

# **CMPE 258 ASSIGNMENT #2**

## **PART 2a:**

**Execute AutoML Vision and  
Timeseries Forecasting models**

# 1. Downloading code and training dataset

Train and deploy on-device image classification model with AutoML Vision in ML Kit

⌚ 55 mins remaining

English ▾



- 1 Introduction
- 2 Setup
- 3 Prepare training dataset
- 4 Train a model
- 5 Use the model in mobile apps
- 6 (Optional) Use remote model
- 7 Congratulations!

## 2. Setup

### Download the code and training dataset

Download a zip archive that contains the source code for this codelab and a training dataset. Extract the archive in your local machine.

📄 Download ZIP

- There is an Android app and an iOS app provided in this codelab that have similar functionalities. You can choose either platform that you are more familiar with to proceed.
- The flower\_photos.zip file in the archive is the training dataset that you will use in step 3: Prepare training dataset

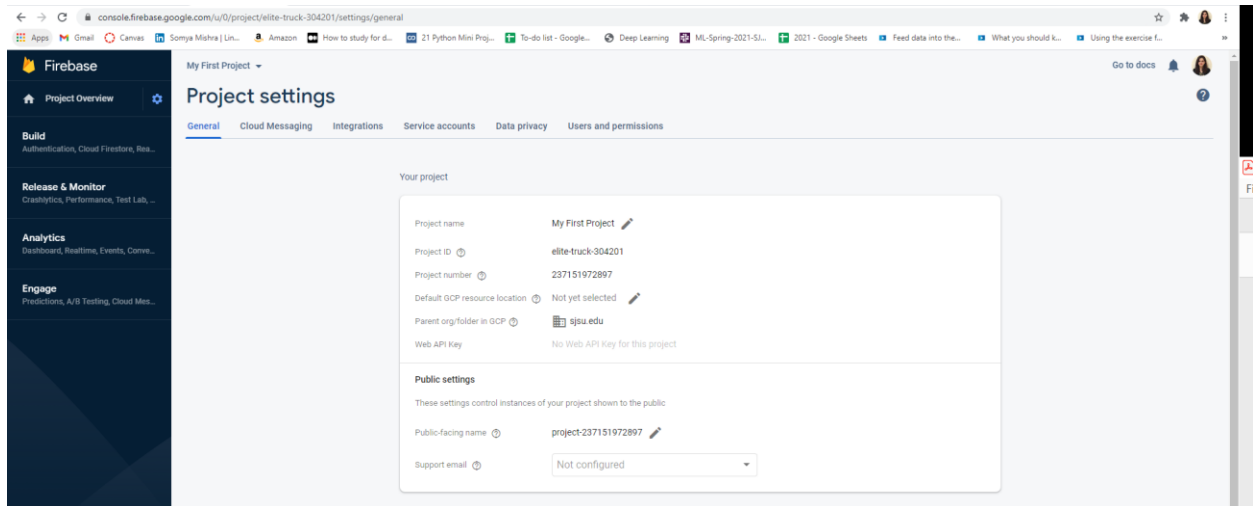
### Create Firebase console Project

1. Go to the [Firebase console](#).
2. Select **Create New Project**, and name your project "ML Kit Codelab."

### Setup the Android app

1. Add our Android app to the Firebase project. Android package name: `com.google.firebase.codelab.mlkit.automl`
2. Download the `google-services.json` config file, and put it in the Android app at `android/mlkit-automl/app/google-services.json`.

## 2. Creating Firebase console Project



## 3. Adding Firebase to iOS

× Add Firebase to your iOS app

1

Register app

iOS bundle ID ⓘ

com.google.firebase.codelab.mlkit.automl

App nickname (optional) ⓘ

My iOS App

App Store ID (optional) ⓘ

123456789

Register app

ios apps

ios

My iOS App

com.google.firebase.codelab.mlkit.automl

SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

See SDK instructions

GoogleService-Info.plist

App ID

1:237151972897:ios:fb0d248255d3ecca88fd60

Encoded App ID

app-1-237151972897-ios-fb0d248255d3ecca88fd60

App nickname

My iOS App

Bundle ID

com.google.firebase.codelab.mlkit.automl

App Store ID

Add an App Store ID

Team ID

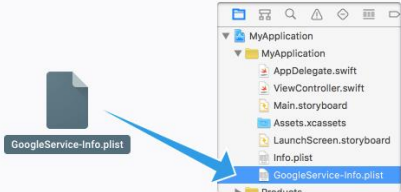
Add a Team ID

Remove this app

#### 4. Setup the iOS App


### × Add Firebase to your iOS app

✓ Register app  
iOS bundle ID: com.google.firebase.codelab.mlkit.automl, App nickname: MLKit Demo

2 Download config file  
[Instructions for Xcode below](#) | [Unity](#) [C++](#)  
[Download GoogleService-Info.plist](#)  
Move the GoogleService-Info.plist file you just downloaded into the root of your Xcode project and add it to all targets.  
  
