CMPE-258 Deep Learning

Assignment – 2

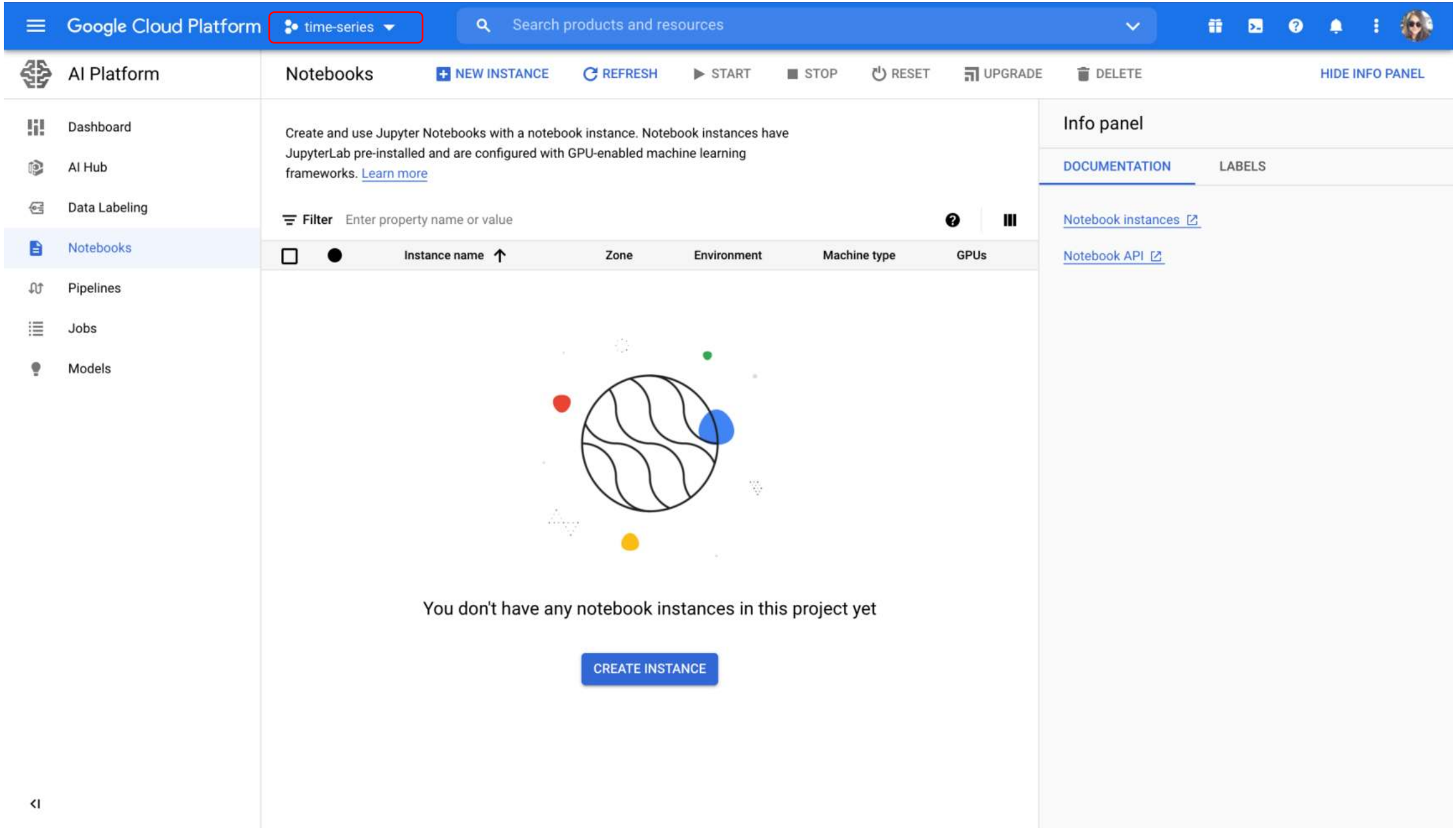
By,

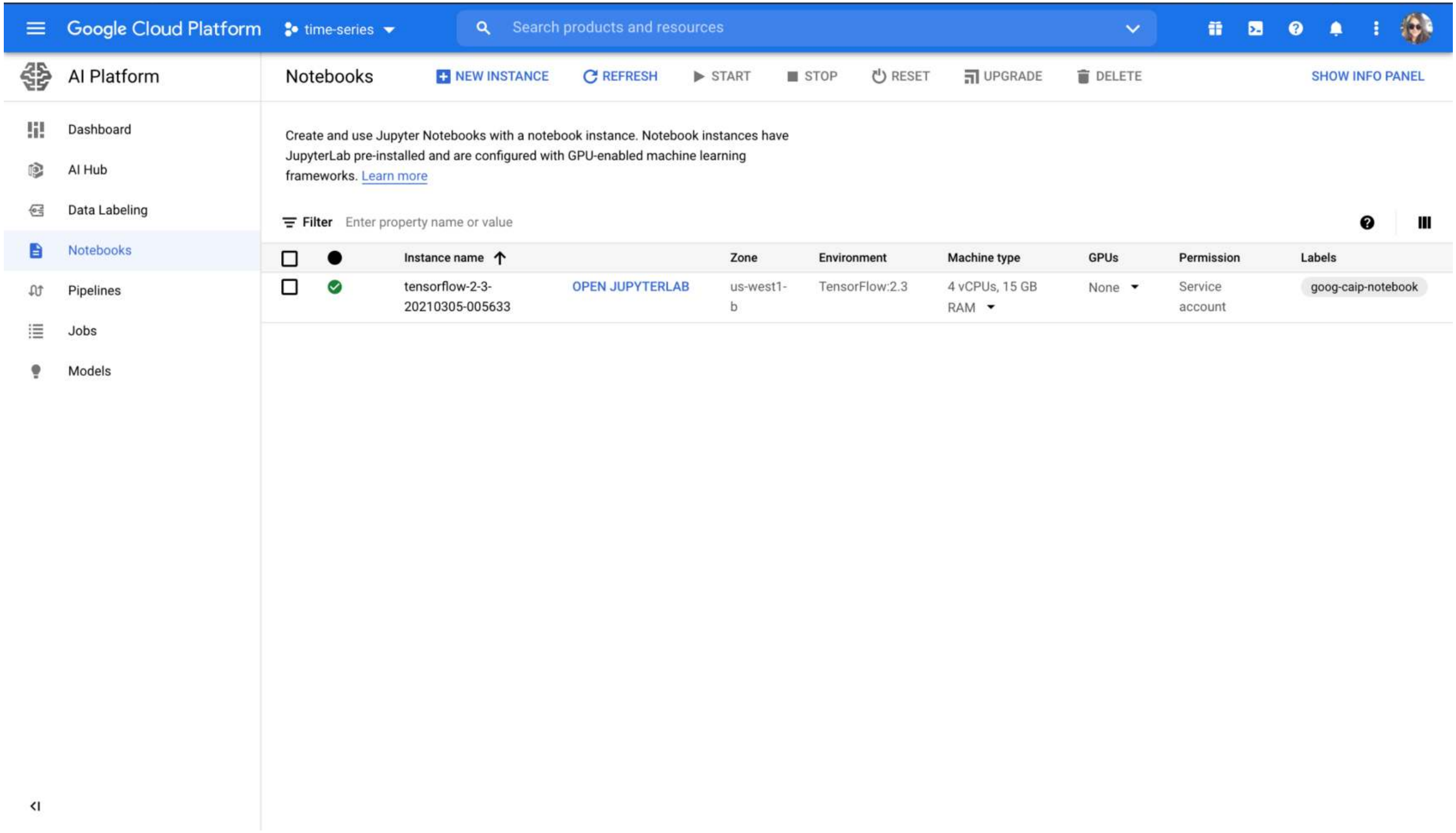
Sheema Murugesh Babu

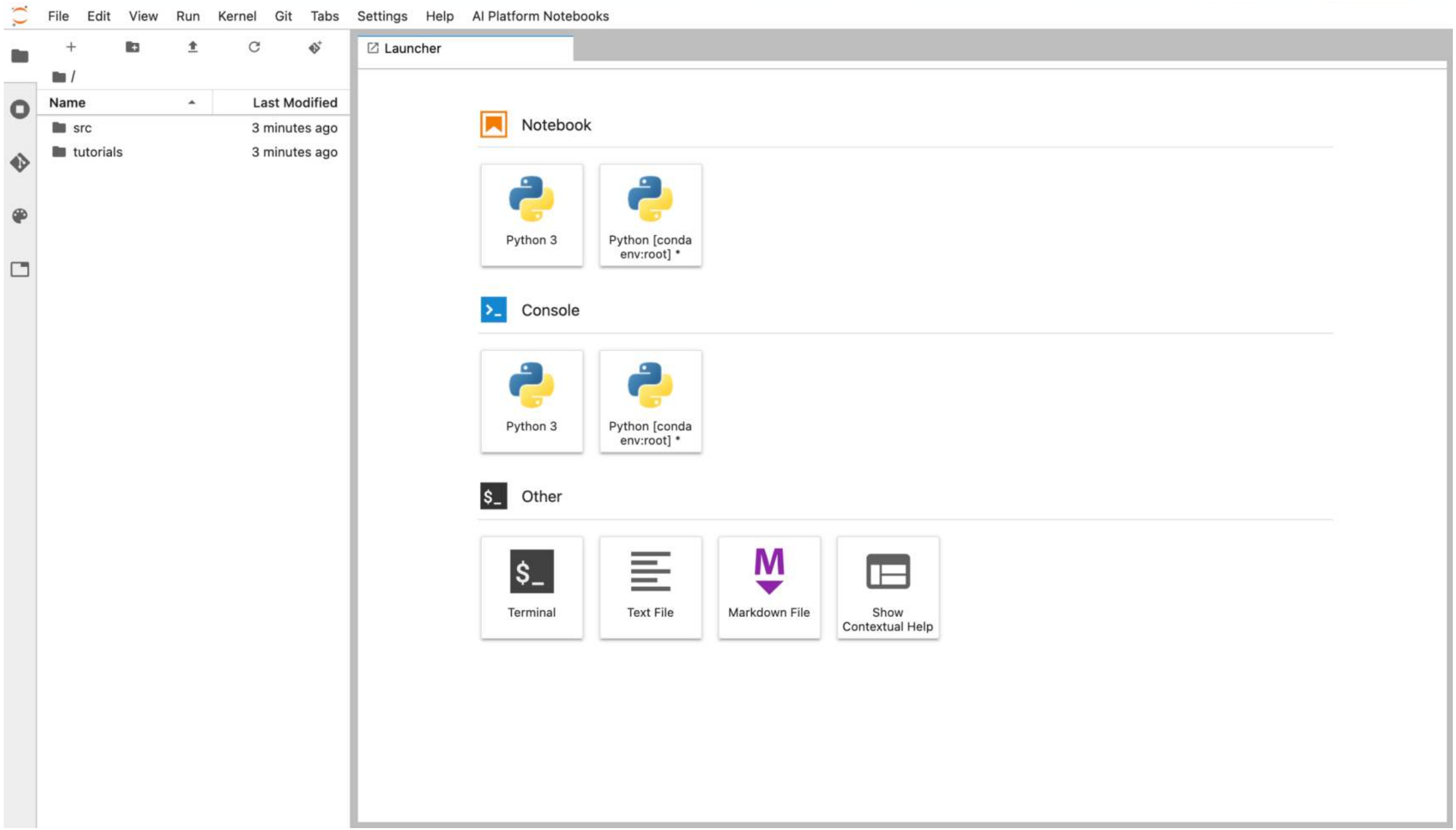
(SJSU ID: 015217176)

Part 2: AutoML Models

TIME SERIES


Google Cloud Platform
time-series
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
You don't have any notebook instances in this project yet
CREATE INSTANCE
Info panel
DOCUMENTATION
LABELS
Notebook instances
Notebook API


Google Cloud Platform
time-series
Search products and resources
AI Platform
Notebooks
NEW INSTANCE
REFRESH
START
STOP
RESET
UPGRADE
DELETE
SHOW 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
tensorflow-2-3-20210305-005633
OPEN JUPYTERLAB
us-west1-b
TensorFlow:2.3
4 vCPUs, 15 GB RAM
None
Service account
goog-caip-notebook


File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks
/
Name
Last Modified
src
3 minutes ago
tutorials
3 minutes ago
Launcher
Notebook
Python 3
Python [conda env:root] *
Console
Python 3
Python [conda env:root] *
Other
Terminal
Text File
Markdown File
Show Contextual Help

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
data	18 minutes ago
01-explore.ipynb	seconds ago
02-model.ipynb	18 minutes ago
03-cloud-training.ip...	18 minutes ago
cta_ridership.csv	seconds ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb

Markdown git Python 3

```
sns.lineplot(data=df, x=df.index, y=df[target]).set_title('Total Rides')
fig = plt.show()
```

[23]:

```
# Explore rides by day type: Weekday (W), Saturday (A), Sunday/Holiday (U)

sns.lineplot(data=df, x=df.index, y=df[target], hue=df['day_type']).set_title('Total Rides by Day Type')
fig = plt.show()
```

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
data	18 minutes ago
01-explore.ipynb	seconds ago
02-model.ipynb	18 minutes ago
03-cloud-training.ip...	18 minutes ago
cta_ridership.csv	a minute ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb

Markdown git Python 3

```
sns.lineplot(data=df, x=df.index, y=df[target], hue=df['day_type']).set_title('Total Rides by Day Type')
fig = plt.show()
```

[24]:

```
# Explore rides by transportation type

sns.lineplot(data=df[['bus','rail_boardings']]).set_title('Total Rides by Transportation Type')
fig = plt.show()
```

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
data	18 minutes ago
01-explore.ipynb	a minute ago
02-model.ipynb	18 minutes ago
03-cloud-training.ip...	18 minutes ago
cta_ridership.csv	a minute ago

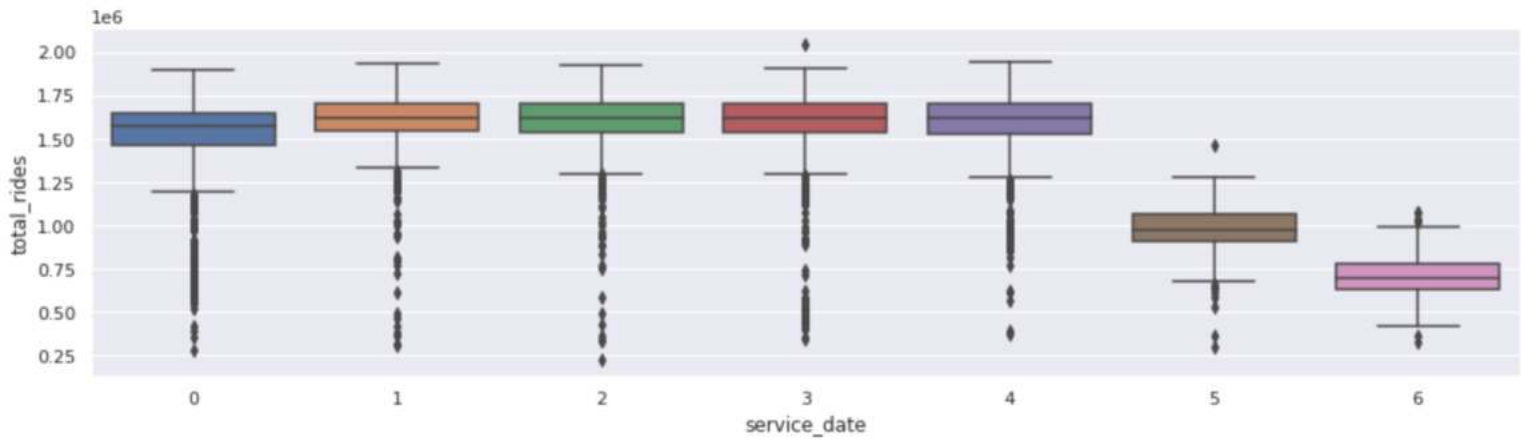
Untitled.ipynb | Terminal 1 | 01-explore.ipynb

Markdown git Python 3

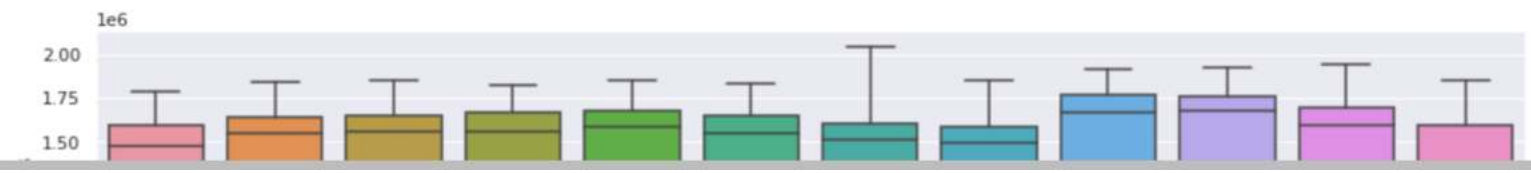
TODO 3: Explore seasonality

- Is there much difference between months?
- Can you extract the trend and seasonal pattern from the data?

