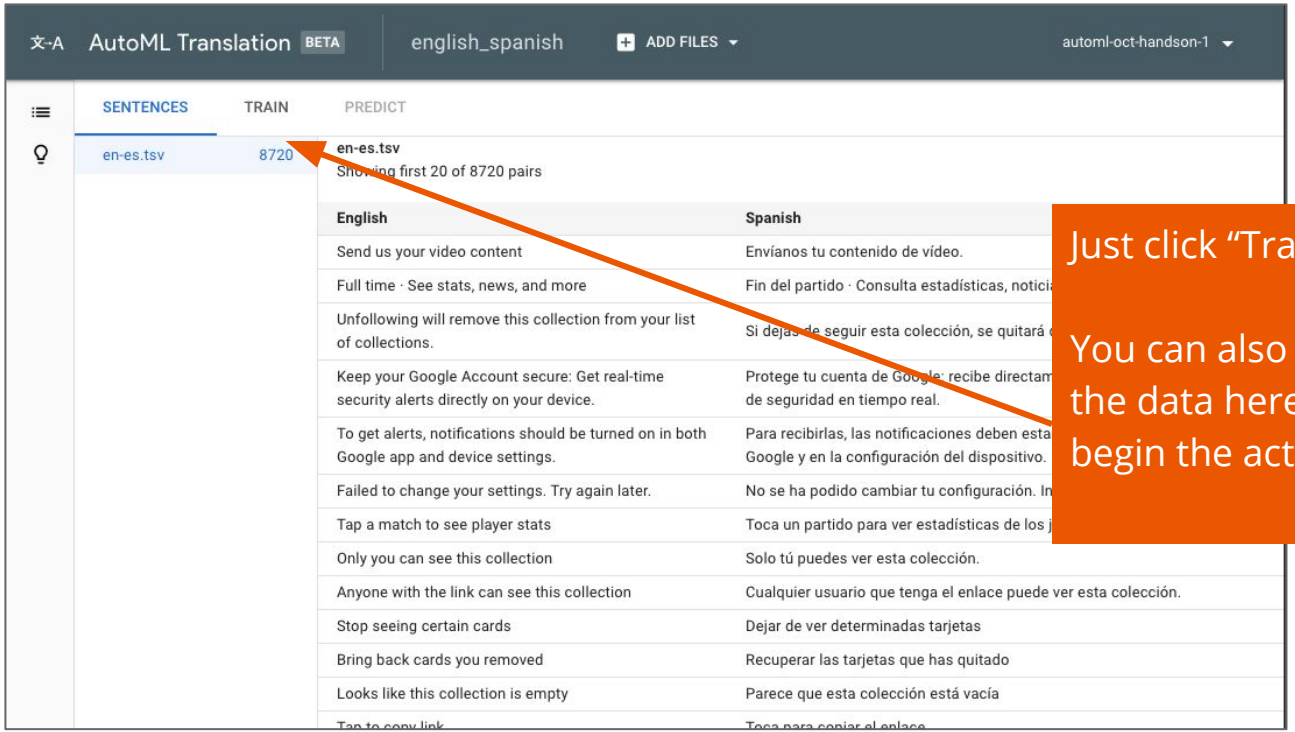
Step 2b / Add New Dataset (sentence pairs)



Step 2c / Upload files, create!