[Next](#)

3 Add Firebase SDK

4 Add initialization code



### × Add Firebase to your iOS app

✓ Register app  
iOS bundle ID: com.google.firebase.codelab.mlkit.automl, App nickname: MLKit Demo

✓ Download config file

3 Add Firebase SDK  
[Instructions for CocoaPods](#) | [Download ZIP](#) [Unity](#) [C++](#)  
Google services use [CocoaPods](#) to install and manage dependencies. Open a terminal window and navigate to the location of the Xcode project for your app.  
Create a Podfile if you don't have one:  

```
$ pod init
```


  
Open your Podfile and add:  

```
# add the Firebase pod for Google Analytics
pod 'Firebase/Analytics'
# add pods for any other desired Firebase products
# https://firebase.google.com/docs/ios/setup#available-pods
```

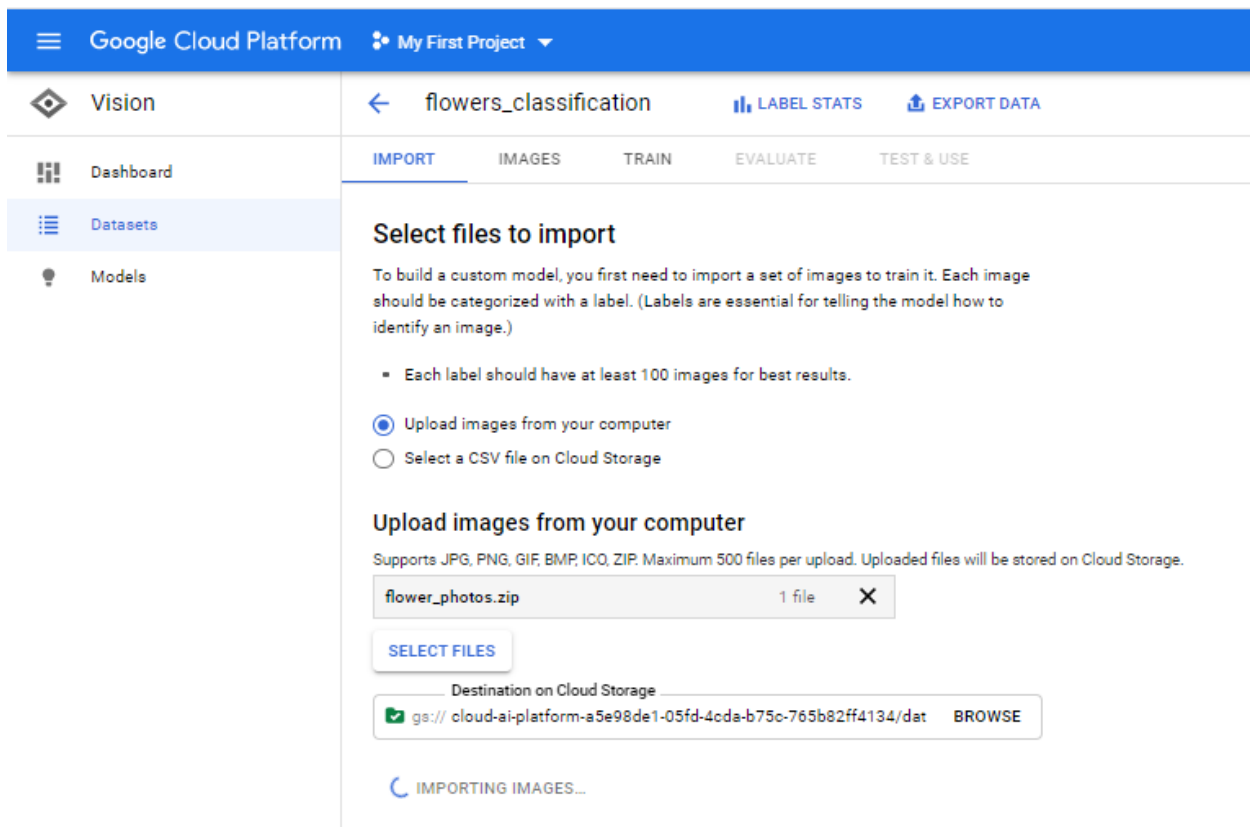
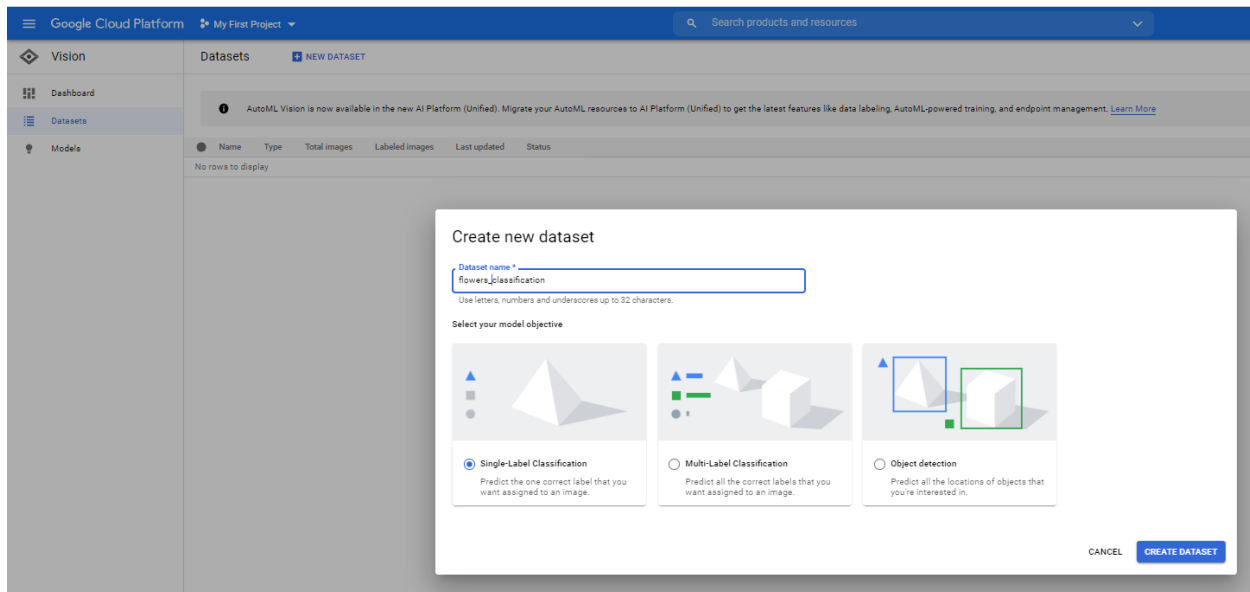
  