[26]:
```
# Show the distribution of values for each day of the week in a boxplot:
# Min, 25th percentile, median, 75th percentile, max

daysofweek = df.index.to_series().dt.dayofweek

fig = sns.boxplot(x=daysofweek, y=df[target])
```



[27]:
```
# Show the distribution of values for each month in a boxplot:

months = df.index.to_series().dt.month

fig = sns.boxplot(x=months, y=df[target])
```



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

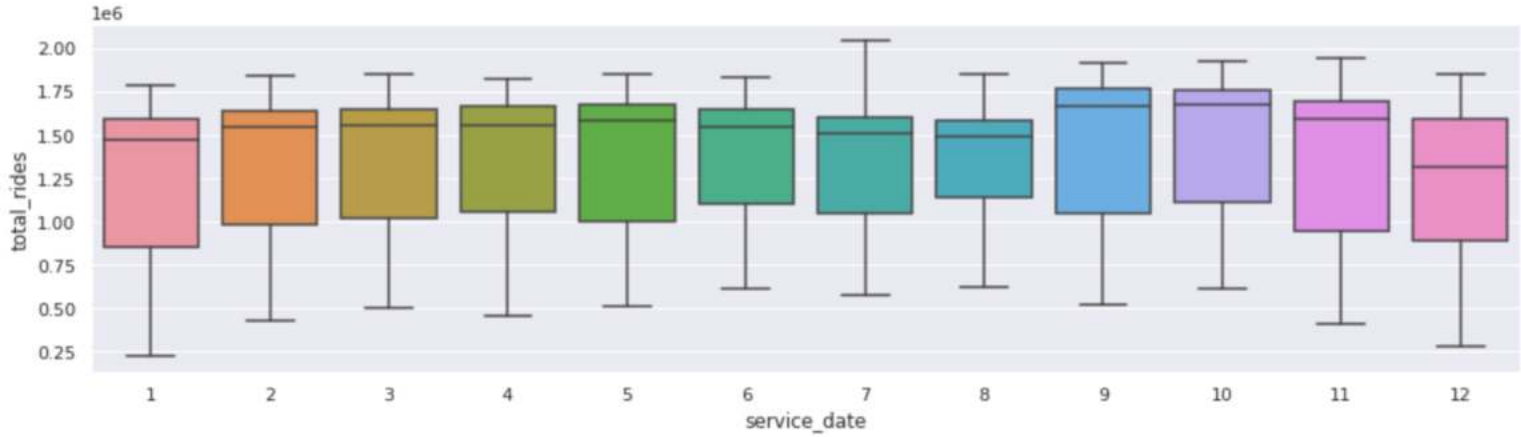
/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
data	19 minutes ago
01-explore.ipynb	a minute ago
02-model.ipynb	19 minutes ago
03-cloud-training.ip...	19 minutes ago
cta_ridership.csv	a minute ago

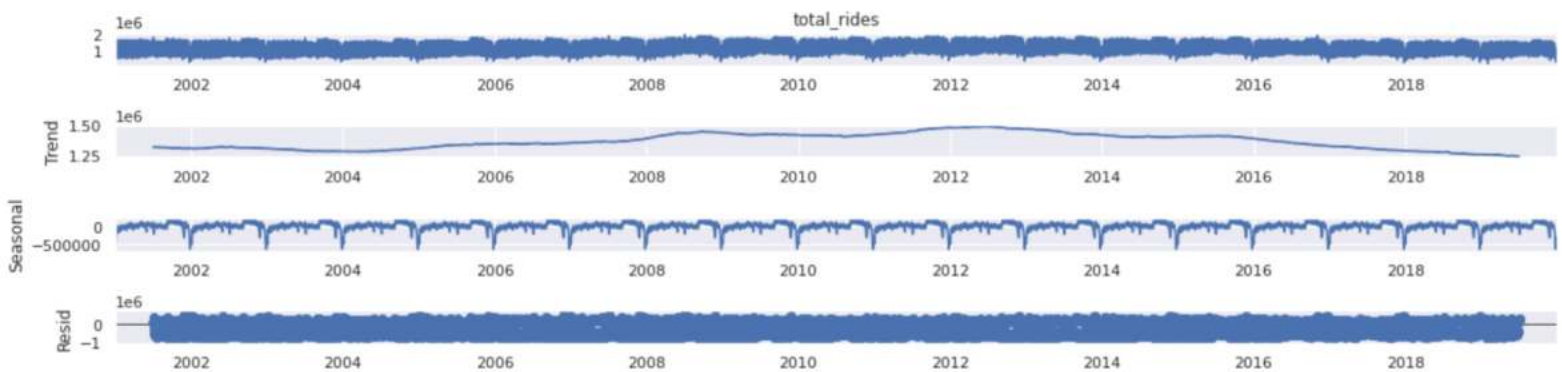
Untitled.ipynb | Terminal 1 | 01-explore.ipynb

Markdown git Python 3

```
fig = sns.boxplot(x=months, y=df[target])
```



[28]:
```
# Decompose the data into trend and seasonal components

result = seasonal_decompose(df[target], period=365)
fig = result.plot()
```



Auto-correlation

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
data	19 minutes ago
01-explore.ipynb	a minute ago
02-model.ipynb	19 minutes ago
03-cloud-training.ip...	19 minutes ago
cta_ridership.csv	a minute ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb

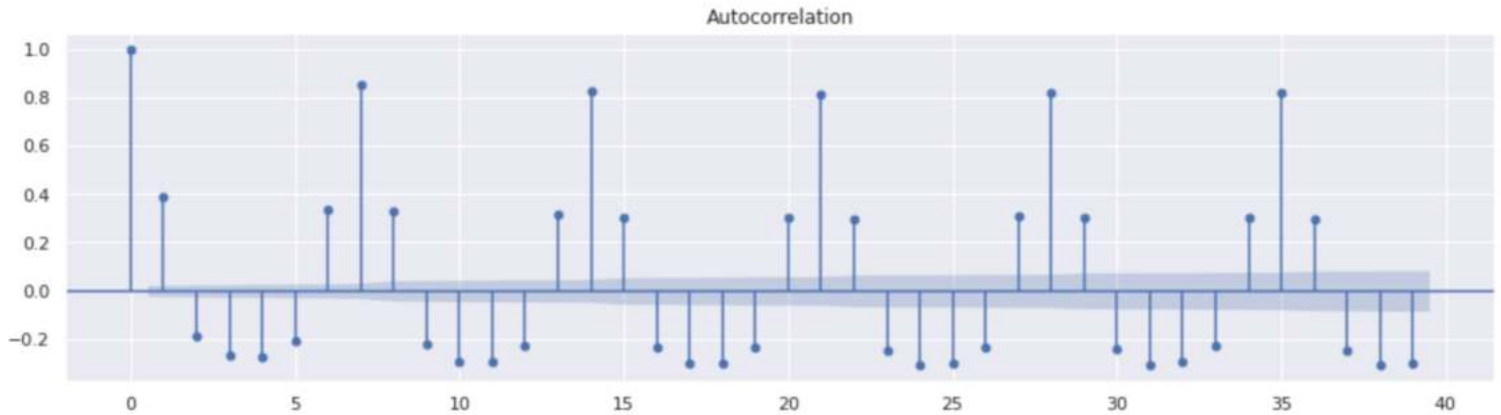
Markdown git Python 3

Auto-correlation

Next, we will create an auto-correlation plot, to show how correlated a time-series is with itself. Each point on the x-axis indicates the correlation at a given lag. The shaded area indicates the confidence interval.

Note that the correlation gradually decreases over time, but reflects weekly seasonality (e.g. `t-7` and `t-14` stand out).

[29]:
```
plot_acf(df[target])

fig = plt.show()
```



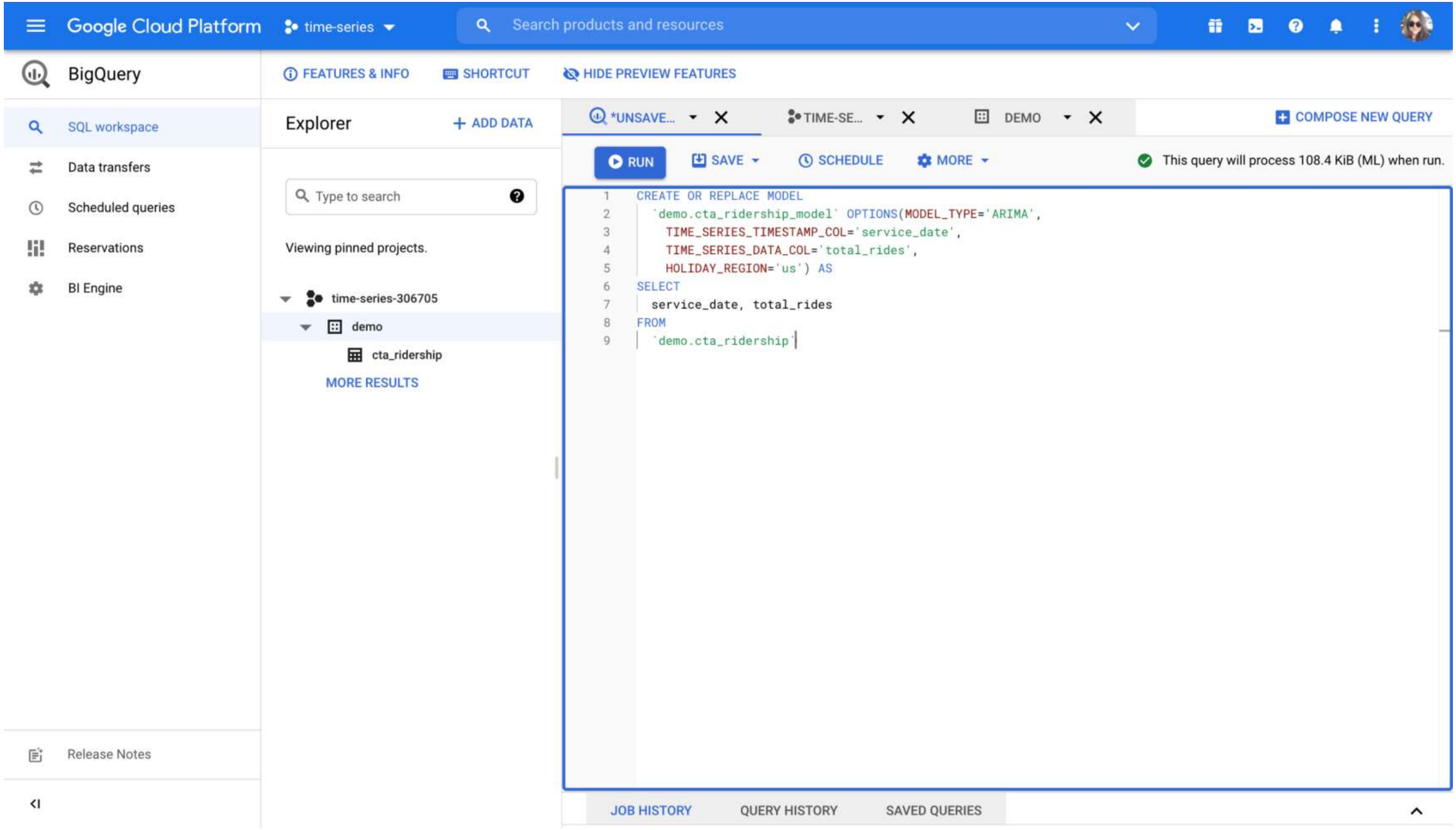
Export data

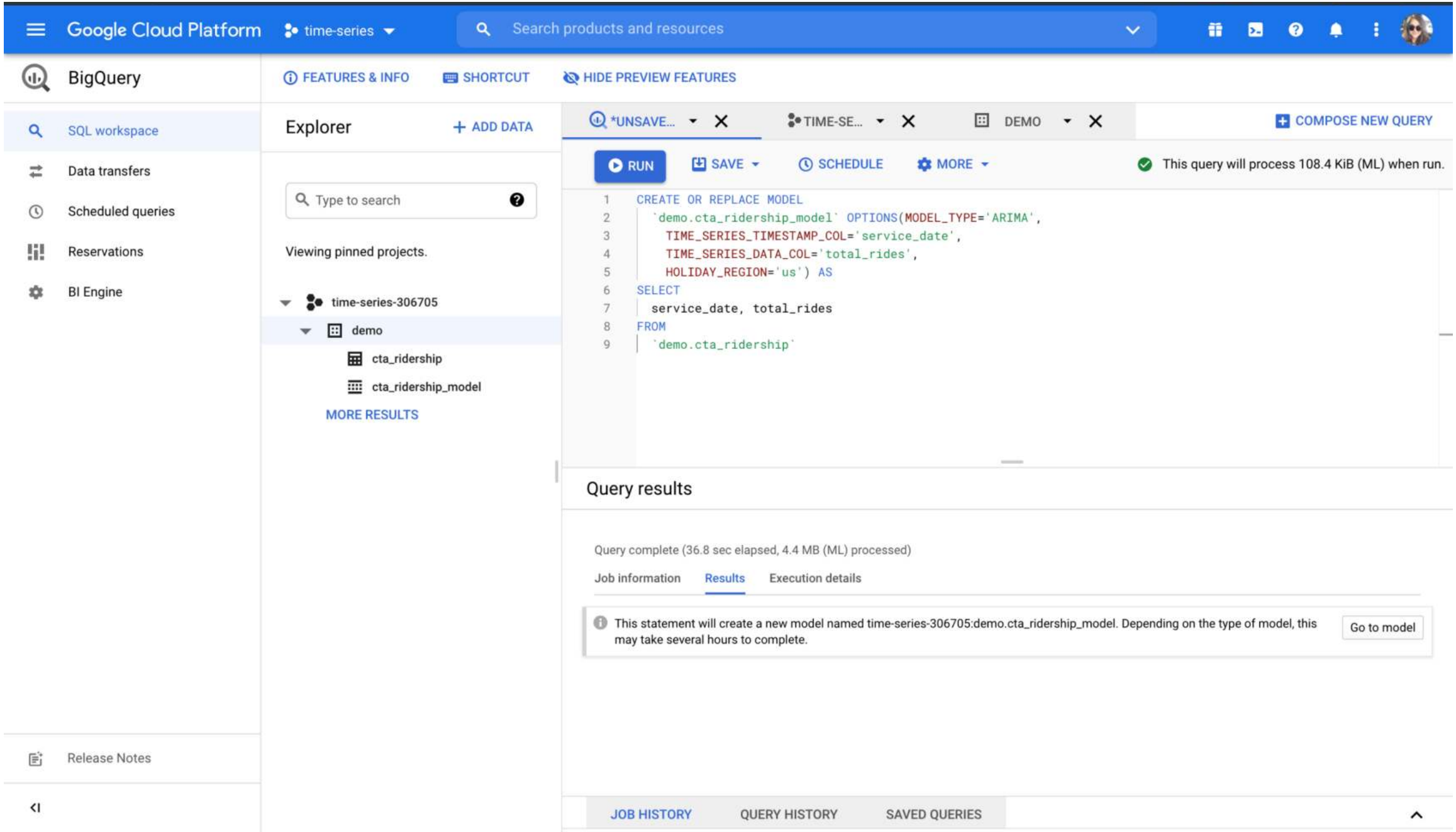
This will generate a CSV file, which you will use in the next labs of this quest. Inspect the CSV file to see what the data looks like.

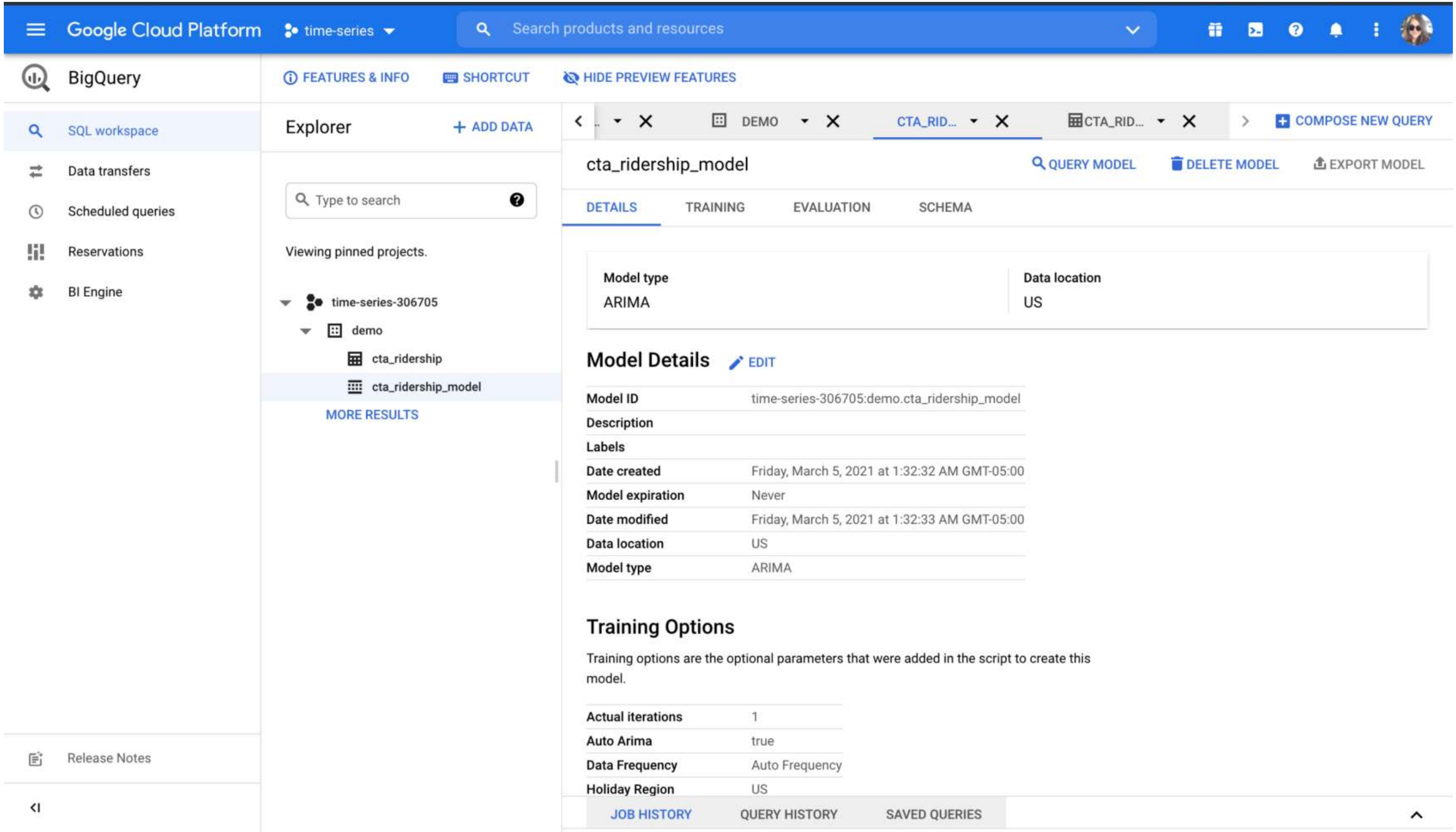
[30]:
```
df[[target]].to_csv(processed_file, index=True, index_label=ts_col)
```