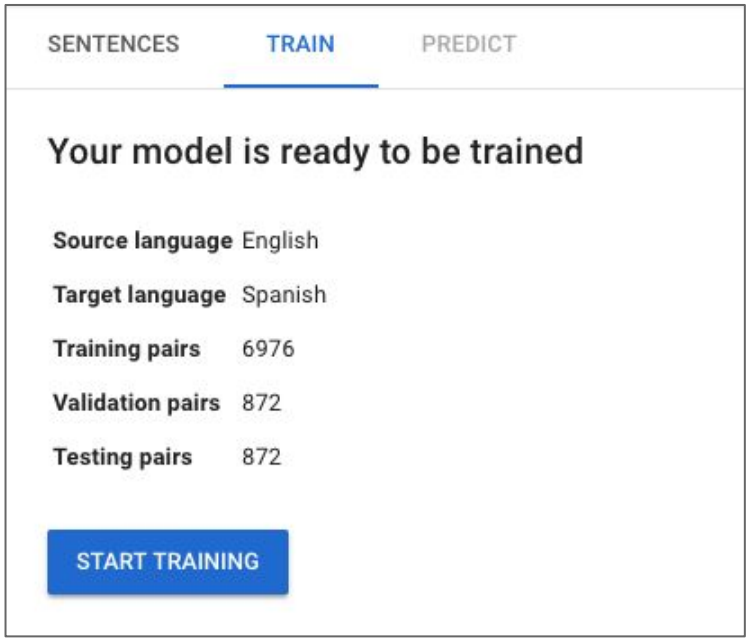
Step 2d / Dataset created



Just click “Train”

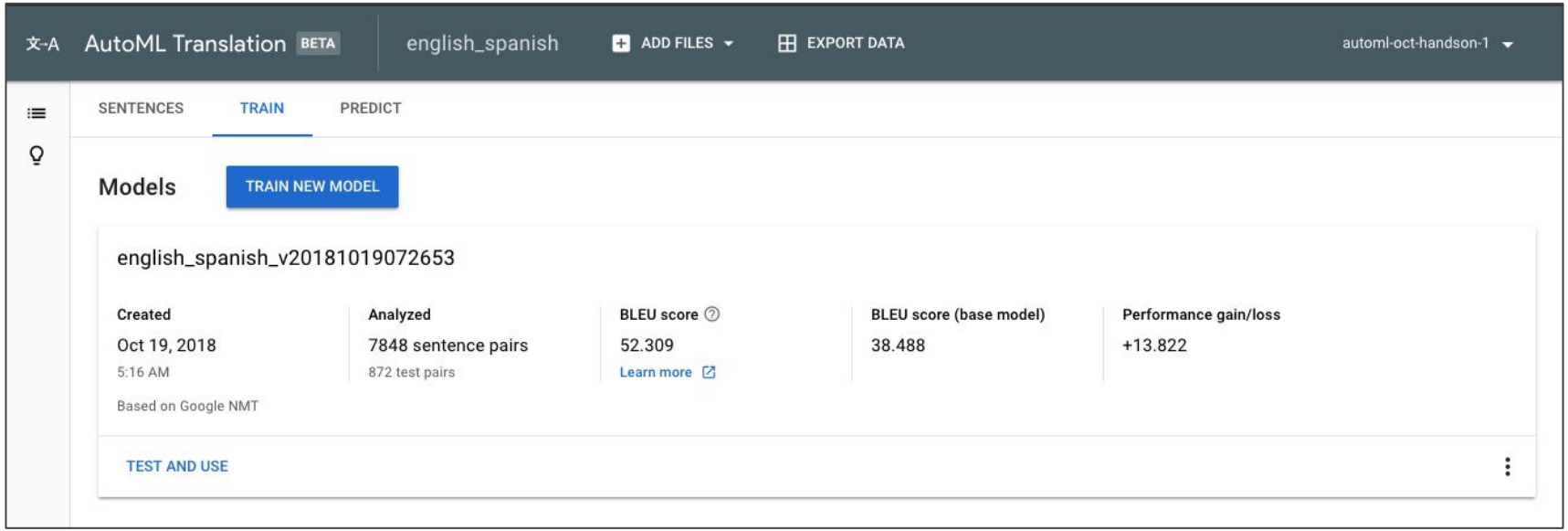
You can also edit (and clean) the data here before you begin the actual training..