Save the file and run:  

```
$ pod install
```

  
This creates an .xcworkspace file for your app. Use this file for all future development on your application.



## 5. Preparing Training Dataset



Google Cloud Platform

My First Project

Search products and resources

Vision

untitled\_1615418069987

LABEL STATS

EXPORT DATA

IMPORT

IMAGES

TRAIN

EVALUATE

TEST & USE

Single-Label Classification

all images 1,000

Filter images

Labelled 1,000

Unlabelled 0

Filter labels

ADD NEW LABEL

daisy 200

dandelion 200

roses 200

sunflowers 200

tulips 200

Google Cloud Platform

My First Project

Search products and resources

Vision

untitled\_1615418069987

LABEL STATS

EXPORT DATA

IMPORT

IMAGES

TRAIN

EVALUATE

TEST & USE

You have enough images to start 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 also have at least 8, 1, 1 images each assigned to your Train, Validation and Test sets.

Labels	Images	Train	Validation	Test
daisy	<div><div></div></div> 200	160	20	20
dandelion	<div><div></div></div> 200	160	20	20
roses	<div><div></div></div> 200	160	20	20
sunflowers	<div><div></div></div> 200	160	20	20
tulips	<div><div></div></div> 200	160	20	20

START TRAINING

Google Cloud Platform

My First Project

Search products and resources

Vision

Dashboard

Datasets

Models

untitled\_1615418069987

LABEL STATS

EXPORT DATA

IMPORT

IMAGES

TRAIN

EVALUATE

TEST & USE

You have enough images to start 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 also have at least 8, 1, 1 images each assigned to your [Train](#), [Validation](#) and [Test sets](#).

Labels	Images	Train	Validation	Test
berry	<div></div>	200	160	20
cardamom	<div></div>	200	160	20
rose	<div></div>	200	160	20
coriander	<div></div>	200	160	20
tulip	<div></div>	200	160	20

START TRAINING

Train new model

Define your model

Set a node hour budget

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

Enter the maximum number of node hours you want to spend training your model.

We recommend using [10 node hours](#) for your dataset. However, you can train for as little as 8 node hours. You may also be eligible to train with free node hours. [Pricing guide](#)