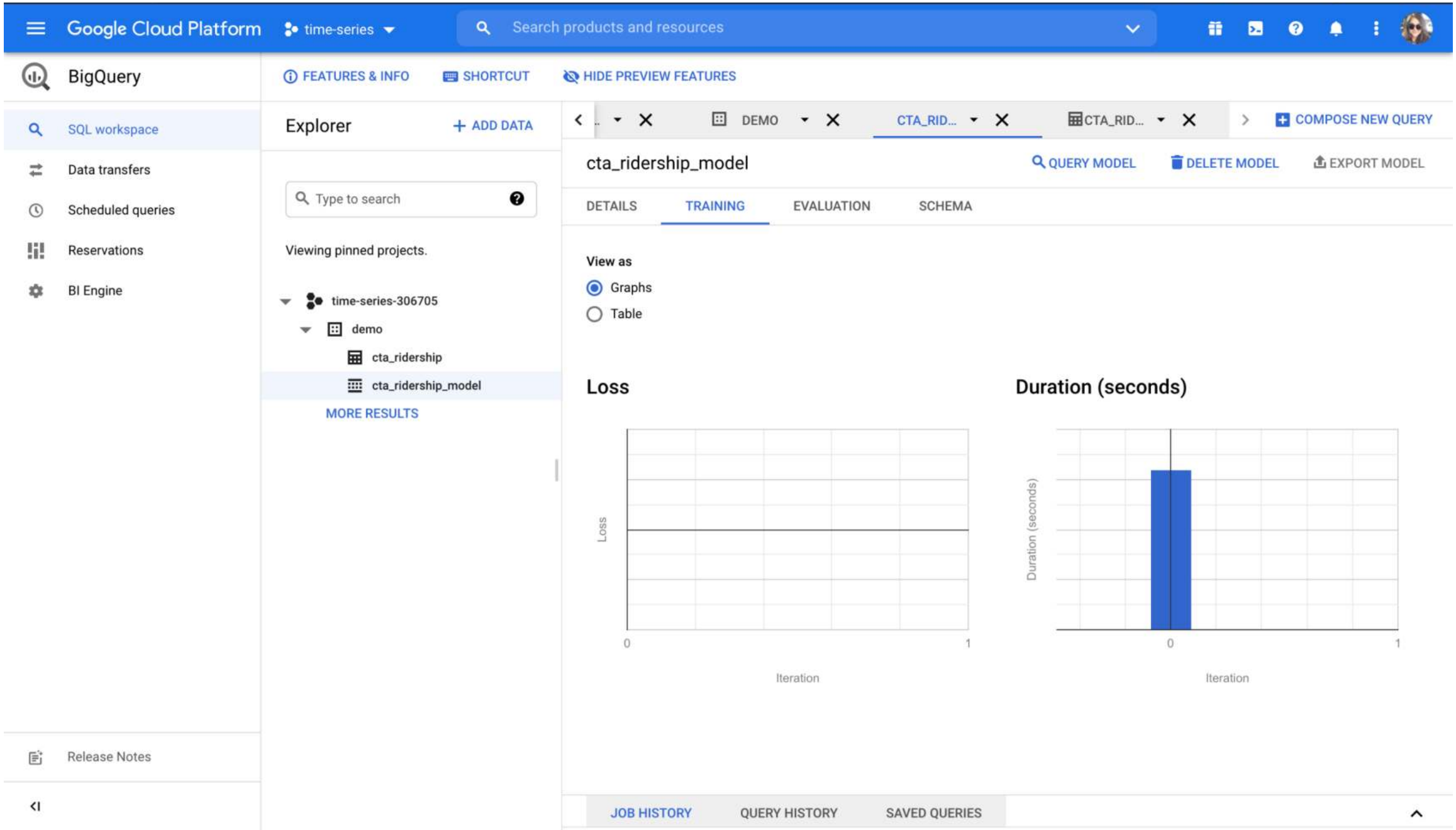
Conclusion

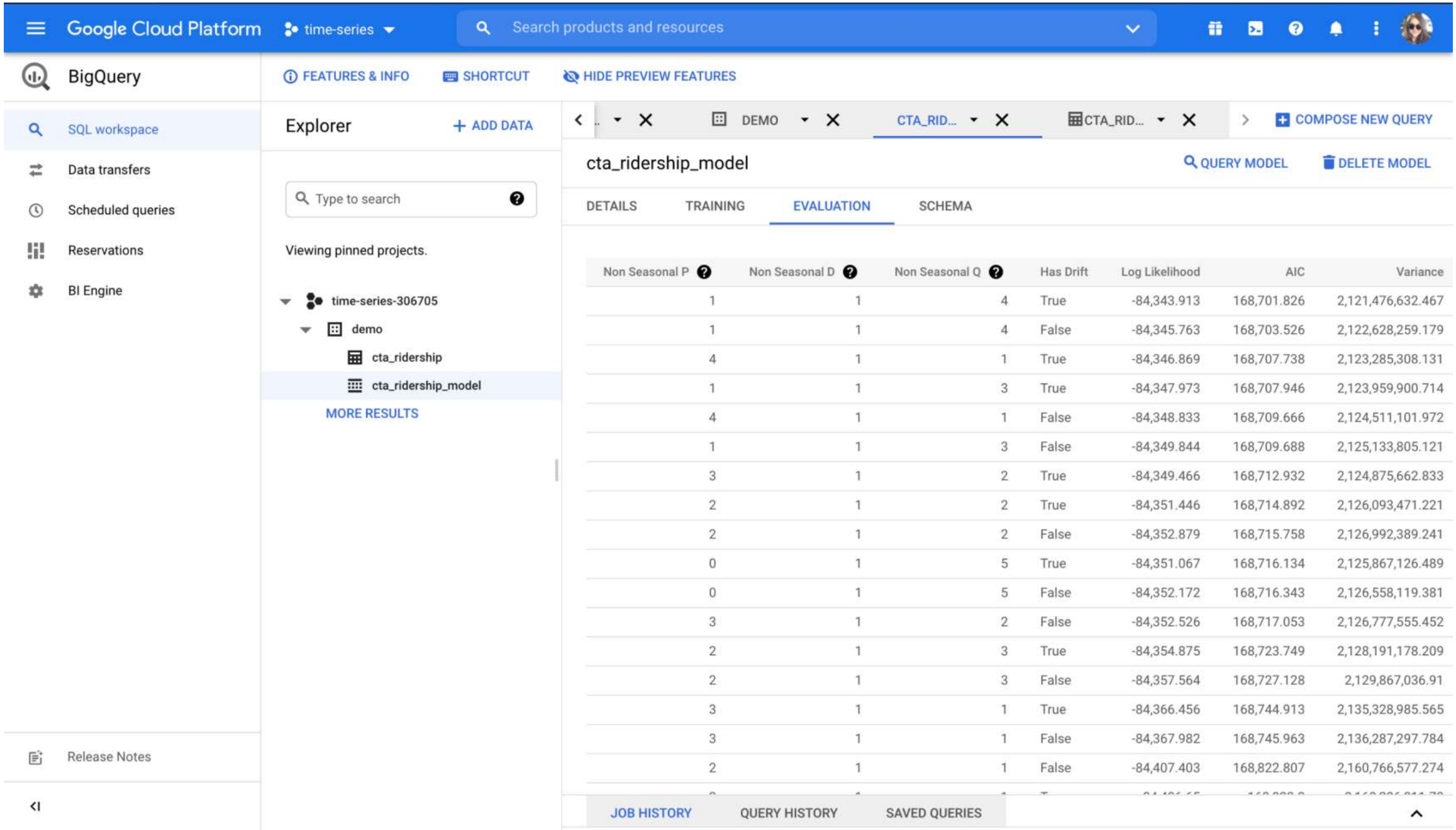
You've successfully completed the exploration and visualization lab. You've learned how to:


Google Cloud Platform
time-series
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
HIDE PREVIEW FEATURES
SQL workspace
Data transfers
Scheduled queries
Reservations
BI Engine
Release Notes
Explorer
ADD DATA
Type to search
Viewing pinned projects.
time-series-306705
demo
cta_ridership
MORE RESULTS
*UNSAVE...
TIME-SE...
DEMO
COMPOSE NEW QUERY
RUN
SAVE
SCHEDULE
MORE
This query will process 108.4 KiB (ML) when run.
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`
JOB HISTORY
QUERY HISTORY
SAVED QUERIES

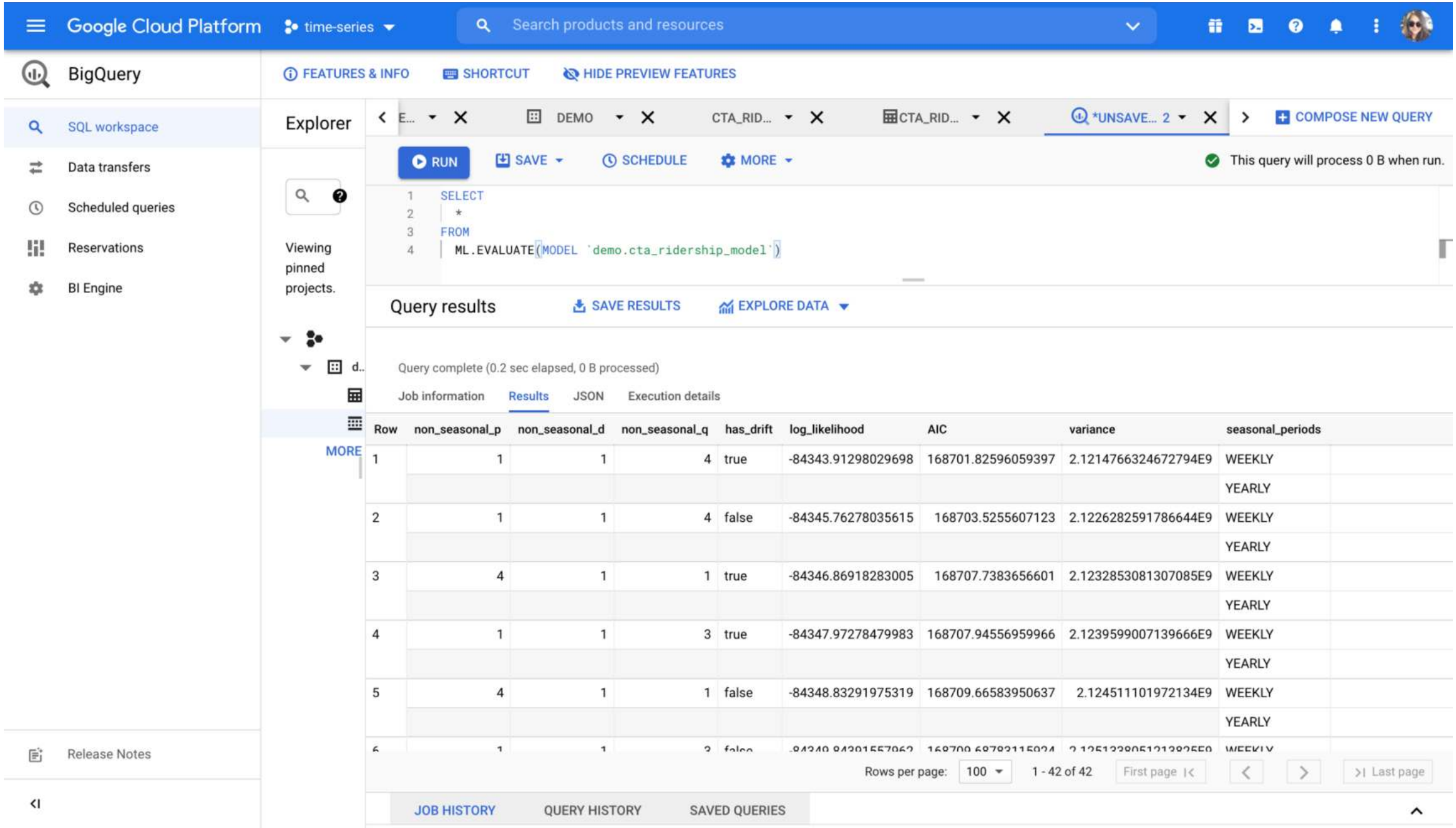

Google Cloud Platform
time-series
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
HIDE PREVIEW FEATURES
SQL workspace
Data transfers
Scheduled queries
Reservations
BI Engine
Release Notes
Explorer
ADD DATA
Type to search
Viewing pinned projects.
time-series-306705
demo
cta_ridership
cta_ridership_model
MORE RESULTS
*UNSAVE...
TIME-SE...
DEMO
COMPOSE NEW QUERY
RUN
SAVE
SCHEDULE
MORE
This query will process 108.4 KiB (ML) when run.
1 CREATE OR REPLACE MODEL
2 `demo.cta_ridership_model` OPTIONS(MODEL_TYPE='ARIMA',
3 TIME_SERIES_TIMESTAMP_COL='service_date',
4 TIME_SERIES_DATA_COL='total_rides',
5 HOLIDAY_REGION='us') AS
6 SELECT
7 service_date, total_rides
8 FROM
9 `demo.cta_ridership`
Query results
Query complete (36.8 sec elapsed, 4.4 MB (ML) processed)
Job information
Results
Execution details
This statement will create a new model named time-series-306705:demo.cta_ridership_model. Depending on the type of model, this may take several hours to complete.
Go to model
JOB HISTORY
QUERY HISTORY
SAVED QUERIES

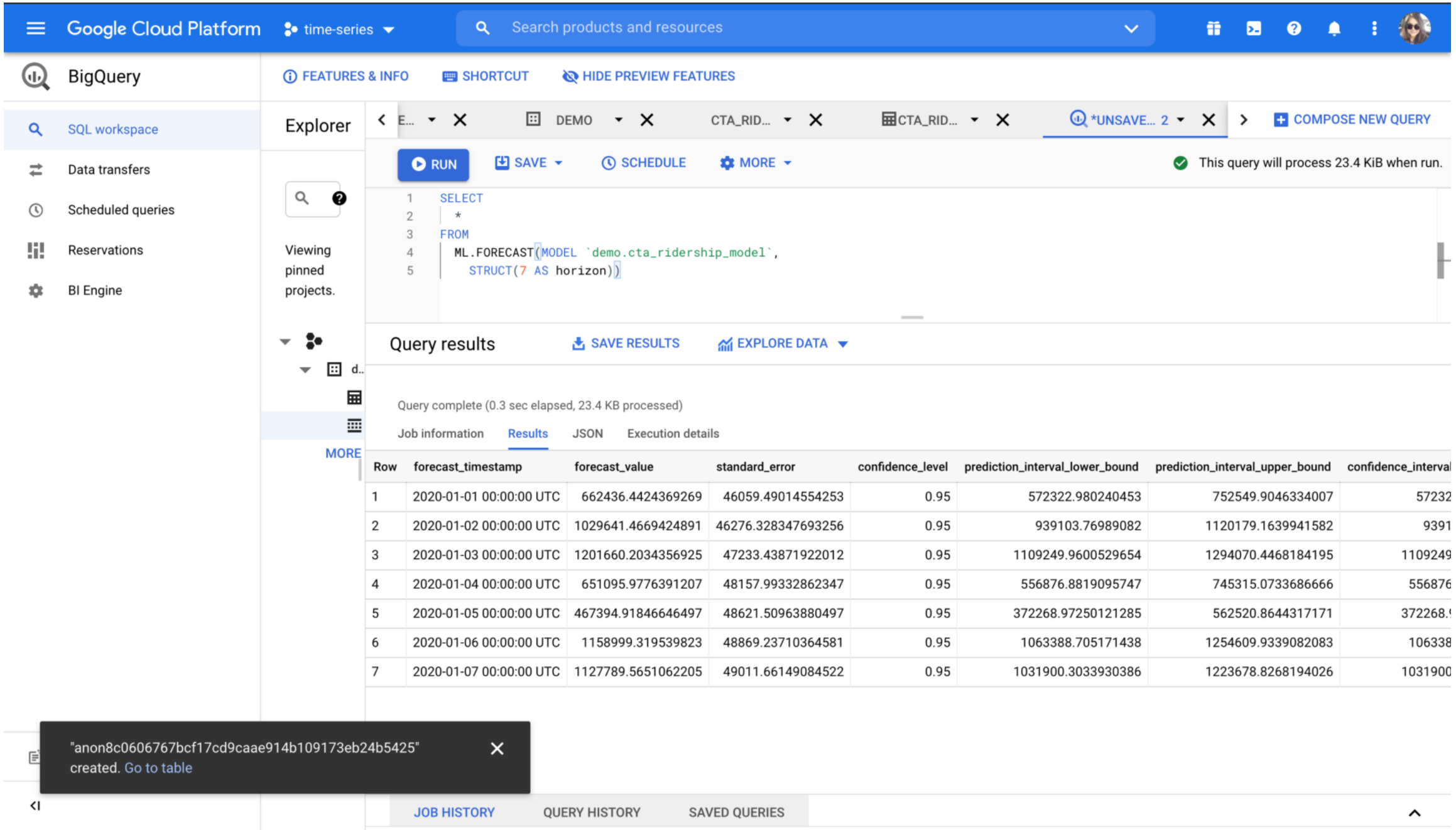

Google Cloud Platform
time-series
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
HIDE PREVIEW FEATURES
SQL workspace
Data transfers
Scheduled queries
Reservations
BI Engine
Release Notes
Explorer
ADD DATA
Type to search
Viewing pinned projects.
time-series-306705
demo
cta_ridership
cta_ridership_model
MORE RESULTS
DEMO
CTA_RID...
CTA_RID...
COMPOSE NEW QUERY
cta_ridership_model
QUERY MODEL
DELETE MODEL
EXPORT MODEL
DETAILS
TRAINING
EVALUATION
SCHEMA
Model type
ARIMA
Data location
US
Model Details
EDIT
Model ID
time-series-306705:demo.cta_ridership_model
Description
Labels
Date created
Friday, March 5, 2021 at 1:32:32 AM GMT-05:00
Model expiration
Never
Date modified
Friday, March 5, 2021 at 1:32:33 AM GMT-05:00
Data location
US
Model type
ARIMA
Training Options
Training options are the optional parameters that were added in the script to create this model.
Actual iterations
1
Auto Arima
true
Data Frequency
Auto Frequency
Holiday Region
US
JOB HISTORY
QUERY HISTORY
SAVED QUERIES


Google Cloud Platform
time-series
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
HIDE PREVIEW FEATURES
SQL workspace
Data transfers
Scheduled queries
Reservations
BI Engine
Release Notes
Explorer
ADD DATA
Type to search
Viewing pinned projects.
time-series-306705
demo
cta_ridership
cta_ridership_model
MORE RESULTS
DEMO
CTA_RID...
CTA_RID...
COMPOSE NEW QUERY
cta_ridership_model
QUERY MODEL
DELETE MODEL
EXPORT MODEL
DETAILS
TRAINING
EVALUATION
SCHEMA
View as
Graphs
Table
Loss
Loss
0
1
Iteration
Duration (seconds)
Duration (seconds)
0
1
Iteration
JOB HISTORY
QUERY HISTORY
SAVED QUERIES

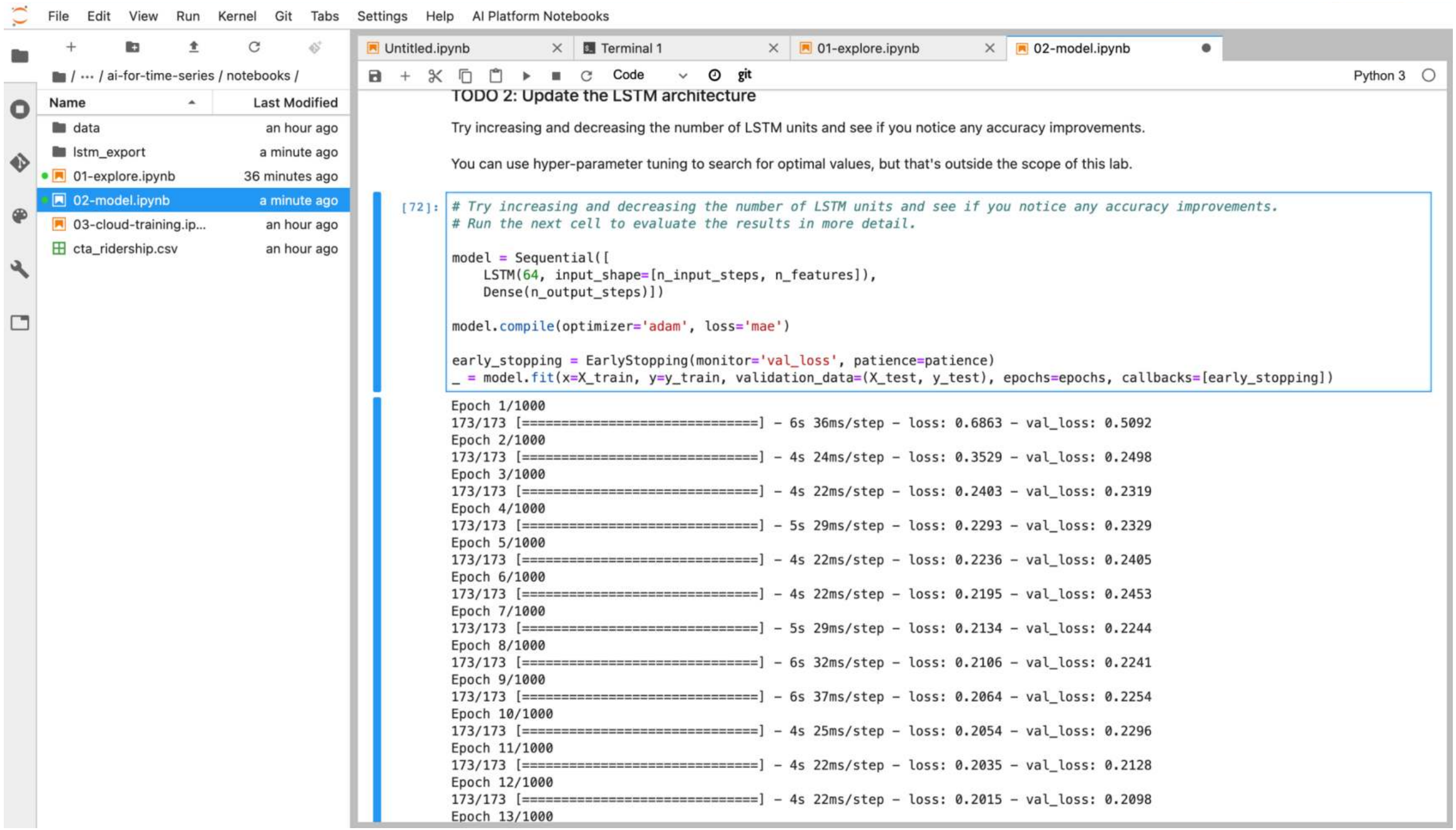


Non Seasonal P	Non Seasonal D	Non Seasonal Q	Has Drift	Log Likelihood	AIC	Variance
1	1	4	True	-84,343.913	168,701.826	2,121,476,632.467
1	1	4	False	-84,345.763	168,703.526	2,122,628,259.179
4	1	1	True	-84,346.869	168,707.738	2,123,285,308.131
1	1	3	True	-84,347.973	168,707.946	2,123,959,900.714
4	1	1	False	-84,348.833	168,709.666	2,124,511,101.972
1	1	3	False	-84,349.844	168,709.688	2,125,133,805.121
3	1	2	True	-84,349.466	168,712.932	2,124,875,662.833
2	1	2	True	-84,351.446	168,714.892	2,126,093,471.221
2	1	2	False	-84,352.879	168,715.758	2,126,992,389.241
0	1	5	True	-84,351.067	168,716.134	2,125,867,126.489
0	1	5	False	-84,352.172	168,716.343	2,126,558,119.381
3	1	2	False	-84,352.526	168,717.053	2,126,777,555.452
2	1	3	True	-84,354.875	168,723.749	2,128,191,178.209
2	1	3	False	-84,357.564	168,727.128	2,129,867,036.91
3	1	1	True	-84,366.456	168,744.913	2,135,328,985.565
3	1	1	False	-84,367.982	168,745.963	2,136,287,297.784
2	1	1	False	-84,407.403	168,822.807	2,160,766,577.274