Step 3: Train Model Using Dataset

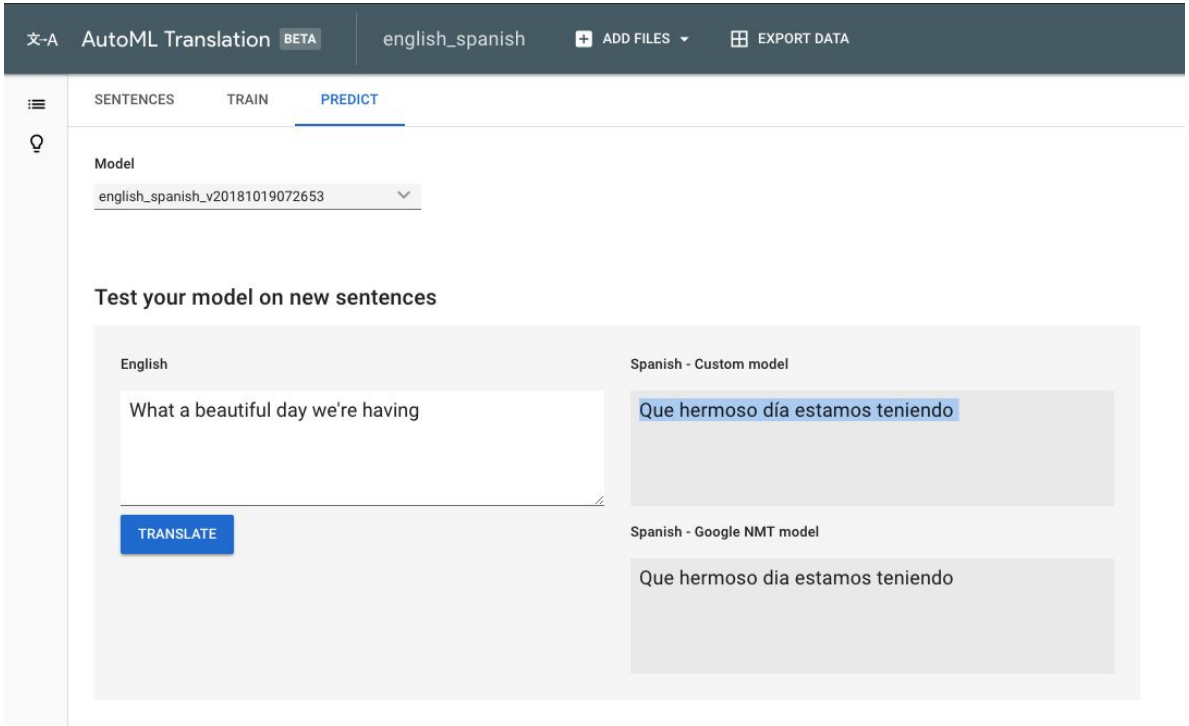


Note how dataset is automatically split into training, validation, testing sets (80-10-10)