Set your budget \*

node hours

Estimated completion date: Mar 10, 2021 5 PM GMT-6

Deploy model to 1 node after training

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

START TRAINING

CANCEL

untitled\_1615418069987

LABEL STATS

EXPORT DATA

IMPORT

IMAGES

TRAIN

EVALUATE

TEST & USE

Models

TRAIN NEW MODEL

untitled\_16154180\_20210310042238

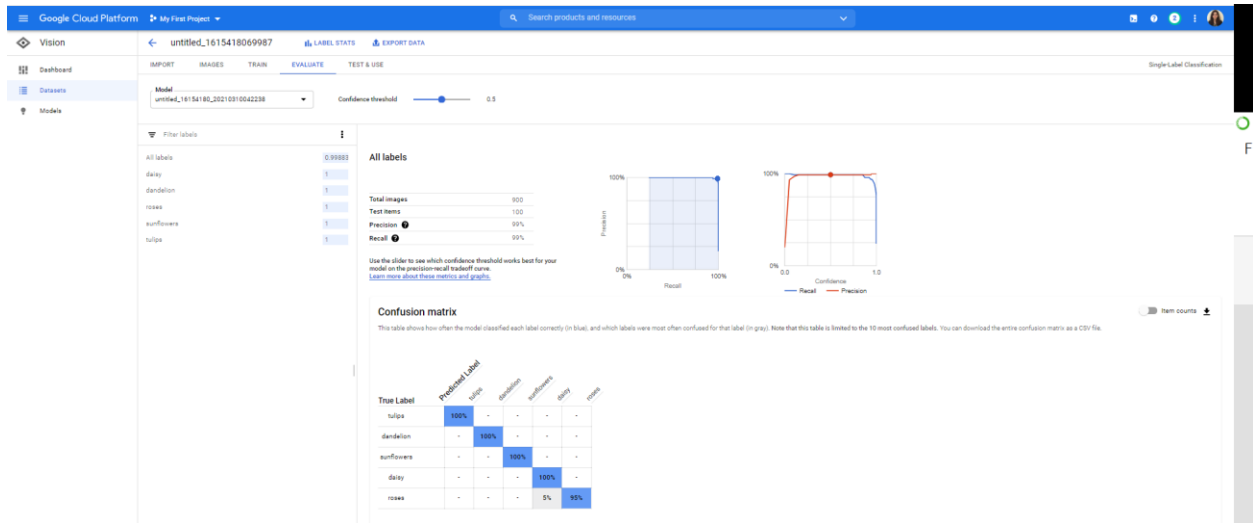
Training may take several hours. This includes node training time as well as infrastructure set up and tear down, which you aren't charged for.

You will be emailed once training completes.

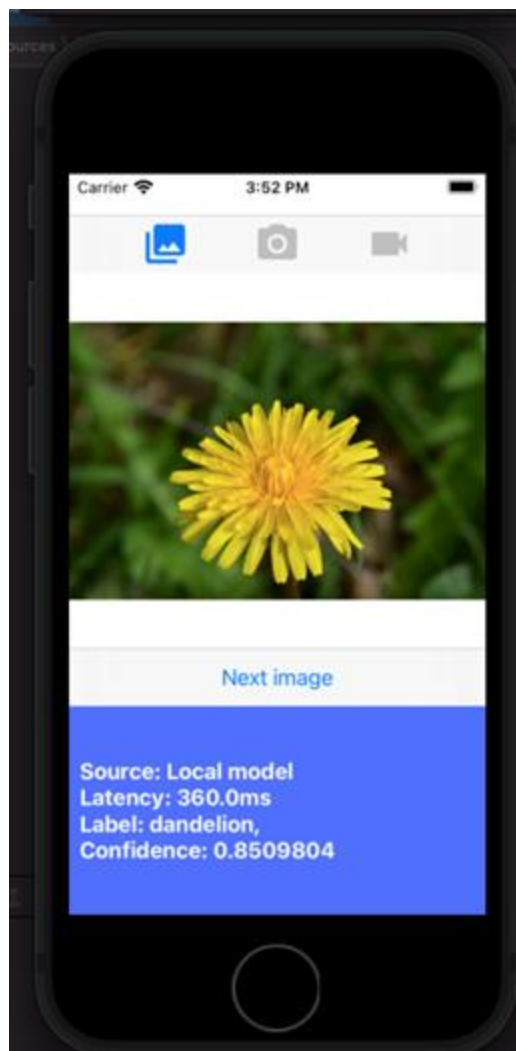
Training model...

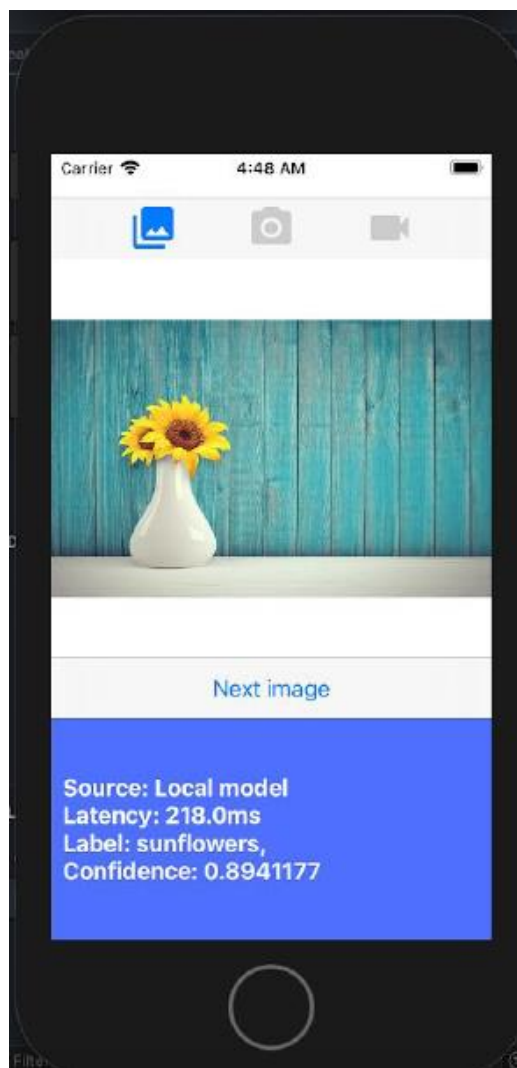
CANCEL





## 6. Run the MLVisionExample.app in XCode using IOS App Simulator





## **PART 2b**

**End to end deployment of a Vision  
Model using AutoML to mobile device**

## 1. Setting up Notebook

### New notebook instance

Instance name

63-char limit with lowercase letters, digits, or '-' only. Must start with a letter. Cannot end with a '-'.