Google Cloud Platform
time-series
Search products and resources
BigQuery
FEATURES & INFO
SHORTCUT
HIDE PREVIEW FEATURES
SQL workspace
Data transfers
Scheduled queries
Reservations
BI Engine
Release Notes
Explorer
Viewing pinned projects.
MORE
DEMO
CTA_RID...
*UNSAVE... 2
COMPOSE NEW QUERY
RUN
SAVE
SCHEDULE
MORE
This query will process 0 B when run.
SELECT
*
FROM
ML.EVALUATE(MODEL `demo.cta_ridership_model`)
Query results
SAVE RESULTS
EXPLORE DATA
Query complete (0.2 sec elapsed, 0 B processed)
Job information
Results
JSON
Execution details
Row | non_seasonal_p | non_seasonal_d | non_seasonal_q | has_drift | log_likelihood | AIC | variance | seasonal_periods
1 | 1 | 1 | 4 | true | -84343.91298029698 | 168701.82596059397 | 2.1214766324672794E9 | WEEKLY
YEARLY
2 | 1 | 1 | 4 | false | -84345.76278035615 | 168703.5255607123 | 2.1226282591786644E9 | WEEKLY
YEARLY
3 | 4 | 1 | 1 | true | -84346.86918283005 | 168707.7383656601 | 2.1232853081307085E9 | WEEKLY
YEARLY
4 | 1 | 1 | 3 | true | -84347.97278479983 | 168707.94556959966 | 2.1239599007139666E9 | WEEKLY
YEARLY
5 | 4 | 1 | 1 | false | -84348.83291975319 | 168709.66583950637 | 2.124511101972134E9 | WEEKLY
YEARLY
Rows per page: 100
1 - 42 of 42
First page
Last page
JOB HISTORY
QUERY HISTORY
SAVED QUERIES



Row	forecast_timestamp	forecast_value	standard_error	confidence_level	prediction_interval_lower_bound	prediction_interval_upper_bound	confidence_interval
1	2020-01-01 00:00:00 UTC	662436.4424369269	46059.49014554253	0.95	572322.980240453	752549.9046334007	57232
2	2020-01-02 00:00:00 UTC	1029641.4669424891	46276.328347693256	0.95	939103.76989082	1120179.1639941582	9391
3	2020-01-03 00:00:00 UTC	1201660.2034356925	47233.43871922012	0.95	1109249.9600529654	1294070.4468184195	1109249
4	2020-01-04 00:00:00 UTC	651095.9776391207	48157.99332862347	0.95	556876.8819095747	745315.0733686666	556876
5	2020-01-05 00:00:00 UTC	467394.91846646497	48621.50963880497	0.95	372268.97250121285	562520.8644317171	372268.9
6	2020-01-06 00:00:00 UTC	1158999.319539823	48869.23710364581	0.95	1063388.705171438	1254609.9339082083	106338
7	2020-01-07 00:00:00 UTC	1127789.5651062205	49011.66149084522	0.95	1031900.3033930386	1223678.8268194026	1031900


File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks
/ ··· / ai-for-time-series / notebooks /
Name
Last Modified
data
an hour ago
lstm_export
a minute ago
01-explore.ipynb
36 minutes ago
02-model.ipynb
a minute ago
03-cloud-training.ip...
an hour ago
cta_ridership.csv
an hour ago
Untitled.ipynb
Terminal 1
01-explore.ipynb
02-model.ipynb
Code
git
Python 3
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.
[72]:
# 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])
Epoch 1/1000
173/173 [==========] - 6s 36ms/step - loss: 0.6863 - val_loss: 0.5092
Epoch 2/1000
173/173 [==========] - 4s 24ms/step - loss: 0.3529 - val_loss: 0.2498
Epoch 3/1000
173/173 [==========] - 4s 22ms/step - loss: 0.2403 - val_loss: 0.2319
Epoch 4/1000
173/173 [==========] - 5s 29ms/step - loss: 0.2293 - val_loss: 0.2329
Epoch 5/1000
173/173 [==========] - 4s 22ms/step - loss: 0.2236 - val_loss: 0.2405
Epoch 6/1000
173/173 [==========] - 4s 22ms/step - loss: 0.2195 - val_loss: 0.2453
Epoch 7/1000
173/173 [==========] - 5s 29ms/step - loss: 0.2134 - val_loss: 0.2244
Epoch 8/1000
173/173 [==========] - 6s 32ms/step - loss: 0.2106 - val_loss: 0.2241
Epoch 9/1000
173/173 [==========] - 6s 37ms/step - loss: 0.2064 - val_loss: 0.2254
Epoch 10/1000
173/173 [==========] - 4s 25ms/step - loss: 0.2054 - val_loss: 0.2296
Epoch 11/1000
173/173 [==========] - 4s 22ms/step - loss: 0.2035 - val_loss: 0.2128
Epoch 12/1000
173/173 [==========] - 4s 22ms/step - loss: 0.2015 - val_loss: 0.2098
Epoch 13/1000

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

/ ··· / ai-for-time-series / notebooks /

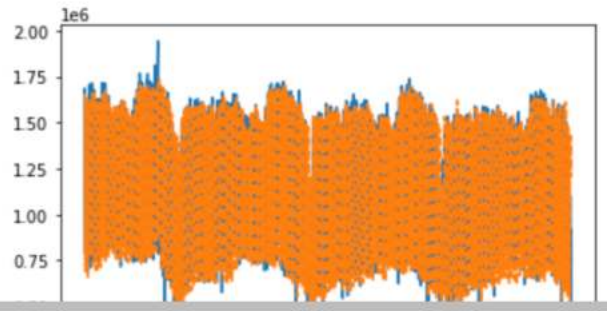
Name	Last Modified
data	an hour ago
lstm_export	2 minutes ago
01-explore.ipynb	36 minutes ago
02-model.ipynb	a minute ago
03-cloud-training.ip...	an hour ago
cta_ridership.csv	an hour ago

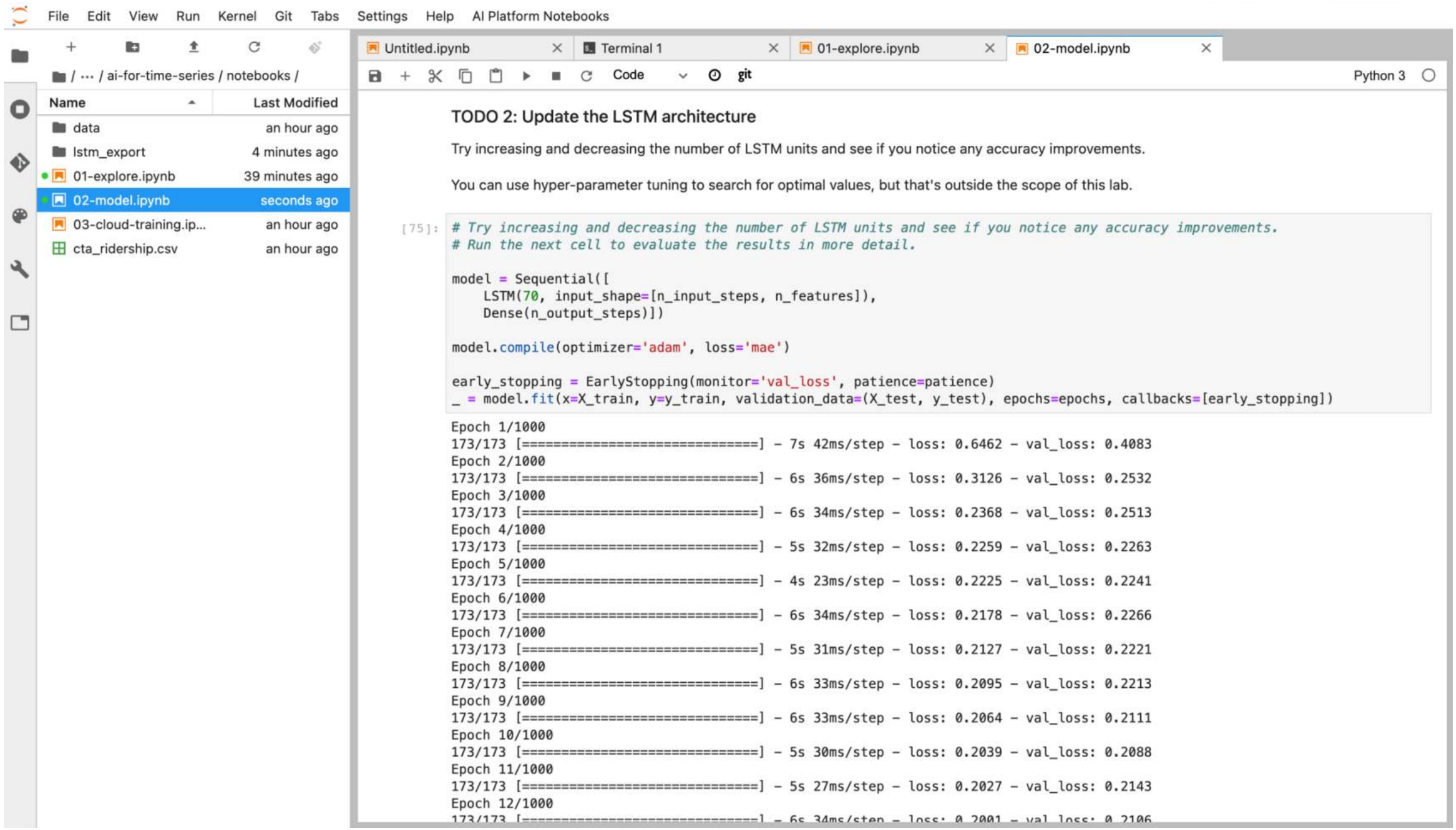
Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

Code git Python 3

```
=== t+(1-7) ===
R^2:  0.804
MAPE: 0.093
MAE:  80759.066

=== t+1 ===
R^2:  0.85
MAPE: 0.078
MAE:  66168.836
```


```
=== t+2 ===
R^2:  0.823
MAPE: 0.088
MAE:  75914.069
```




File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks
/ ··· / ai-for-time-series / notebooks /
Name
Last Modified
data
an hour ago
lstm_export
4 minutes ago
01-explore.ipynb
39 minutes ago
02-model.ipynb
seconds ago
03-cloud-training.ip...
an hour ago
cta_ridership.csv
an hour ago
Untitled.ipynb
Terminal 1
01-explore.ipynb
02-model.ipynb
Code
git
Python 3
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.
[75]:
# 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(70, 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])
Epoch 1/1000
173/173 [==========] - 7s 42ms/step - loss: 0.6462 - val_loss: 0.4083
Epoch 2/1000
173/173 [==========] - 6s 36ms/step - loss: 0.3126 - val_loss: 0.2532
Epoch 3/1000
173/173 [==========] - 6s 34ms/step - loss: 0.2368 - val_loss: 0.2513
Epoch 4/1000
173/173 [==========] - 5s 32ms/step - loss: 0.2259 - val_loss: 0.2263
Epoch 5/1000
173/173 [==========] - 4s 23ms/step - loss: 0.2225 - val_loss: 0.2241
Epoch 6/1000
173/173 [==========] - 6s 34ms/step - loss: 0.2178 - val_loss: 0.2266
Epoch 7/1000
173/173 [==========] - 5s 31ms/step - loss: 0.2127 - val_loss: 0.2221
Epoch 8/1000
173/173 [==========] - 6s 33ms/step - loss: 0.2095 - val_loss: 0.2213
Epoch 9/1000
173/173 [==========] - 6s 33ms/step - loss: 0.2064 - val_loss: 0.2111
Epoch 10/1000
173/173 [==========] - 5s 30ms/step - loss: 0.2039 - val_loss: 0.2088
Epoch 11/1000
173/173 [==========] - 5s 27ms/step - loss: 0.2027 - val_loss: 0.2143
Epoch 12/1000

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
data	an hour ago
lstm_export	seconds ago
01-explore.ipynb	40 minutes ago
02-model.ipynb	seconds ago
03-cloud-training.ip...	an hour ago
cta_ridership.csv	an hour ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

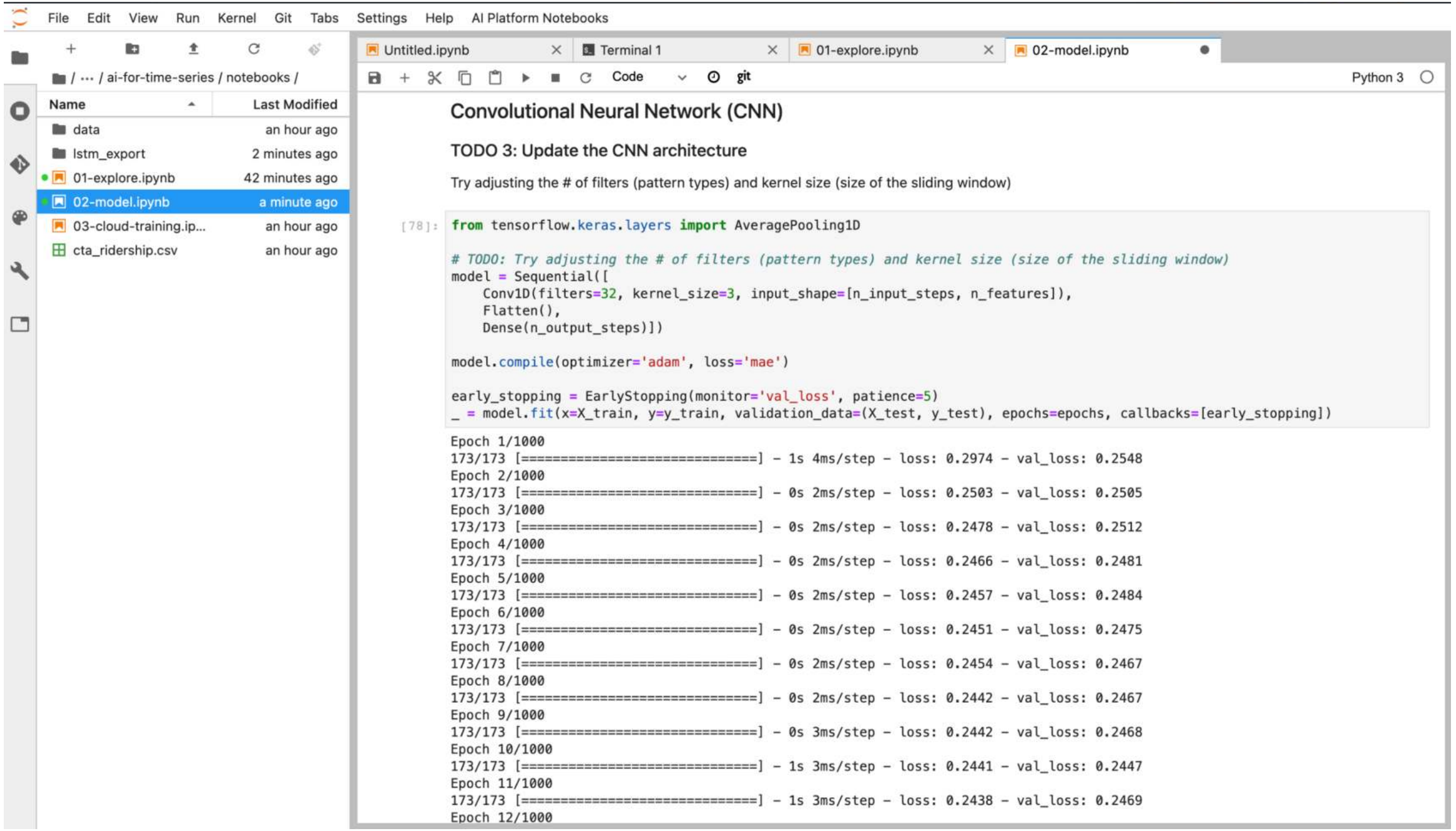
Markdown git Python 3

```
INFO:tensorflow:Assets written to: ./lstm_export/assets
```

[77]:
```
# Predict the results, and then reverse the transformation that scaled all values to a mean of 0 and std. dev. of 1
preds = model.predict(X_test)
y_pred_lstm = inverse_scale(preds)

# Evaluate the overall results and for each time step
evaluate(y_pred_lstm)

# The plot will show the R^2 value (0 lowest -> 1 highest) and the MAE (mean absolute error) for the entire prediction windo
# It will also show individual plots for 1 day out, 2 days out, etc. comparing the actual vs the predicted value.
```

```
=== t+(1-7) ===
R^2:  0.805
MAPE: 0.094
MAE:  81756.44

=== t+1 ===
R^2:  0.856
MAPE: 0.081
MAE:  68995.976
```


```
=== t+2 ===
R^2:  0.815
MAPE: 0.091
MAE:  78130.804
```


File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks
/ ··· / ai-for-time-series / notebooks /
Name
Last Modified
data
an hour ago
lstm_export
2 minutes ago
01-explore.ipynb
42 minutes ago
02-model.ipynb
a minute ago
03-cloud-training.ip...
an hour ago
cta_ridership.csv
an hour ago
Untitled.ipynb
Terminal 1
01-explore.ipynb
02-model.ipynb
Code
git
Python 3
Convolutional Neural Network (CNN)
TODO 3: Update the CNN architecture
Try adjusting the # of filters (pattern types) and kernel size (size of the sliding window)
[78]:
from tensorflow.keras.layers import AveragePooling1D
# TODO: Try adjusting the # of filters (pattern types) and kernel size (size of the sliding window)
model = Sequential([
Conv1D(filters=32, kernel_size=3, input_shape=[n_input_steps, n_features]),
Flatten(),
Dense(n_output_steps)])
model.compile(optimizer='adam', loss='mae')
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
_ = model.fit(x=X_train, y=y_train, validation_data=(X_test, y_test), epochs=epochs, callbacks=[early_stopping])
Epoch 1/1000
173/173 [==========] - 1s 4ms/step - loss: 0.2974 - val_loss: 0.2548
Epoch 2/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2503 - val_loss: 0.2505
Epoch 3/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2478 - val_loss: 0.2512
Epoch 4/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2466 - val_loss: 0.2481
Epoch 5/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2457 - val_loss: 0.2484
Epoch 6/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2451 - val_loss: 0.2475
Epoch 7/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2454 - val_loss: 0.2467
Epoch 8/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2442 - val_loss: 0.2467
Epoch 9/1000
173/173 [==========] - 0s 3ms/step - loss: 0.2442 - val_loss: 0.2468
Epoch 10/1000
173/173 [==========] - 1s 3ms/step - loss: 0.2441 - val_loss: 0.2447
Epoch 11/1000
173/173 [==========] - 1s 3ms/step - loss: 0.2438 - val_loss: 0.2469
Epoch 12/1000

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

/ ··· / ai-for-time-series / notebooks /

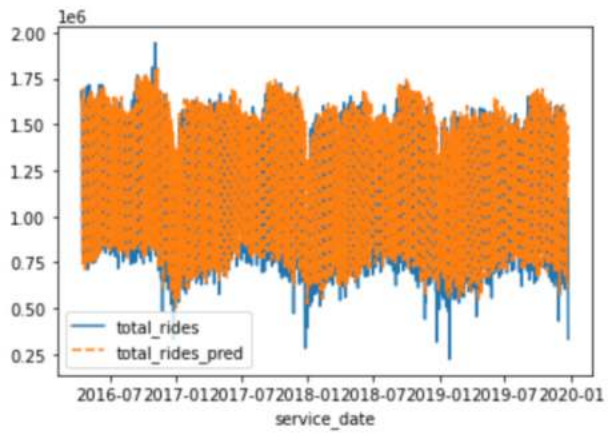
Name	Last Modified
cnn_export	seconds ago
data	an hour ago
lstm_export	3 minutes ago
01-explore.ipynb	43 minutes ago
02-model.ipynb	a minute ago
03-cloud-training.ip...	an hour ago
cta_ridership.csv	an hour ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