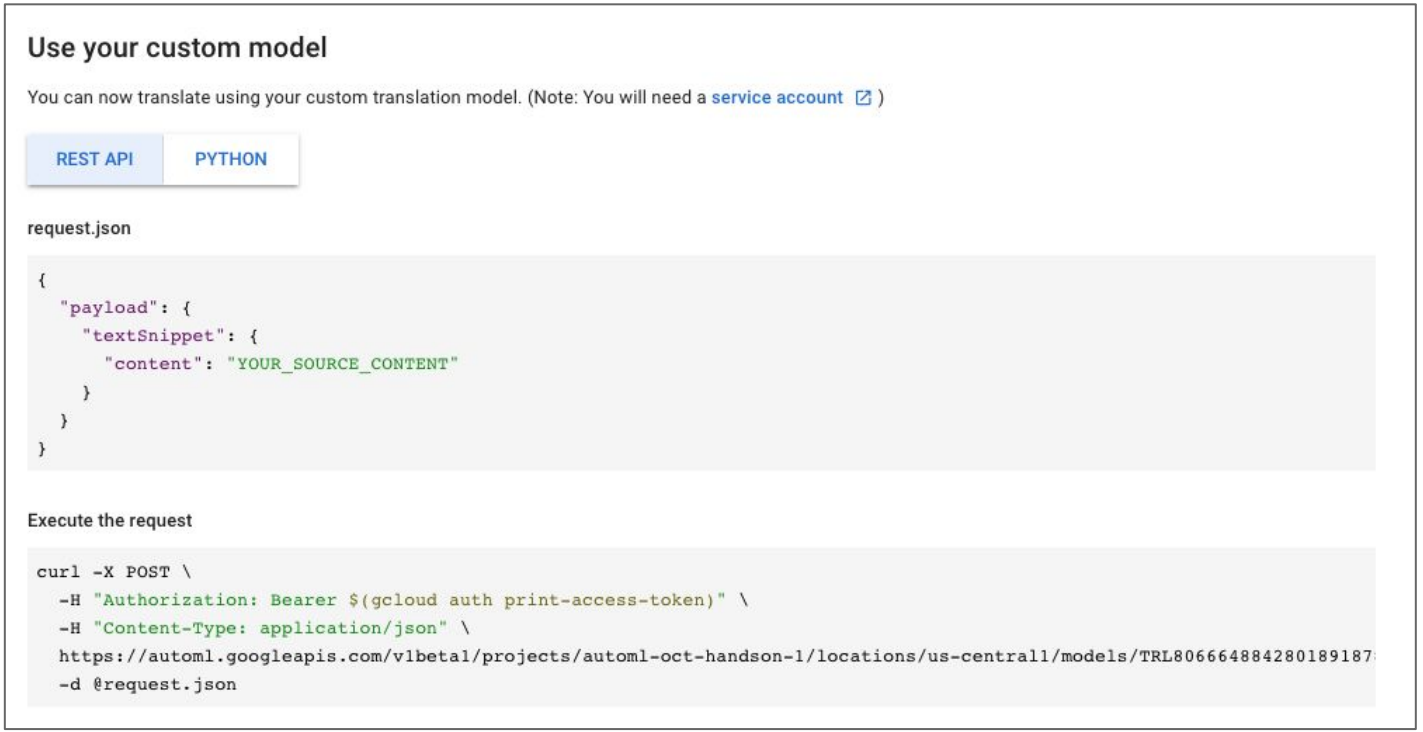
Step 4: Evaluate Trained Model



Step 4: Evaluate Trained Model



Step 5: Use Trained Model to Make Prediction



Use your custom model

You can now translate using your custom translation model. (Note: You will need a service account)

REST API | PYTHON

request.json

```
{
  "payload": {
    "textSnippet": {
      "content": "YOUR_SOURCE_CONTENT"
    }
  }
}
```

Execute the request

```
curl -X POST \
  -H "Authorization: Bearer $(gcloud auth print-access-token)" \
  -H "Content-Type: application/json" \
  https://automl.googleapis.com/v1beta1/projects/automl-oct-handson-1/locations/us-central1/models/TRL806664884280189187
  -d @request.json
```

Short Break

Be right back

5 mins

Putting It All Together

Google ML Ecosystem // Cloud ML, AutoML, TensorFlow, Firebase MLKit

What We Covered // AutoML Vision, AutoML Natural Language, AutoML Translate

Where Is It Useful // Application Examples? Share them with me

What's Next // Resources & Learning Paths

25 mins

AutoML Resources for Self-Study

- AutoML Vision | Docs & Quickstart | Tutorial
- AutoML Natural Language | Docs & Quickstart | Tutorial
- AutoML Translate | Docs & Quickstart | Tutorial
- Fast.ai | AutoML & Neural Architecture Search (pt1, pt2, pt3)
- Content | AI Adventures (@yufengg) & Medium (@sRobTweets)
- Codelabs | AutoML Vision

FOR MORE INFORMATION

THANK YOU

- **Questions**
  contact@nityan.me
- **Slides**
  http://bit.ly/automl18-slides
- **Website**
  https://automl18.bitnbot.com

I hope to update the website with new datasets & resources in upcoming weeks.