Region \*

Zone \*

#### Instance properties

Environment ?	TensorFlow Enterprise 2.1 (with 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
Subnetwork	<input type="text" value="default(10.138.0.0/20)"/>
External IP	Ephemeral(Automatic)
Extensions ?	<input type="button" value="SELECT EXTENSIONS"/> None selected
Permission	Compute Engine default service account

ADVANCED OPTIONS

Google Cloud Platform

My Project 48541

Search products and resources

AI Platform

Notebooks

NEW INSTANCE

REFRESH

START

STOP

RESET

UPGRADE

DELETE

HIDE INFO PANEL

Dashboard

AI Hub

Data Labeling

Notebooks

Pipelines

Jobs

Models

Create and use Jupyter Notebooks with a notebook instance. Notebook instances have JupyterLab pre-installed and are configured with GPU-enabled machine learning frameworks. [Learn more](#)

Filter

Enter property name or value

	Instance name	Zone	Environment	Machine type	GPUs	Permission	Labels
	<div><div></div><div></div></div>						
	<div><div></div><div></div></div>						

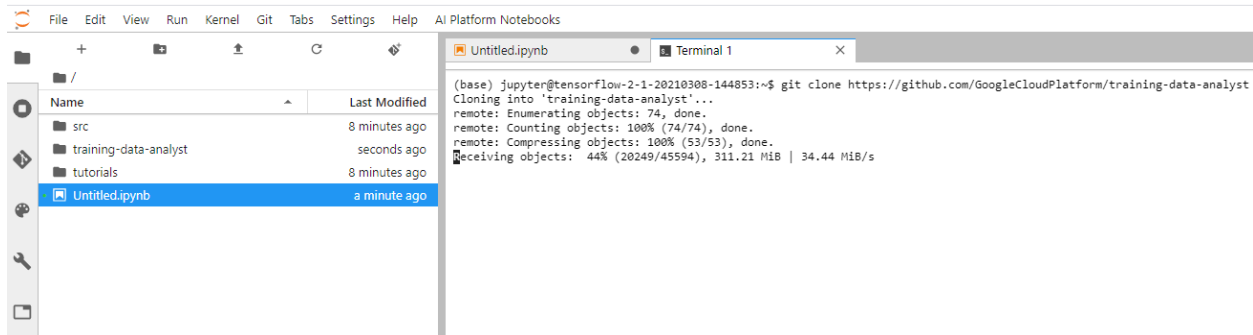
Info panel

DOCUMENTATION

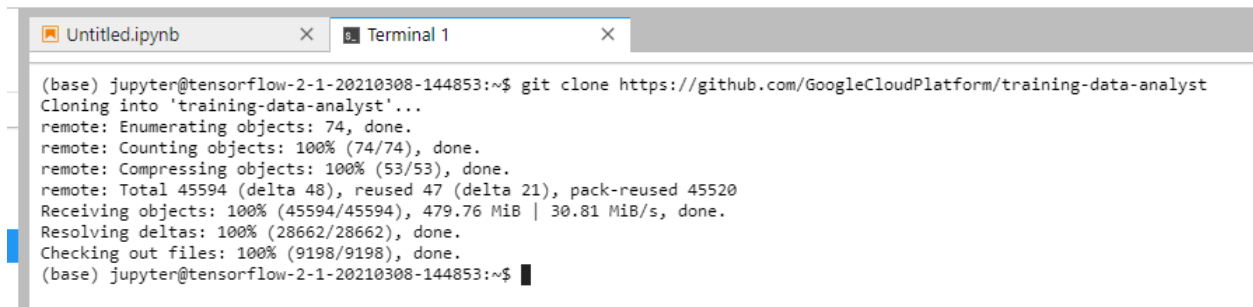
LABELS

[Notebook instances](#)