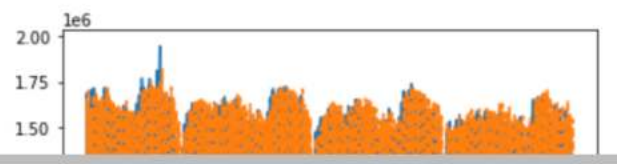
Markdown git Python 3

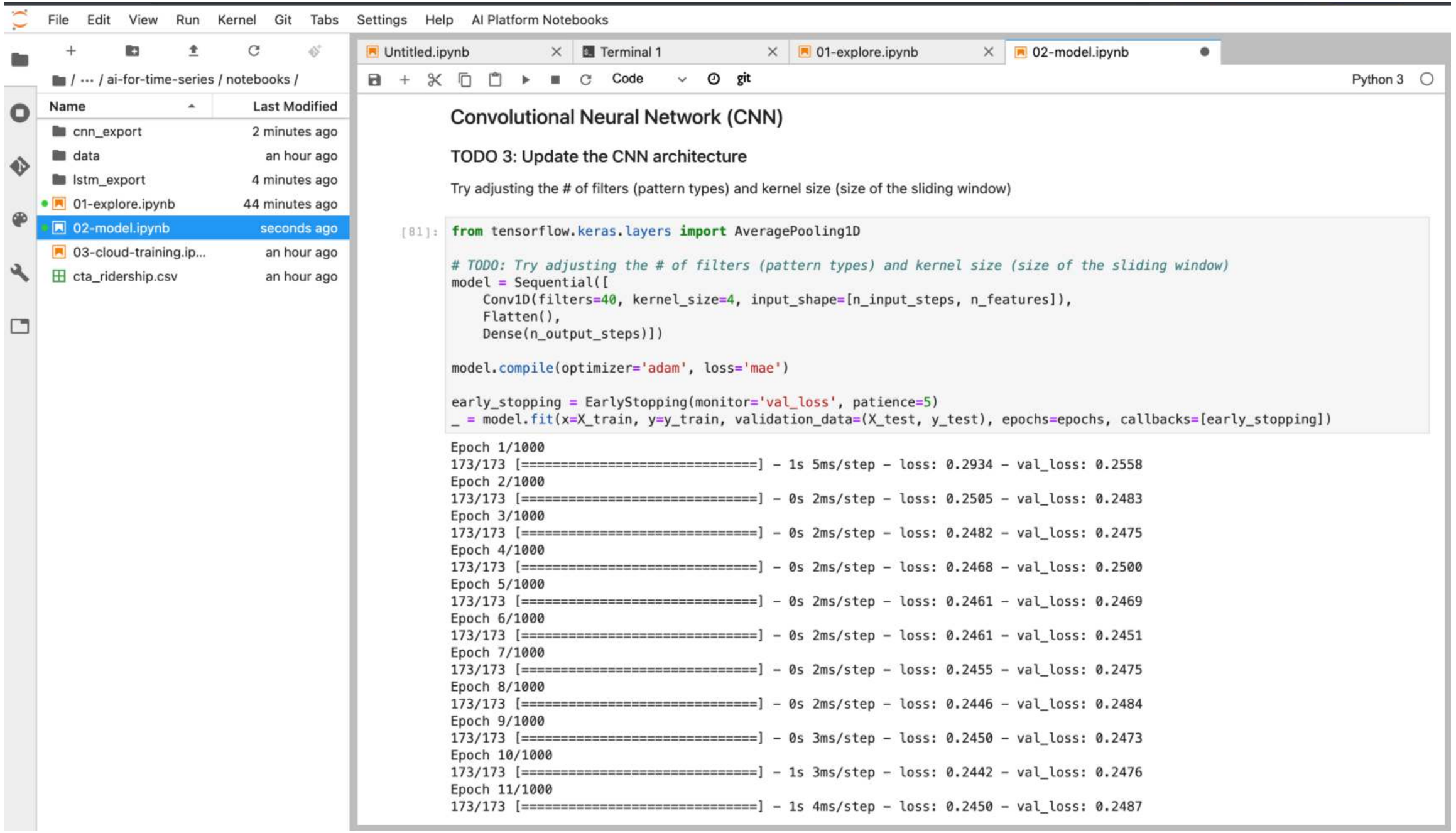
[80]:
```
preds = model.predict(X_test)
y_pred_cnn = inverse_scale(preds)

evaluate(y_pred_cnn)
```

```
=== t+(1-7) ===
R^2:  0.765
MAPE: 0.106
MAE:  97795.296

=== t+1 ===
R^2:  0.802
MAPE: 0.095
MAE:  89388.89
```




```
=== t+2 ===
R^2:  0.766
MAPE: 0.105
MAE:  97860.081
```




File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks
/ ··· / ai-for-time-series / notebooks /
Name
Last Modified
cnn_export 2 minutes ago
data an hour ago
lstm_export 4 minutes ago
01-explore.ipynb 44 minutes ago
02-model.ipynb seconds ago
03-cloud-training.ip... an hour ago
cta_ridership.csv an hour ago
Untitled.ipynb
Terminal 1
01-explore.ipynb
02-model.ipynb
Code
git
Python 3
Convolutional Neural Network (CNN)
TODO 3: Update the CNN architecture
Try adjusting the # of filters (pattern types) and kernel size (size of the sliding window)
[81]:
from tensorflow.keras.layers import AveragePooling1D
# TODO: Try adjusting the # of filters (pattern types) and kernel size (size of the sliding window)
model = Sequential([
Conv1D(filters=40, kernel_size=4, input_shape=[n_input_steps, n_features]),
Flatten(),
Dense(n_output_steps)])
model.compile(optimizer='adam', loss='mae')
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
_ = model.fit(x=X_train, y=y_train, validation_data=(X_test, y_test), epochs=epochs, callbacks=[early_stopping])
Epoch 1/1000
173/173 [==========] - 1s 5ms/step - loss: 0.2934 - val_loss: 0.2558
Epoch 2/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2505 - val_loss: 0.2483
Epoch 3/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2482 - val_loss: 0.2475
Epoch 4/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2468 - val_loss: 0.2500
Epoch 5/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2461 - val_loss: 0.2469
Epoch 6/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2461 - val_loss: 0.2451
Epoch 7/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2455 - val_loss: 0.2475
Epoch 8/1000
173/173 [==========] - 0s 2ms/step - loss: 0.2446 - val_loss: 0.2484
Epoch 9/1000
173/173 [==========] - 0s 3ms/step - loss: 0.2450 - val_loss: 0.2473
Epoch 10/1000
173/173 [==========] - 1s 3ms/step - loss: 0.2442 - val_loss: 0.2476
Epoch 11/1000
173/173 [==========] - 1s 4ms/step - loss: 0.2450 - val_loss: 0.2487

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
cnn_export	seconds ago
data	an hour ago
lstm_export	5 minutes ago
01-explore.ipynb	44 minutes ago
02-model.ipynb	a minute ago
03-cloud-training.ip...	an hour ago
cta_ridership.csv	an hour ago

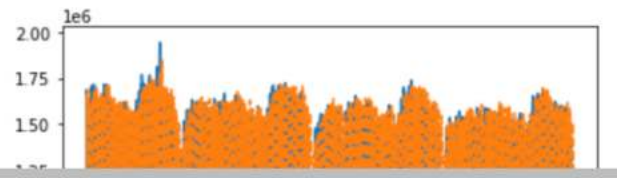
Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

Markdown git Python 3

[83]:
```
preds = model.predict(X_test)
y_pred_cnn = inverse_scale(preds)

evaluate(y_pred_cnn)
```

```
=== t+(1-7) ===
R^2:  0.763
MAPE: 0.105
MAE:  97976.89

=== t+1 ===
R^2:  0.805
MAPE: 0.094
MAE:  89624.479
```


```
=== t+2 ===
R^2:  0.76
MAPE: 0.104
MAE:  97324.473
```



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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
cnn_export	3 minutes ago
data	an hour ago
lstm_export	8 minutes ago
01-explore.ipynb	an hour ago
02-model.ipynb	seconds ago
03-cloud-training.ip...	an hour ago
cta_ridership.csv	an hour ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

Markdown git Python 3

Naïve Models

So-called "naïve models" can be surprisingly hard to beat. These can serve as a useful benchmark for your model's performance.

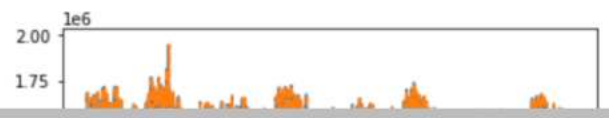
Random Walk

Assume that future value(s) will be the same as the most recent value.

[84]:
```
from statsmodels.tsa.arima.model import ARIMA

hist = df_train[target_col].copy() # Predict based on historical data. Start with the training data
hist.index.freq = pd.infer_freq(hist.index) # To avoid warnings, explicitly specify the dataframe frequency
n_pred = len(df_test) + 1 # Number of predictions: 1 on the training set; and then 1 for each additional
y_pred_rw = np.empty([n_pred,n_output_steps]) # Create an array to hold predictions, with a number of predictions equal to t

for t in range(n_pred):
    mod = ARIMA(hist, order=(0, 1, 0))
    res = mod.fit()
    pred = res.forecast(n_output_steps)
    y_pred_rw[t] = pred.values
    if t < n_pred - 1:
        hist.loc[df_test.iloc[t].name] = df_test[target_col][t] # Append the latest test data row to the history, for fittin
        hist.index.freq = pd.infer_freq(hist.index)
```

[85]:
```
evaluate(y_pred_rw, 0)
```

```
=== t+(1-7) ===
R^2:  -0.834
MAPE: 0.366
MAE:  364578.376

=== t+1 ===
R^2:  -0.19
MAPE: 0.257
MAE:  269441.826
```



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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
cnn_export	3 minutes ago
data	an hour ago
lstm_export	8 minutes ago
01-explore.ipynb	an hour ago
02-model.ipynb	seconds ago
03-cloud-training.ip...	an hour ago
cta_ridership.csv	an hour ago

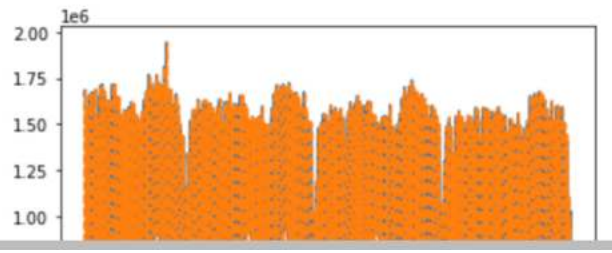
Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

Markdown git Python 3

[85]:
```
evaluate(y_pred_rw, 0)
```

```
=== t+(1-7) ===
R^2:  -0.834
MAPE: 0.366
MAE:  364578.376

=== t+1 ===
R^2:  -0.19
MAPE: 0.257
MAE:  269441.826
```


```
=== t+2 ===
R^2:  -1.383
MAPE: 0.452
MAE:  451553.613
```



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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
cnn_export	10 minutes ago
data	2 hours ago
lstm_export	15 minutes ago
01-explore.ipynb	an hour ago
02-model.ipynb	5 minutes ago
03-cloud-training.ip...	2 hours ago
cta_ridership.csv	an hour ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

Markdown git Python 3

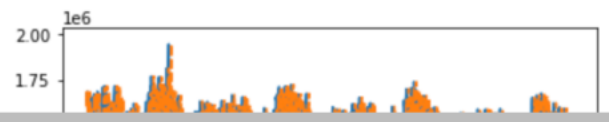
Seasonal Naïve

Similar to random walk, but instead of using the previous value, you'll use the value from the previous seasonal period. For example, if you're predicting July's forecast, you'll use last July's value, rather than June's value.

[87]:
```
# You will use a walk-forward approach, in which a model is fit on all historical data available.
# As you progress through the test set to evaluate the model, you will be creating new models for each row in the test set.
# Each new model will be fit on not only the training data, but on prior test data.

from statsmodels.tsa.statespace.sarimax import SARIMAX

hist = df_train[target_col].copy() # Predict based on historical data. Start with the training data
hist.index.freq = pd.infer_freq(hist.index) # To avoid warnings, explicitly specify the dataframe frequency
n_pred = len(df_test) + 1 # Number of predictions: 1 on the training set; and then 1 for each additional
y_pred_sn = np.empty([n_pred,n_output_steps]) # Create an array to hold predictions, with a number of predictions equal to t

for t in range(n_pred):
    mod = SARIMAX(hist, order=(0, 0, 0), seasonal_order=(0, 1, 0, n_seasons))
    res = mod.fit()
    pred = res.forecast(n_output_steps)
    y_pred_sn[t] = pred.values
    if t < n_pred - 1:
        hist.loc[df_test.iloc[t].name] = df_test[target_col][t] # Append the latest test data row to the history, for fittin
        hist.index.freq = pd.infer_freq(hist.index)
```

[88]:
```
evaluate(y_pred_sn, 0)
```

```
=== t+(1-7) ===
R^2:  0.675
MAPE: 0.11
MAE:  108722.34

=== t+1 ===
R^2:  0.676
MAPE: 0.11
MAE:  108556.529
```



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

/ ··· / ai-for-time-series / notebooks /

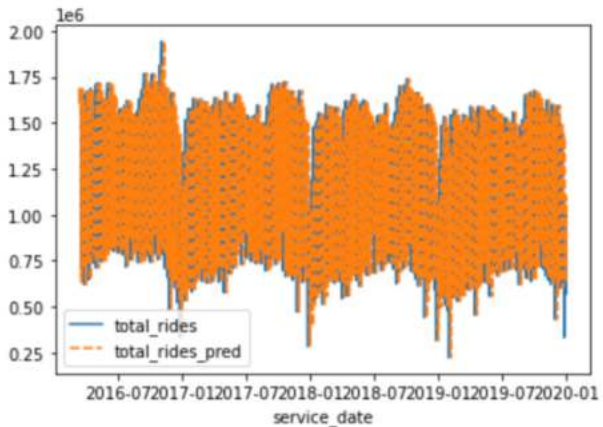
Name	Last Modified
cnn_export	10 minutes ago
data	2 hours ago
lstm_export	15 minutes ago
01-explore.ipynb	an hour ago
02-model.ipynb	5 minutes ago
03-cloud-training.ip...	2 hours ago
cta_ridership.csv	an hour ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

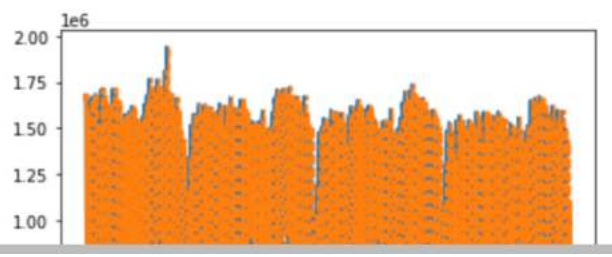
Markdown git Python 3

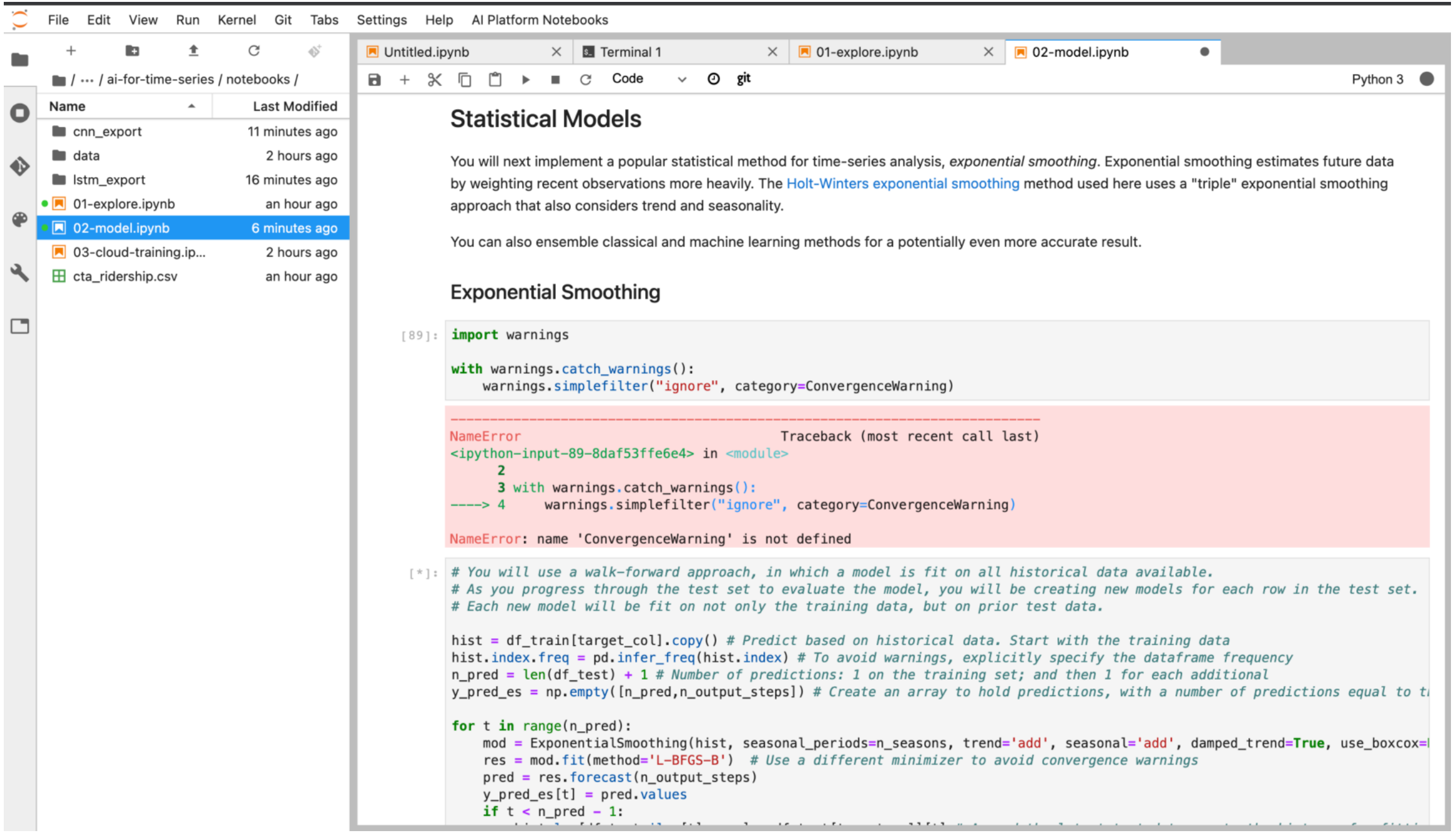
[88]:
```
evaluate(y_pred_sn, 0)
```

```
=== t+(1-7) ===
R^2:  0.675
MAPE: 0.11
MAE:  108722.34

=== t+1 ===
R^2:  0.676
MAPE: 0.11
MAE:  108556.529
```




```
=== t+2 ===
R^2:  0.675
MAPE: 0.11
MAE:  108611.732
```