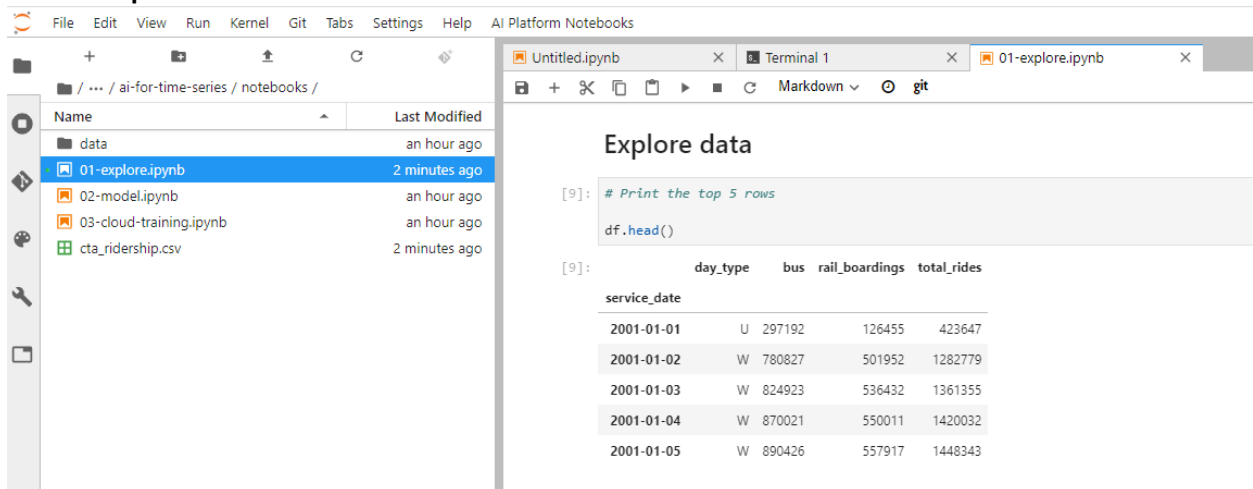
[Notebook API](#)