File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks
/ ··· / ai-for-time-series / notebooks /
Name
Last Modified
cnn_export 11 minutes ago
data 2 hours ago
lstm_export 16 minutes ago
01-explore.ipynb an hour ago
02-model.ipynb 6 minutes ago
03-cloud-training.ip... 2 hours ago
cta_ridership.csv an hour ago
Untitled.ipynb
Terminal 1
01-explore.ipynb
02-model.ipynb
Code
git
Python 3
Statistical Models
You will next implement a popular statistical method for time-series analysis, exponential smoothing. Exponential smoothing estimates future data by weighting recent observations more heavily. The Holt-Winters exponential smoothing method used here uses a "triple" exponential smoothing approach that also considers trend and seasonality.
You can also ensemble classical and machine learning methods for a potentially even more accurate result.
Exponential Smoothing
[89]:
import warnings
with warnings.catch_warnings():
warnings.simplefilter("ignore", category=ConvergenceWarning)
NameError Traceback (most recent call last)
<ipython-input-89-8daf53ffe6e4> in <module>
2
3 with warnings.catch_warnings():
----> 4 warnings.simplefilter("ignore", category=ConvergenceWarning)
NameError: name 'ConvergenceWarning' is not defined
[*]:
# You will use a walk-forward approach, in which a model is fit on all historical data available.
# As you progress through the test set to evaluate the model, you will be creating new models for each row in the test set.
# Each new model will be fit on not only the training data, but on prior test data.
hist = df_train[target_col].copy() # Predict based on historical data. Start with the training data
hist.index.freq = pd.infer_freq(hist.index) # To avoid warnings, explicitly specify the dataframe frequency
n_pred = len(df_test) + 1 # Number of predictions: 1 on the training set; and then 1 for each additional
y_pred_es = np.empty([n_pred,n_output_steps]) # Create an array to hold predictions, with a number of predictions equal to t
for t in range(n_pred):
mod = ExponentialSmoothing(hist, seasonal_periods=n_seasons, trend='add', seasonal='add', damped_trend=True, use_boxcox=
res = mod.fit(method='L-BFGS-B') # Use a different minimizer to avoid convergence warnings
pred = res.forecast(n_output_steps)
y_pred_es[t] = pred.values
if t < n_pred - 1:

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
cnn_export	39 minutes ago
data	2 hours ago
lstm_export	43 minutes ago
01-explore.ipynb	an hour ago
02-model.ipynb	19 minutes ago
03-cloud-training.ip...	2 hours ago
cta_ridership.csv	2 hours ago

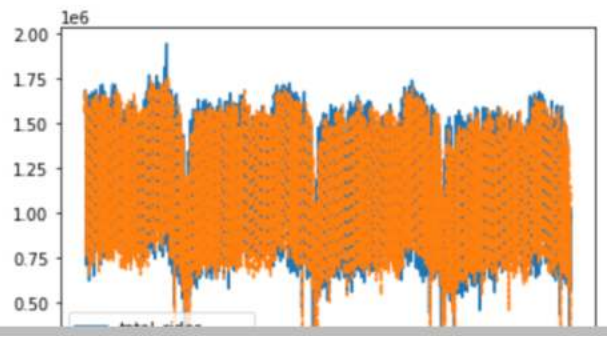
Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

Code git Python 3

[90]:
```
# You will use a walk-forward approach, in which a model is fit on all historical data available.
# As you progress through the test set to evaluate the model, you will be creating new models for each row in the test set.
# Each new model will be fit on not only the training data, but on prior test data.

hist = df_train[target_col].copy() # Predict based on historical data. Start with the training data
hist.index.freq = pd.infer_freq(hist.index) # To avoid warnings, explicitly specify the dataframe frequency
n_pred = len(df_test) + 1 # Number of predictions: 1 on the training set; and then 1 for each additional
y_pred_es = np.empty([n_pred,n_output_steps]) # Create an array to hold predictions, with a number of predictions equal to t

for t in range(n_pred):
    mod = ExponentialSmoothing(hist, seasonal_periods=n_seasons, trend='add', seasonal='add', damped_trend=True, use_boxcox=
    res = mod.fit(method='L-BFGS-B')  # Use a different minimizer to avoid convergence warnings
    pred = res.forecast(n_output_steps)
    y_pred_es[t] = pred.values
    if t < n_pred - 1:
        hist.loc[df_test.iloc[t].name] = df_test[target_col][t] # Append the latest test data row to the history, for fittin
        hist.index.freq = pd.infer_freq(hist.index)
```

[91]:
```
evaluate(y_pred_es, 0)
```

```
=== t+(1-7) ===
R^2:  0.787
MAPE: 0.107
MAE:  99395.043

=== t+1 ===
R^2:  0.834
MAPE: 0.095
MAE:  86633.491
```



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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
cnn_export	39 minutes ago
data	2 hours ago
lstm_export	44 minutes ago
01-explore.ipynb	an hour ago
02-model.ipynb	20 minutes ago
03-cloud-training.ip...	2 hours ago
cta_ridership.csv	2 hours ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

Code git Python 3

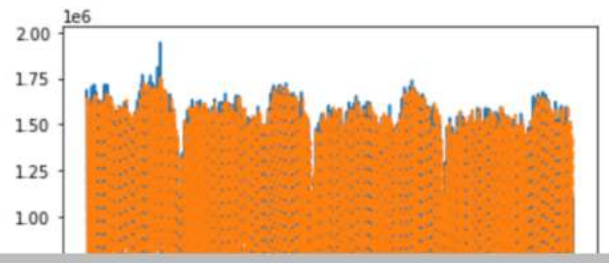
Ensemble ML and Statistical Models

If the performance of the ML and statistical models are similar, ensembling them can be helpful, because their approaches are quite different.

[92]:
```
# Start by adjusting the sizes of the prediction arrays to match.
# Some methods predict the initial timesteps of the test set.
# Others start after the first sequence length.
# So, you will remove the test data that doesn't exist in both sets.

def trunc(df, test_set=df_test, n_input_steps = n_input_steps, n_output_steps = n_output_steps):
    return df[n_input_steps: -n_output_steps]

y_pred_es_trunc = trunc(y_pred_es)
y_true_trunc = trunc(df_test)
```

[93]:
```
models = [y_pred_lstm, y_pred_cnn, y_pred_es_trunc]
weights = [2, 1, 1]

y_pred_ensemble = np.average( np.array(models), axis=0, weights=weights)

evaluate(y_pred_ensemble, 0, y_true_trunc)
```

```
=== t+(1-7) ===
R^2:  0.815
MAPE: 0.089
MAE:  80804.535

=== t+1 ===
R^2:  0.858
MAPE: 0.078
MAE:  70066.568
```



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

/ ··· / ai-for-time-series / notebooks /

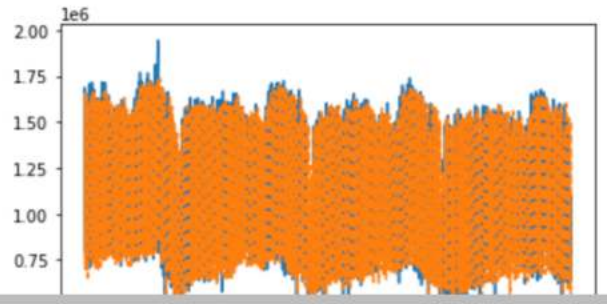
Name	Last Modified
cnn_export	39 minutes ago
data	2 hours ago
lstm_export	44 minutes ago
01-explore.ipynb	an hour ago
02-model.ipynb	20 minutes ago
03-cloud-training.ip...	2 hours ago
cta_ridership.csv	2 hours ago

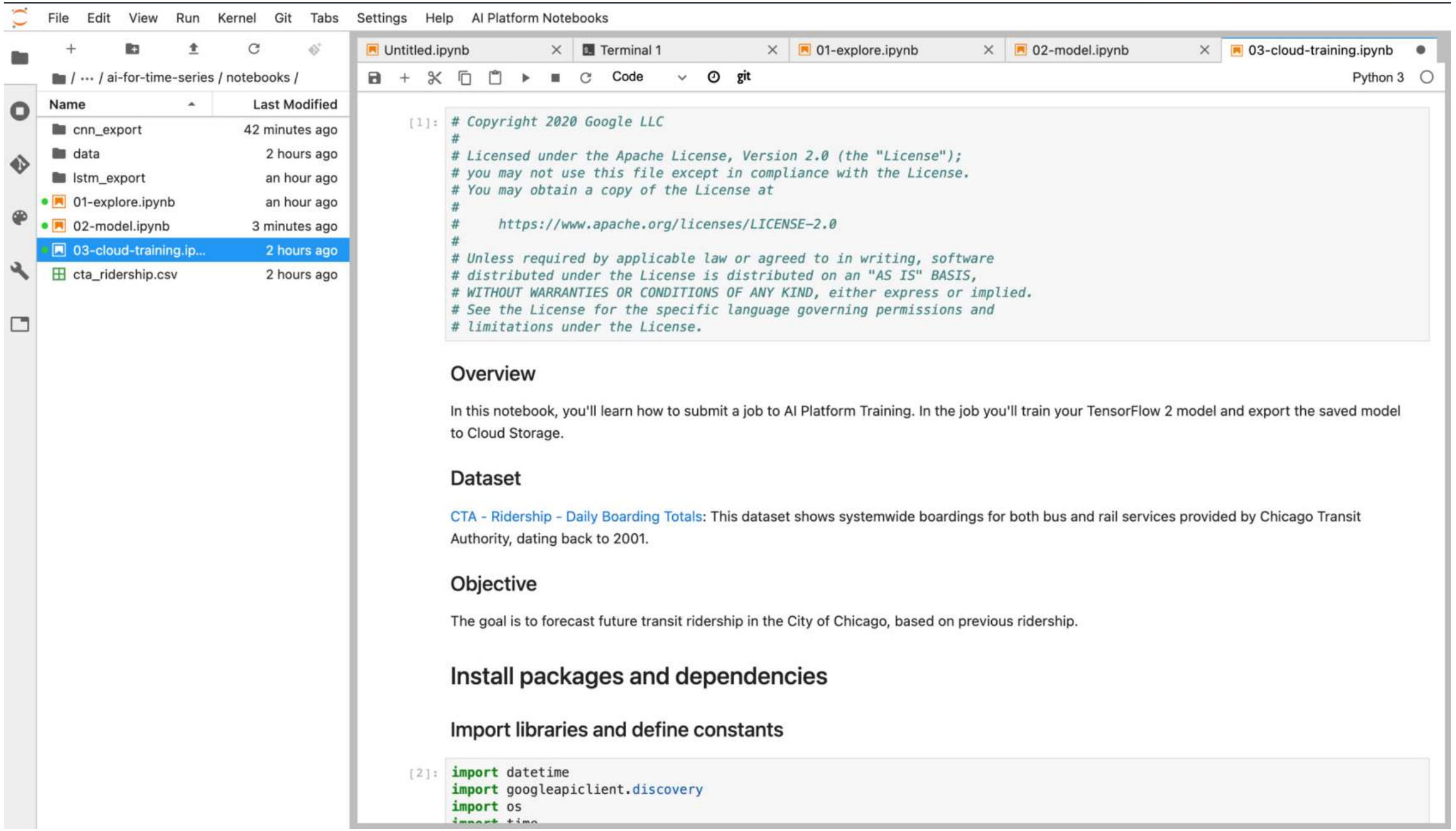
Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

Code git Python 3

```
=== t+(1-7) ===
R^2:  0.815
MAPE: 0.089
MAE:  80804.535

=== t+1 ===
R^2:  0.858
MAPE: 0.078
MAE:  70066.568
```


```
=== t+2 ===
R^2:  0.824
MAPE: 0.087
MAE:  77289.138
```




File Edit View Run Kernel Git Tabs Settings Help AI Platform Notebooks
/ ··· / ai-for-time-series / notebooks /
Name
Last Modified
cnn_export 42 minutes ago
data 2 hours ago
lstm_export an hour ago
01-explore.ipynb an hour ago
02-model.ipynb 3 minutes ago
03-cloud-training.ip... 2 hours ago
cta_ridership.csv 2 hours ago
Untitled.ipynb
Terminal 1
01-explore.ipynb
02-model.ipynb
03-cloud-training.ipynb
Code
git
Python 3
[1]:
Copyright 2020 Google LLC
#
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
#
https://www.apache.org/licenses/LICENSE-2.0
#
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Overview
In this notebook, you'll learn how to submit a job to AI Platform Training. In the job you'll train your TensorFlow 2 model and export the saved model to Cloud Storage.
Dataset
CTA - Ridership - Daily Boarding Totals: This dataset shows systemwide boardings for both bus and rail services provided by Chicago Transit Authority, dating back to 2001.
Objective
The goal is to forecast future transit ridership in the City of Chicago, based on previous ridership.
Install packages and dependencies
Import libraries and define constants
[2]:
import datetime
import googleapiclient.discovery
import os

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
cnn_export	43 minutes ago
data	2 hours ago
lstm_export	an hour ago
01-explore.ipynb	an hour ago
02-model.ipynb	3 minutes ago
03-cloud-training.ip...	seconds ago
cta_ridership.csv	2 hours ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb | 03-cloud-training.ipynb

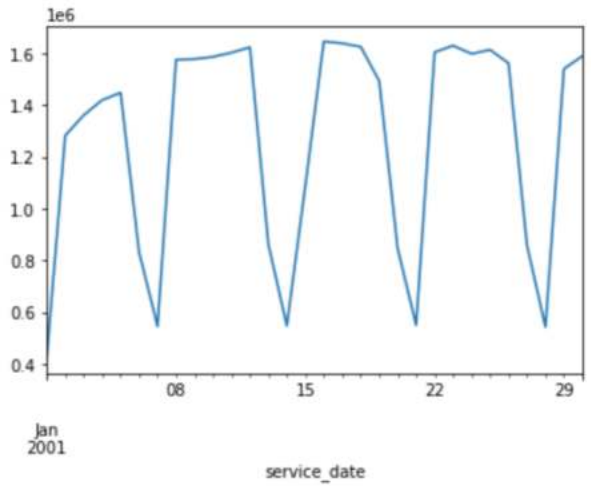
Code git Python 3

Load and preview the data

Pre-processing on the original dataset has been done for you and made available on Cloud Storage.

[9]:
```
processed_file = 'cta_ridership.csv' # Which file to save the results to

if os.path.exists(processed_file):
    input_file = processed_file # File created in previous lab
else:
    input_file = f'data/{processed_file}'
```

[10]:
```
df = pd.read_csv(input_file, index_col=ts_col, parse_dates=True)

# Plot 30 days of ridership
_ = df[target_col][:30].plot()
```



[ ]:
```
# Define some characteristics of the data that will be used later
n_features = len(df.columns)

# Index of target column. Used later when creating dataframes.
target_col_num = df.columns.get_loc(target_col)
```

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
cnn_export	43 minutes ago
data	2 hours ago
lstm_export	an hour ago
01-explore.ipynb	an hour ago
02-model.ipynb	4 minutes ago
03-cloud-training.ip...	seconds ago
cta_ridership.csv	2 hours ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb | 03-cloud-training.ipynb

Markdown git Python 3

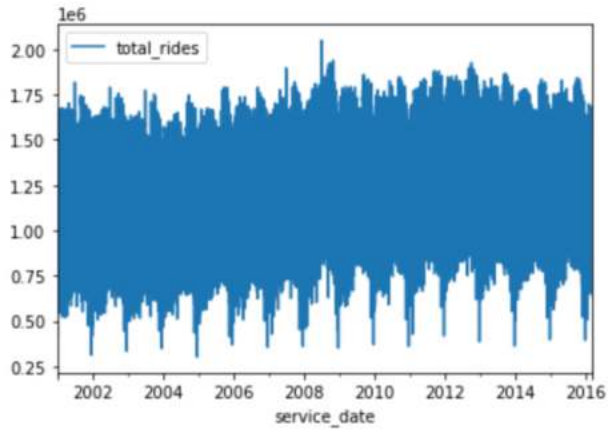
Process data

[12]:
```
# Split data

size = int(len(df) * train_split)
df_train, df_test = df[0:size].copy(deep=True), df[size:len(df)].copy(deep=True)

df_train.head()
```

[12]:

service_date	total_rides
2001-01-01	423647
2001-01-02	1282779
2001-01-03	1361355
2001-01-04	1420032
2001-01-05	1448343

[13]:
```
_ = df_train.plot()
```



Scale values

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

/ ··· / ai-for-time-series / notebooks /

Name	Last Modified
cnn_export	an hour ago
data	an hour ago
lstm_export	an hour ago
01-explore.ipynb	an hour ago
02-model.ipynb	seconds ago
03-cloud-training.ipynb	an hour ago

Untitled.ipynb | Terminal 1 | 01-explore.ipynb | 02-model.ipynb

Code git Python 3

Statistical Models

You will next implement a popular statistical method for time-series analysis, *exponential smoothing*. Exponential smoothing estimates future data by weighting recent observations more heavily. The Holt-Winters exponential smoothing method used here uses a "triple" exponential smoothing approach that also considers trend and seasonality.

You can also ensemble classical and machine learning methods for a potentially even more accurate result.

Exponential Smoothing

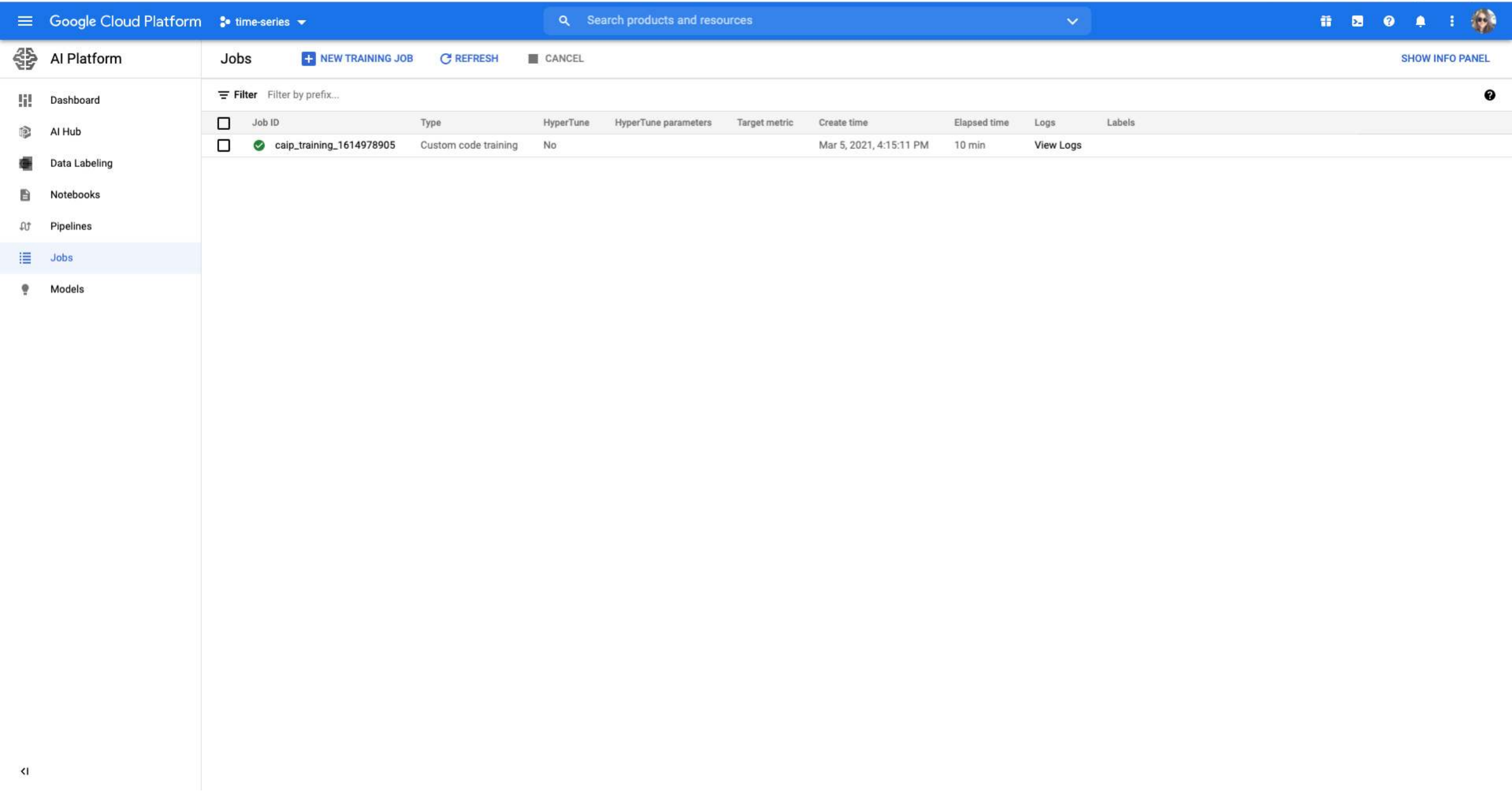
[32]:
```
import warnings
from sklearn.exceptions import ConvergenceWarning

with warnings.catch_warnings():
    warnings.simplefilter("ignore", category=ConvergenceWarning)
```

[33]:
```
# You will use a walk-forward approach, in which a model is fit on all historical data available.
# As you progress through the test set to evaluate the model, you will be creating new models for each row in the test set.
# Each new model will be fit on not only the training data, but on prior test data.

hist = df_train[target_col].copy() # Predict based on historical data. Start with the training data
hist.index.freq = pd.infer_freq(hist.index) # To avoid warnings, explicitly specify the dataframe frequency
n_pred = len(df_test) + 1 # Number of predictions: 1 on the training set; and then 1 for each additional
y_pred_es = np.empty([n_pred,n_output_steps]) # Create an array to hold predictions, with a number of predictions equal to the test set size, each containing the # of tim

for t in range(n_pred):
    mod = ExponentialSmoothing(hist, seasonal_periods=n_seasons, trend='add', seasonal='add', damped_trend=True, use_boxcox=False, initialization_method='heuristic')
    res = mod.fit(method='L-BFGS-B')  # Use a different minimizer to avoid convergence warnings
    pred = res.forecast(n_output_steps)
    y_pred_es[t] = pred.values
    if t < n_pred - 1:
        hist.loc[df_test.iloc[t].name] = df_test[target_col][t] # Append the latest test data row to the history, for fitting the next model
        hist.index.freq = pd.infer_freq(hist.index)
```

[34]:
```
evaluate(y_pred_es, 0)
```

```
=== t+(1-7) ===
R^2:  0.787
MAPE: 0.107
MAE:  99395.043

=== t+1 ===
R^2:  0.834
MAPE: 0.095
MAE:  86633.491
```

1 3 tensorflow-2-1-20210305-143944 | time-series-306705 Git: refreshing... Python 3 | Idle Mode: Command Ln 1, Col 1 02-model.ipynb


Google Cloud Platform
time-series
Search products and resources
AI Platform
Jobs
NEW TRAINING JOB
REFRESH
CANCEL
SHOW INFO PANEL
Dashboard
AI Hub
Data Labeling
Notebooks
Pipelines
Jobs
Models
Filter
Filter by prefix...
Job ID
Type
HyperTune
HyperTune parameters
Target metric
Create time
Elapsed time
Logs
Labels
caip_training_1614978905
Custom code training
No
Mar 5, 2021, 4:15:11 PM
10 min
View Logs

Job Details DEPLOY MODEL DOWNLOAD MODEL

caip_training_1614978905

Succeeded (10 min)

Creation time	Mar 5, 2021, 4:15:11 PM
Start time	Mar 5, 2021, 4:24:10 PM
End time	Mar 5, 2021, 4:25:11 PM
Logs	View Logs
TensorBoard	TensorBoard is available from this page only for models trained with built-in TensorFlow algorithms
Consumed ML units	0.01
Training input	SHOW JSON
Training output	SHOW JSON
Model location	gs://time-series-306705_2902/

CPU GPU NETWORK

COLLAPSE ALL EXPAND ALL

Master CPU utilization

Percent

Master memory utilization

Percent

iOS APP

Create a project (Step 1 of 3)

Let's start with a name for your project

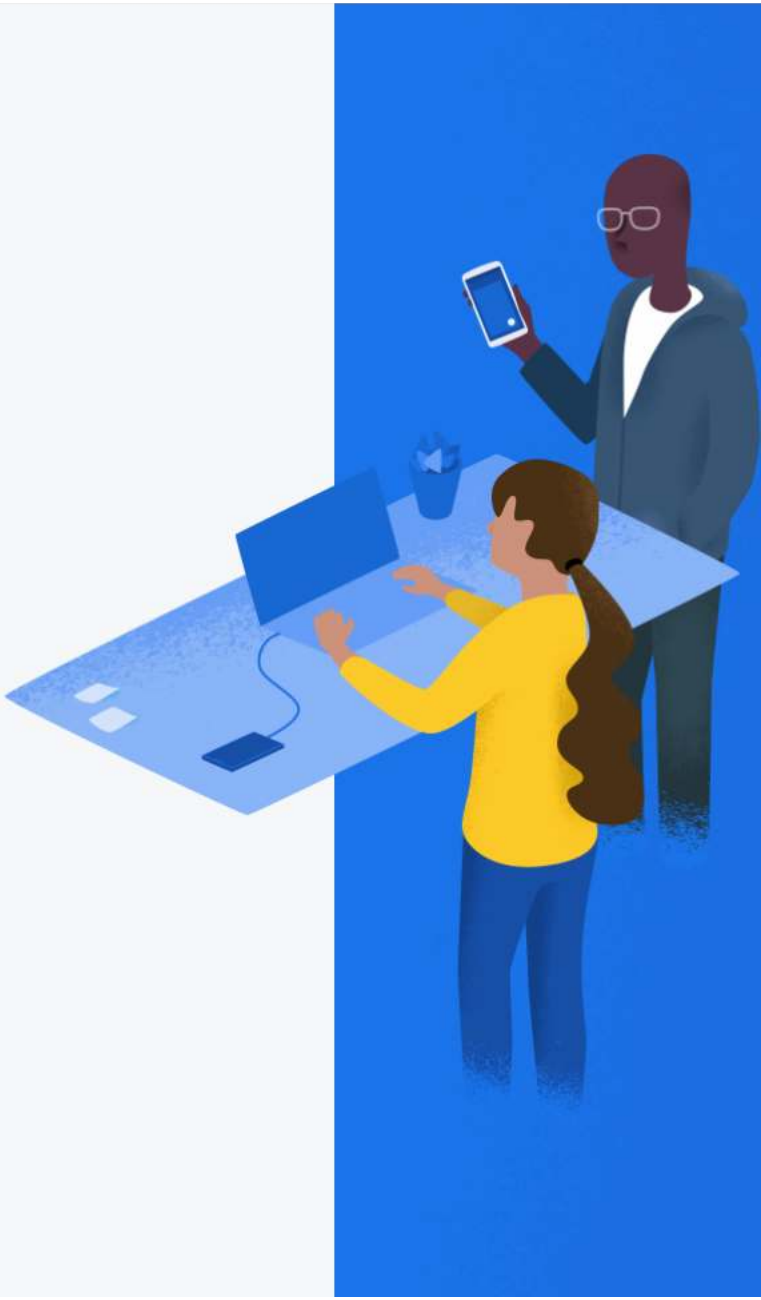
Project name

ML Kit Codelab

ml-kit-codelab-sheema-2902

sjsu.edu

Continue



Create a project (Step 2 of 3)

Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

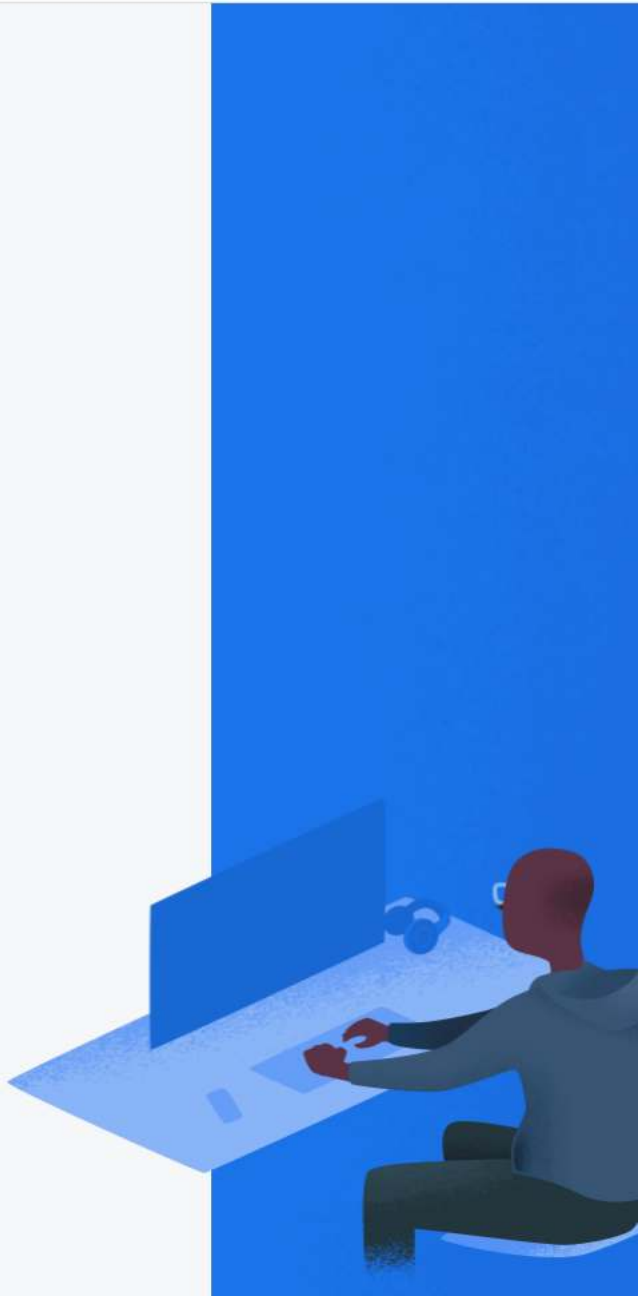
Google Analytics enables:

- A/B testing
- User segmentation & targeting across Firebase products
- Predicting user behavior
- Crash-free users
- Event-based Cloud Functions triggers
- Free unlimited reporting

Enable Google Analytics for this project
Recommended

Previous

Continue



Create a project (Step 3 of 3)

Configure Google Analytics

Choose or create a Google Analytics account

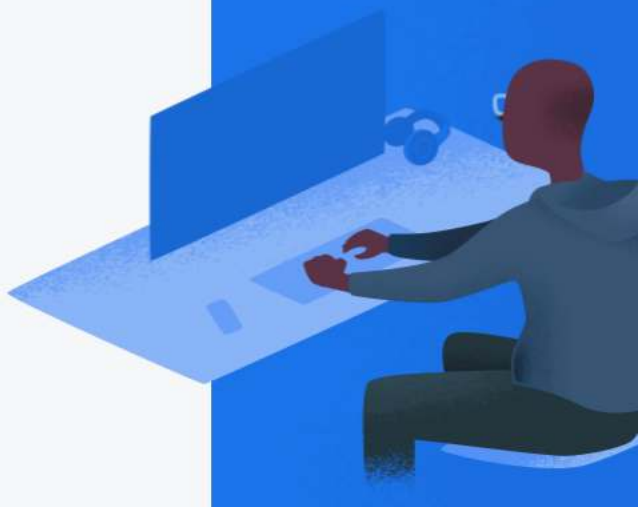
Default Account for Firebase

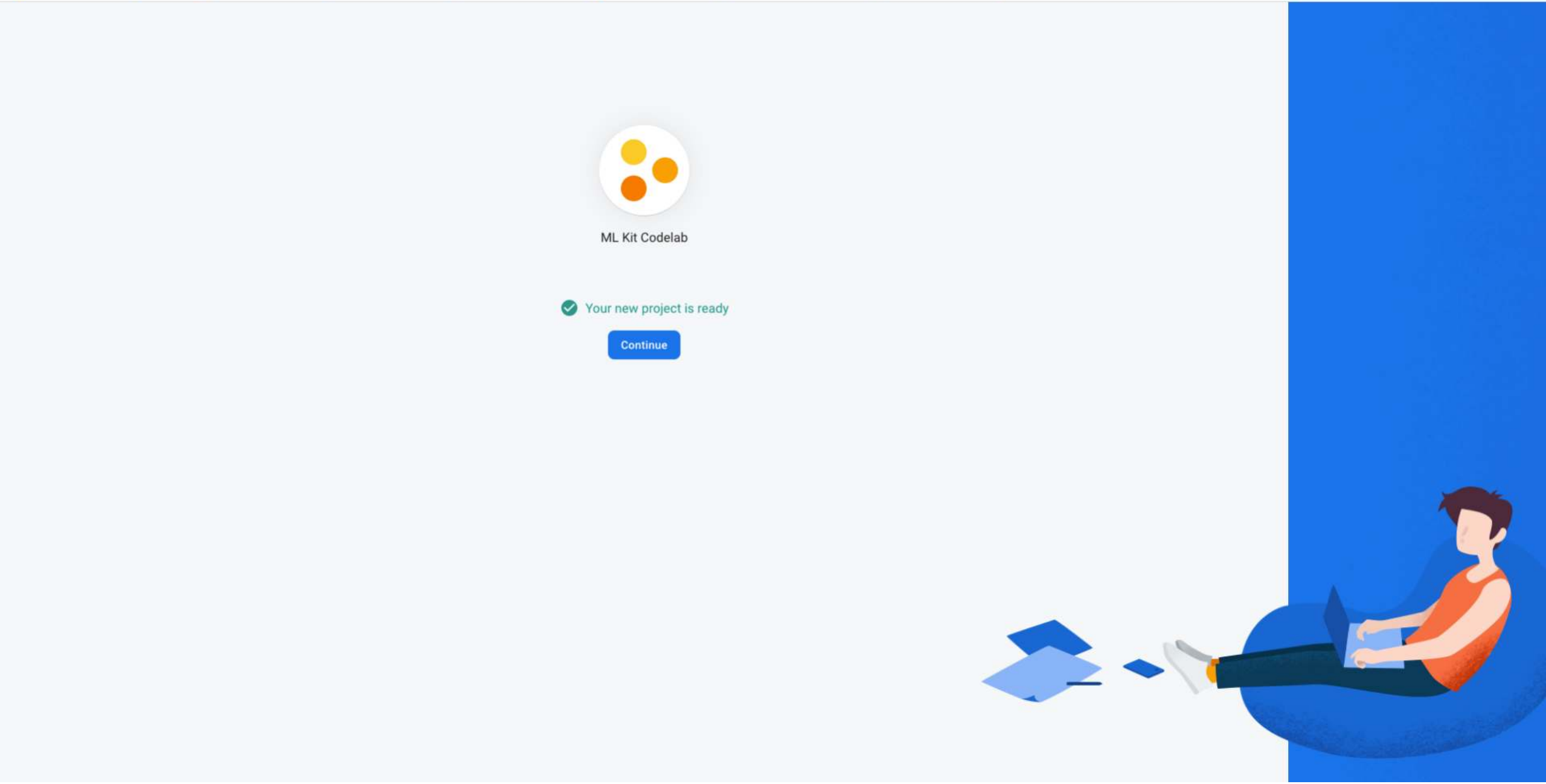
Automatically create a new property in this account

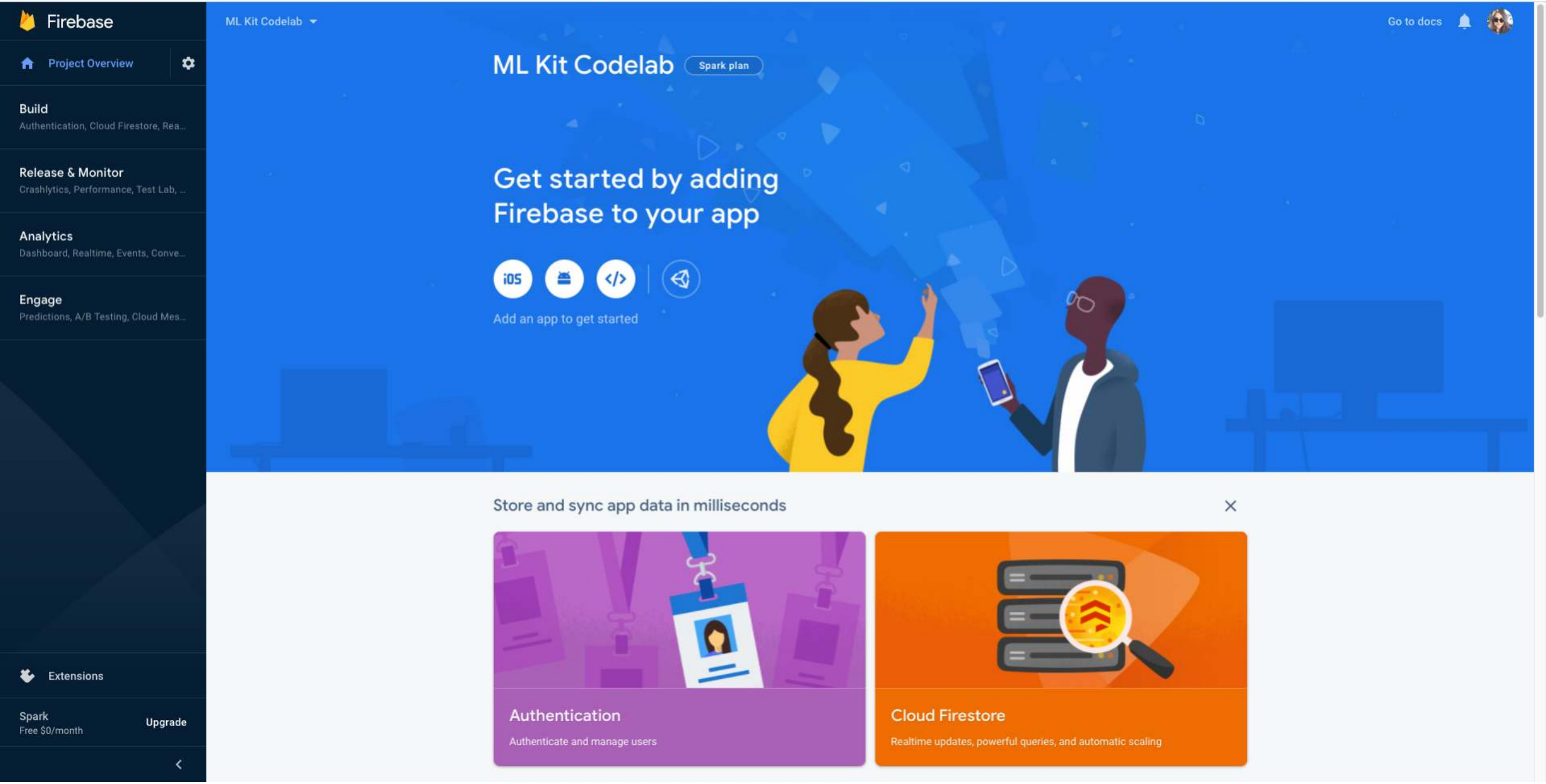
Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. Learn more.