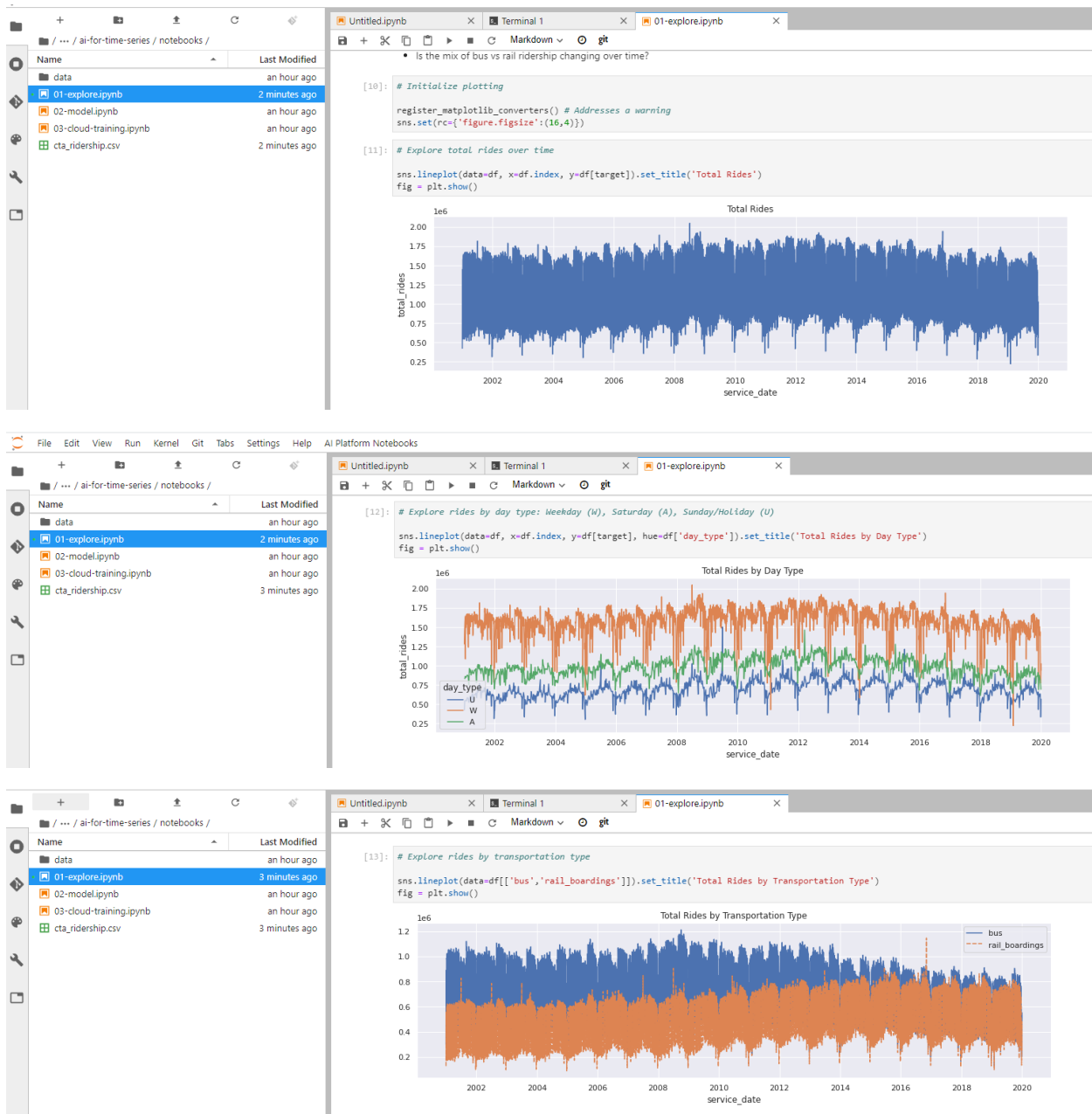


### AI Platform Notebooks



## 3. Explore and Visualize Data





#### 4. Creating a Model with BigQuery Time Series Forecasting

Google Cloud Platform My Project 48541 Search products and resources

FEATURES & INFO SHORTCUT HIDE PREVIEW FEATURES

Explorer + ADD DATA

Type to search

Viewing pinned projects.

resonant-kayak-306121

resonant-kayak-306121

Resources in this dataset

Use the Explorer panel to view your data, or click the Dataset option above.

Create dataset

Dataset ID: demo

Data location (Optional): Default

Default table expiration: ☒ Never ☐ Number of days after table creation:

Encryption: ☒ Google-managed key ☐ No configuration required ☐ Customer-managed key

Create dataset Cancel

Google Cloud Platform My Project 48541 Search products and resources

FEATURES & INFO SHORTCUT HIDE PREVIEW FEATURES

Explorer + ADD DATA

Type to search

Viewing pinned projects.

resonant-kayak-306121

demo

Description: None

Dataset info

Dataset ID: resonant-kayak-306121:demo

Created: Mar 8, 2021, 4:09:14 PM

Default table expiration: Never

Last modified: Mar 8, 2021, 4:09:14 PM

Data location: US

Create table

Source: Create table from: Upload Select file: cta\_ridership.csv File format: CSV

Destination: ☒ Search for a project ☐ Enter a project name

Project name: My Project 48541 Dataset name: demo Table type: Native table

Table name: cta\_ridership

Schema: Auto detect ☒ Schema and input parameters Schema will be automatically generated.

Partition and cluster settings: Partitioning: No partitioning

Clustering order (optional): Clustering order determines the sort order of the data. Clustering can be used on both partitioned and non-partitioned tables. Comma-separated list of fields to define clustering order (up to 4)

Advanced options

Create table Cancel



Google Cloud PlatformMy Project 48541

Search products and resources

FEATURES & INFO

SHORTCUT

HIDE PREVIEW FEATURES

Explorer

+ ADD DATA

Q Type to search

Viewing pinned projects.

resonant-kayak-306121

demo

CANCEL

SAVE

SCHEDULE

MORE

1

2

3

4

5

6

7

8

9

CREATE OR REPLACE MODEL

'demo.cta\_ridership\_model' OPTIONS(MODEL\_TYPE='ARIMA',

TIME\_SERIES\_TIMESTAMP\_COL='service\_date',

TIME\_SERIES\_DATA\_COL='total\_rides',

HOLIDAY\_REGION='us') AS

SELECT

service\_date, total\_rides

FROM

'demo.cta\_ridership'

Query results

Job information

Results

Execution details

This statement will create a new model named resonant-kayak-306121:demo.cta\_ridership\_model. Depending on the type of model, this may take several hours to complete.

Query results

Query complete (59.6 sec elapsed, 4.4 MB (ML) processed)

Job information

Results

Execution details

This statement will create a new model named resonant-kayak-306121:demo.cta\_ridership\_model. Depending on the type of model, this may take several hours to complete.

[illegible]

Google Cloud Platform My Project 48541 Search products and resources

Explorer + ADD DATA

Query results

Query complete (0.2 sec elapsed, 23.4 KB processed)

Row	forecast_timestamp	forecast_value	standard_error	confidence_level	prediction_interval_lower_bound	prediction_interval_upper_bound	confidence_interval_lower_bound	confidence_interval_upper_bound
1	2020-01-01 00:00:00 UTC	662436.4424369269	46059.49014554253	0.95	572322.980240453	752549.9046334007	572322.980240453	752549.9046334007
2	2020-01-02 00:00:00 UTC	1029641.4669424891	46276.328347693256	0.95	939103.76989082	1120179.1639941582	939103.76989082	1120179.1639941582
3	2020-01-03 00:00:00 UTC	1201660.2034356925	47233.43871922012	0.95	1109249.9600529654	1294070.4468184195	1109249.9600529654	1294070.4468184195
4	2020-01-04 00:00:00 UTC	651095.9776391207	48157.99332862347	0.95	556876.8819095747	745315.0733686666	556876.8819095747	745315.0733686666
5	2020-01-05 00:00:00 UTC	467394.91846646497	48621.50963880497	0.95	372268.97250121285	562520.8644317171	372268.97250121285	562520.8644317171
6	2020-01-06 00:00:00 UTC	1158999.319539823	48869.23710364581	0.95	1063388.705171438	1254609.9339082083	1063388.705171438	1254609.9339082083
7	2020-01-07 00:00:00 UTC	1127789.5651062205	49011.66149084522	0.95	1031900.303930386	1223678.8268194026	1031900.303930386	1223678.8268194026

## 5. Building a Custom Forecasting Model

File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks

Untitled.ipynb 01-explore.ipynb 02-model.ipynb

Import libraries and define constants

```
[4]: import os

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

from google.cloud import storage
from pandas.plotting import register_matplotlib_converters
from sklearn.metrics import r2_score, mean_absolute_error, mean_absolute_percentage_error, mean_squared_error
from sklearn.preprocessing import StandardScaler
from statsmodels.tsa.holtwinters import ExponentialSmoothing
from tensorflow.keras import Sequential
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.layers import Conv1D, Dense, Dropout, Flatten, LSTM, MaxPooling1D

register_matplotlib_converters() # Address warning

[5]: # Enter your project, region, and a bucket name. Then run the cell to make sure the
# Cloud SDK uses the right project for all the commands in this notebook.

PROJECT = 'My Project 48541' # REPLACE WITH YOUR PROJECT ID
BUCKET = 'resonant-kayak-306121' # REPLACE WITH A UNIQUE BUCKET NAME e.g. your PROJECT NAME
REGION = 'us-central1' # REPLACE WITH YOUR BUCKET REGION e.g. us-central1
BUCKET_URI = 'gs://' + BUCKET

#Don't change the following command - this is to check if you have changed the project name above.
assert PROJECT != 'your-project-name', 'Don't forget to change the project variables!'

[6]: # Dataset parameters

target_col = 'total_rides' # The variable you are predicting
ts_col = 'service_date' # The name of the column with the date field
```

File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks

Untitled.ipynb 01-explore.ipynb 02-model.ipynb

## ML Models

In this section, you will build models using popular neural network architectures for time-series data.

### Long Short Term Memory (LSTM)

```
[21]: # Reshape test data to match model inputs and outputs
X_train = X_train_reformatted.values.reshape(-1, n_input_steps, n_features)
X_test = X_test_reformatted.values.reshape(-1, n_input_steps, n_features)
y_train = y_train_reformatted.values.reshape(-1, n_output_steps, 1)
y_test = y_test_reformatted.values.reshape(-1, n_output_steps, 1)
```

**TODO 2: Update the LSTM architecture**

Try increasing and decreasing the number of LSTM units and see if you notice any accuracy improvements.

You can use hyper-parameter tuning to search for optimal values, but that's outside the scope of this lab.

```
[22]: # Try increasing and decreasing the number of LSTM units and see if you notice any accuracy improvements.
# Run the next cell to evaluate the results in more detail.

model = Sequential([
    LSTM(64, input_shape=[n_input_steps, n_features]),
    Dense(n_output_steps)])

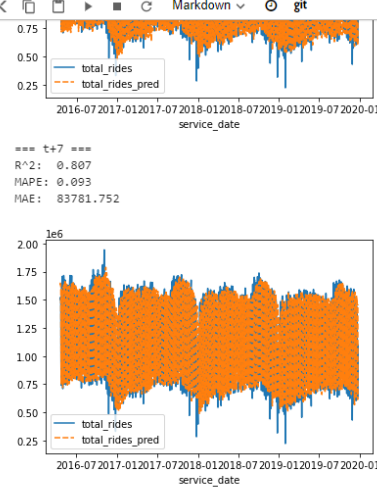
model.compile(optimizer='adam', loss='mae')

early_stopping = EarlyStopping(monitor='val_loss', patience=patience)
_ = model.fit(x=X_train, y=y_train, validation_data=(X_test, y_test), epochs=epochs, callbacks=[early_stopping])

Train on 5515 samples, validate on 1352 samples
Epoch 1/1000
5515/5515 [=====] - 12s 2ms/sample - loss: 0.6263 - val_loss: 0.3976
Epoch 2/1000
5515/5515 [=====] - 7s 1ms/sample - loss: 0.3115 - val_loss: 0.2579
Epoch 3/1000
5515/5515 [=====] - 6s 1ms/sample - loss: 0.2373 - val_loss: 0.2346
Epoch 4/1000
5515/5515 [=====] - 6s 1ms/sample - loss: 0.2235 - val_loss: 0.2326
Epoch 5/1000
5515/5515 [=====] - 6s 1ms/sample - loss: 0.2193 - val_loss: 0.2256
```

File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks

Untitled.ipynb 01-explore.ipynb 02-model.ipynb



```
=== t+7 ===
R^2: 0.807
MAPE: 0.093
MAE: 83781.752
```

## Conclusion

Great job! You've now completed the modeling portion of this workshop. You've covered:

- Removing outliers from the data
- Multi-step forecasting
- Neural network architectures for time-series forecasting: LSTM and CNN
- Statistical models, including Holt-Winters Exponential Smoothing
- Ensembling models

**Refer to notebooks 01-explore.ipynb, 02-model.ipynb and 03-cloud-training.ipynb checked in Github.**