Previous

Create project




ML Kit Codelab
Your new project is ready
Continue


Firebase
ML Kit Codelab
Project Overview
Build
Authentication, Cloud Firestore, Rea...
Release & Monitor
Crashlytics, Performance, Test Lab, ...
Analytics
Dashboard, Realtime, Events, Conve...
Engage
Predictions, A/B Testing, Cloud Mes...
Extensions
Spark
Free $0/month
Upgrade
ML Kit Codelab
Spark plan
Go to docs
Get started by adding
Firebase to your app
iOS
Add an app to get started
Store and sync app data in milliseconds
Authentication
Authenticate and manage users
Cloud Firestore
Realtime updates, powerful queries, and automatic scaling

Add Firebase to your iOS app

1. **Register app**

iOS bundle ID

com.google.firebase.codelab.mlkit.automl

App nickname (optional)

Sheema iOS App

App Store ID (optional)

123456789

Register app

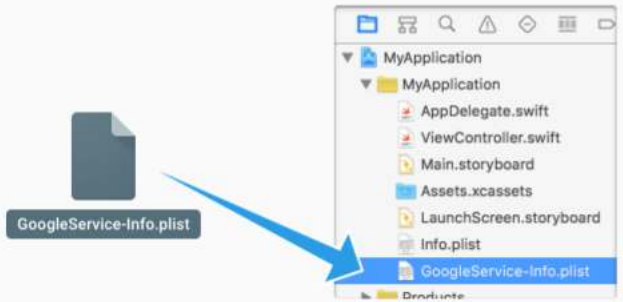
2. Download config file
3. Add Firebase SDK
4. Add initialization code
5. Next steps

Add Firebase to your iOS app

1. Register app
   iOS bundle ID: com.google.firebase.codelab.mlkit.automl, App nickname: Sheema iOS App
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
5. Next steps

Add Firebase to your iOS app

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

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:

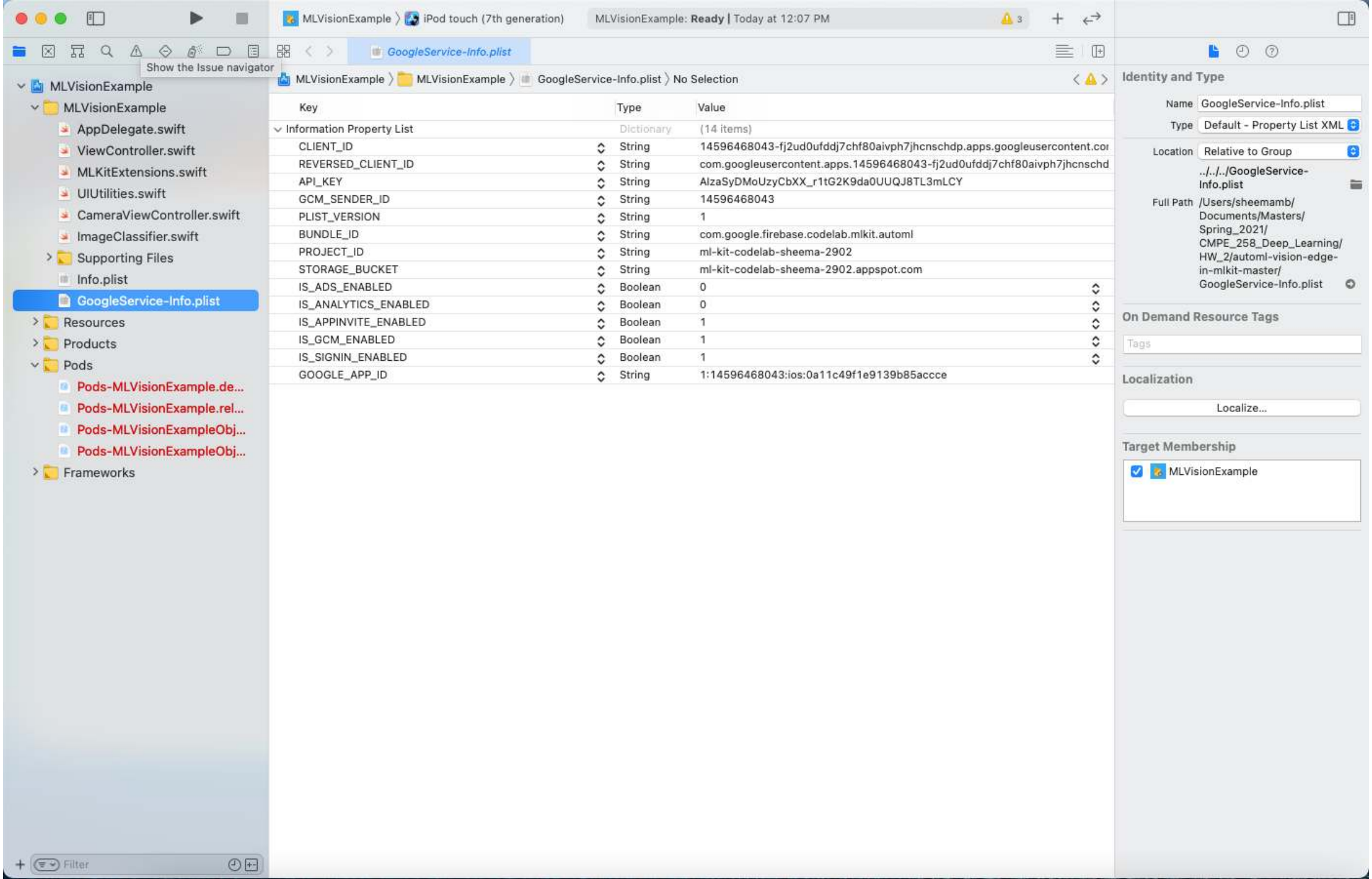
```
$ pod install
```

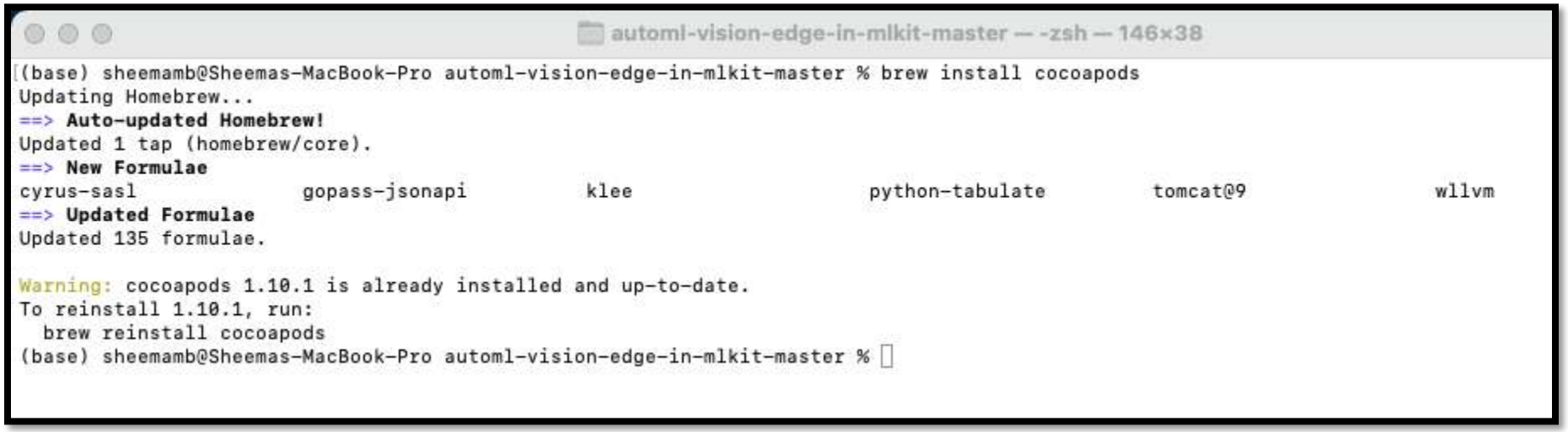
This creates an **.xcworkspace** file for your app. Use this file for all future development on your application.

Previous Next

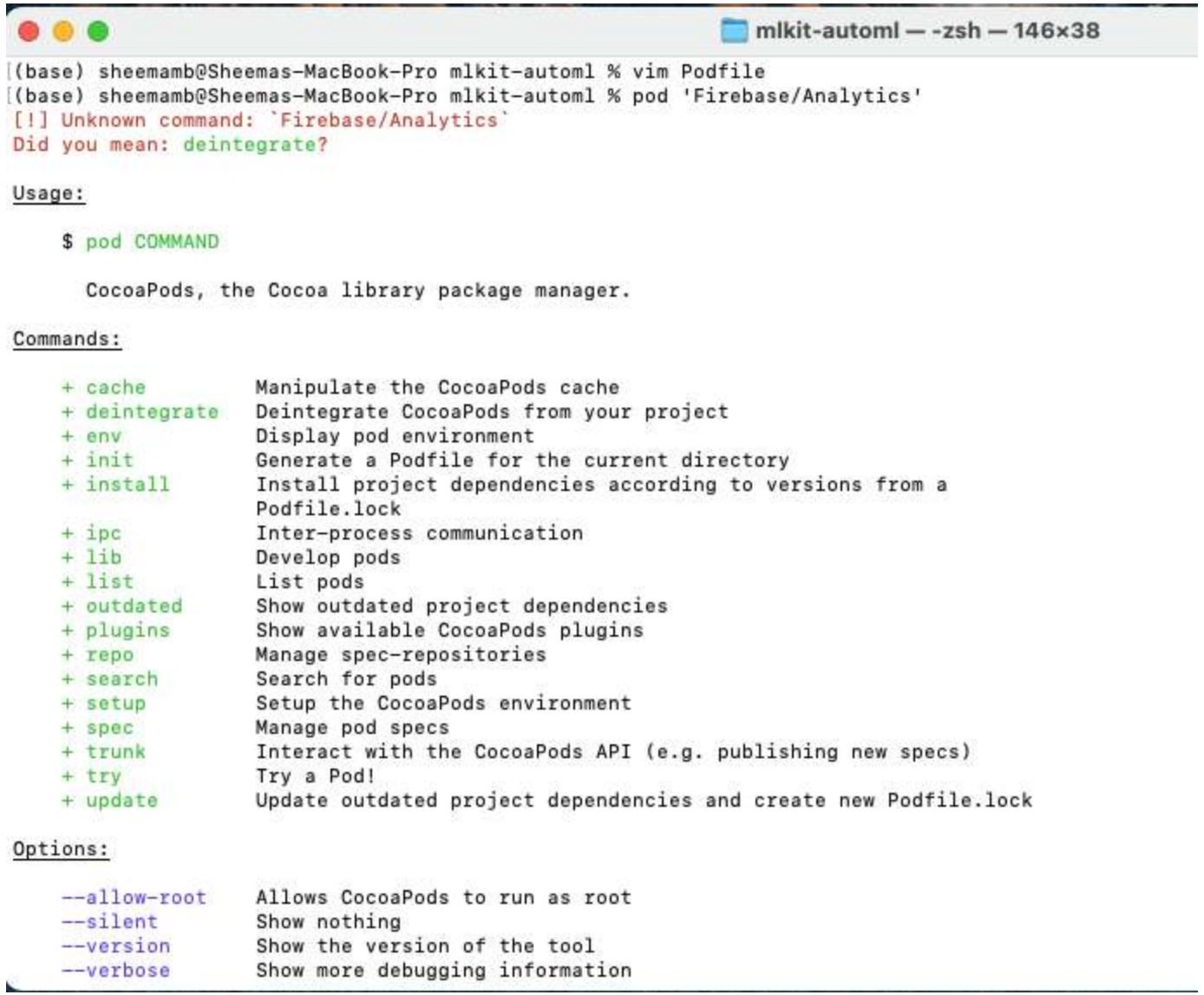
4 Add initialization code

5 Next steps


MLVisionExample 〉 iPod touch (7th generation)
MLVisionExample: Ready | Today at 12:07 PM
Show the Issue navigator
GoogleService-Info.plist
MLVisionExample 〉 MLVisionExample 〉 GoogleService-Info.plist 〉 No Selection
MLVisionExample
MLVisionExample
AppDelegate.swift
ViewController.swift
MLKitExtensions.swift
UIUtilities.swift
CameraViewController.swift
ImageClassifier.swift
Supporting Files
Info.plist
GoogleService-Info.plist
Resources
Products
Pods
Pods-MLVisionExample.de...
Pods-MLVisionExample.rel...
Pods-MLVisionExampleObj...
Pods-MLVisionExampleObj...
Frameworks
Filter
Key
Type
Value
Information Property List
Dictionary
(14 items)
CLIENT_ID
String
14596468043-fj2ud0ufddj7chf80aivph7jhcnschdp.apps.googleusercontent.cor
REVERSED_CLIENT_ID
String
com.googleusercontent.apps.14596468043-fj2ud0ufddj7chf80aivph7jhcnschd
API_KEY
String
AIzaSyDMoUzyCbXX_r1tG2K9da0UUQJ8TL3mLCY
GCM_SENDER_ID
String
14596468043
PLIST_VERSION
String
1
BUNDLE_ID
String
com.google.firebase.codelab.mlkit.automl
PROJECT_ID
String
ml-kit-codelab-sheema-2902
STORAGE_BUCKET
String
ml-kit-codelab-sheema-2902.appspot.com
IS_ADS_ENABLED
Boolean
0
IS_ANALYTICS_ENABLED
Boolean
0
IS_APPINVITE_ENABLED
Boolean
1
IS_GCM_ENABLED
Boolean
1
IS_SIGNIN_ENABLED
Boolean
1
GOOGLE_APP_ID
String
1:14596468043:ios:0a11c49f1e9139b85accce
Identity and Type
Name
GoogleService-Info.plist
Type
Default - Property List XML
Location
Relative to Group
../../../GoogleService-Info.plist
Full Path
/Users/sheemamb/Documents/Masters/Spring_2021/CMPE_258_Deep_Learning/HW_2/automl-vision-edge-in-mlkit-master/GoogleService-Info.plist
On Demand Resource Tags
Tags
Localization
Localize...
Target Membership
MLVisionExample


automl-vision-edge-in-mlkit-master — -zsh — 146×38
(base) sheemamb@Sheemas-MacBook-Pro automl-vision-edge-in-mlkit-master % brew install cocoapods
Updating Homebrew...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
cyrus-sasl gopass-jsonapi klee python-tabulate tomcat@9 wllvm
==> Updated Formulae
Updated 135 formulae.
Warning: cocoapods 1.10.1 is already installed and up-to-date.
To reinstall 1.10.1, run:
brew reinstall cocoapods
(base) sheemamb@Sheemas-MacBook-Pro automl-vision-edge-in-mlkit-master %


mlkit-automl — vim Podfile — 146×38
source 'https://github.com/CocoaPods/Specs.git'
platform :ios, '9.0'
use_frameworks!
target 'MLVisionExample' do
pod 'FirebaseMLVision', '0.16.0'
pod 'FirebaseMLCommon', '0.16.0'
pod 'FirebaseMLVisionAutoML', '0.16.0'
pod 'Firebase/Analytics'
end


mlkit-automl — -zsh — 146×38
(base) sheemamb@Sheemas-MacBook-Pro mlkit-automl % vim Podfile
(base) sheemamb@Sheemas-MacBook-Pro mlkit-automl % pod 'Firebase/Analytics'
[!] Unknown command: `Firebase/Analytics`
Did you mean: deintegrate?
Usage:
$ pod COMMAND
CocoaPods, the Cocoa library package manager.
Commands:
+ cache Manipulate the CocoaPods cache
+ deintegrate Deintegrate CocoaPods from your project
+ env Display pod environment
+ init Generate a Podfile for the current directory
+ install Install project dependencies according to versions from a Podfile.lock
+ ipc Inter-process communication
+ lib Develop pods
+ list List pods
+ outdated Show outdated project dependencies
+ plugins Show available CocoaPods plugins
+ repo Manage spec-repositories
+ search Search for pods
+ setup Setup the CocoaPods environment
+ spec Manage pod specs
+ trunk Interact with the CocoaPods API (e.g. publishing new specs)
+ try Try a Pod!
+ update Update outdated project dependencies and create new Podfile.lock
Options:
--allow-root Allows CocoaPods to run as root
--silent Show nothing
--version Show the version of the tool
--verbose Show more debugging information

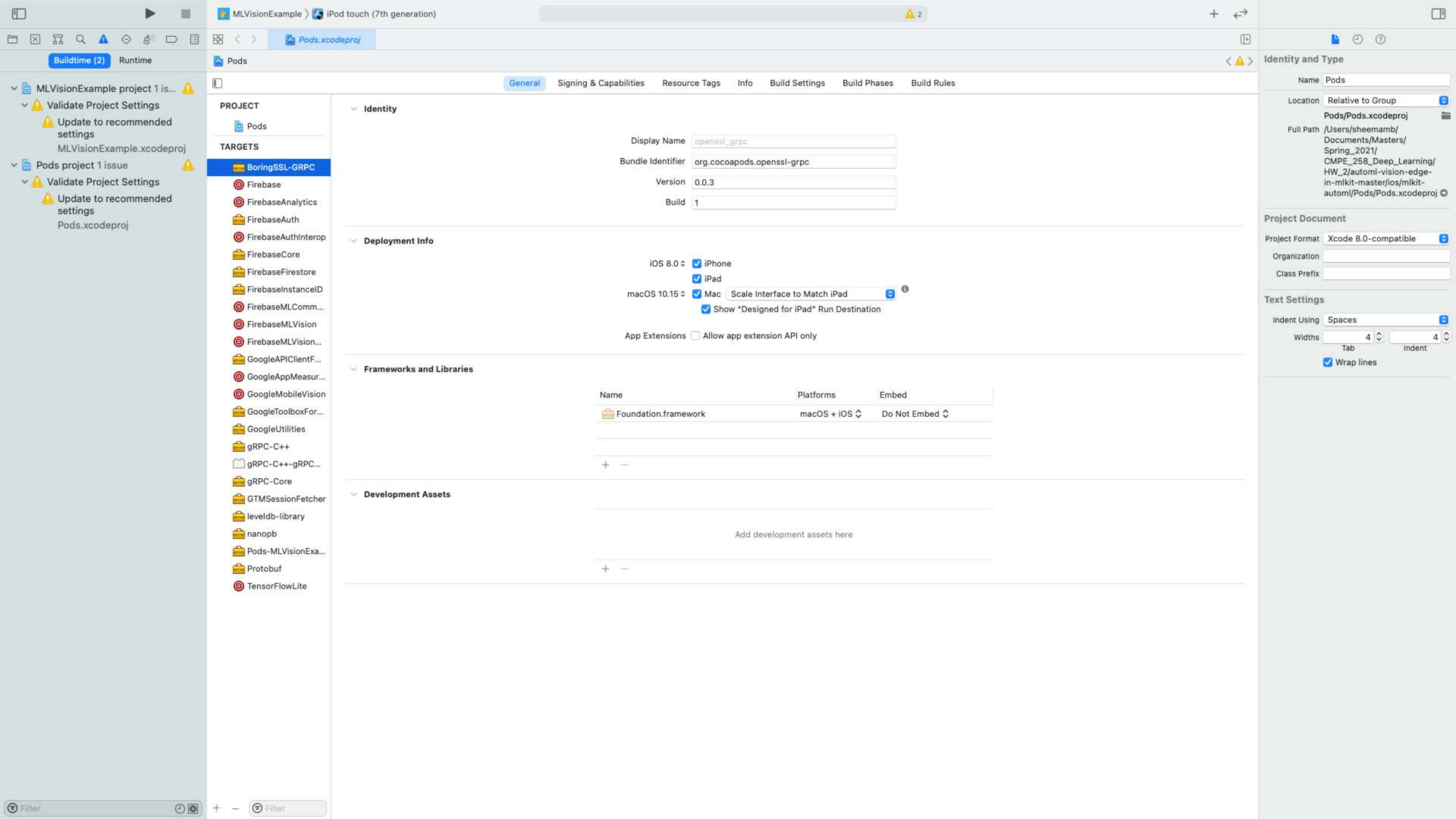

mlkit-automl — -zsh — 146×38
--allow-root Allows CocoaPods to run as root
--silent Show nothing
--version Show the version of the tool
--verbose Show more debugging information
--no-ansi Show output without ANSI codes
--help Show help banner of specified command
(base) sheemamb@Sheemas-MacBook-Pro mlkit-automl % pod install
Analyzing dependencies
Downloading dependencies
Installing Firebase (6.0.0)
Installing FirebaseAnalytics (6.0.0)
Installing FirebaseCore (6.0.0)
Installing FirebaseInstanceID (4.0.0)
Installing FirebaseMLCommon (0.16.0)
Installing FirebaseMLVision (0.16.0)
Installing FirebaseMLVisionAutoML (0.16.0)
Installing GTMSessionFetcher (1.2.1)
Installing GoogleAPIClientForREST (1.3.8)
Installing GoogleAppMeasurement (6.0.0)
Installing GoogleMobileVision (1.6.0)
Installing GoogleToolboxForMac (2.2.0)
Installing GoogleUtilities (6.0.0)
Installing Protobuf (3.7.0)
Installing TensorFlowLite (1.13.1)
Installing nanopb (0.3.9011)
Generating Pods project
Integrating client project
[!] Please close any current Xcode sessions and use `MLVisionExample.xcworkspace` for this project from now on.
Pod installation complete! There are 4 dependencies from the Podfile and 16 total pods installed.
[!] FirebaseMLCommon has been deprecated
[!] FirebaseMLVision has been deprecated
[!] FirebaseMLVisionAutoML has been deprecated in favor of MLKitImageLabelingAutoML
(base) sheemamb@Sheemas-MacBook-Pro mlkit-automl %

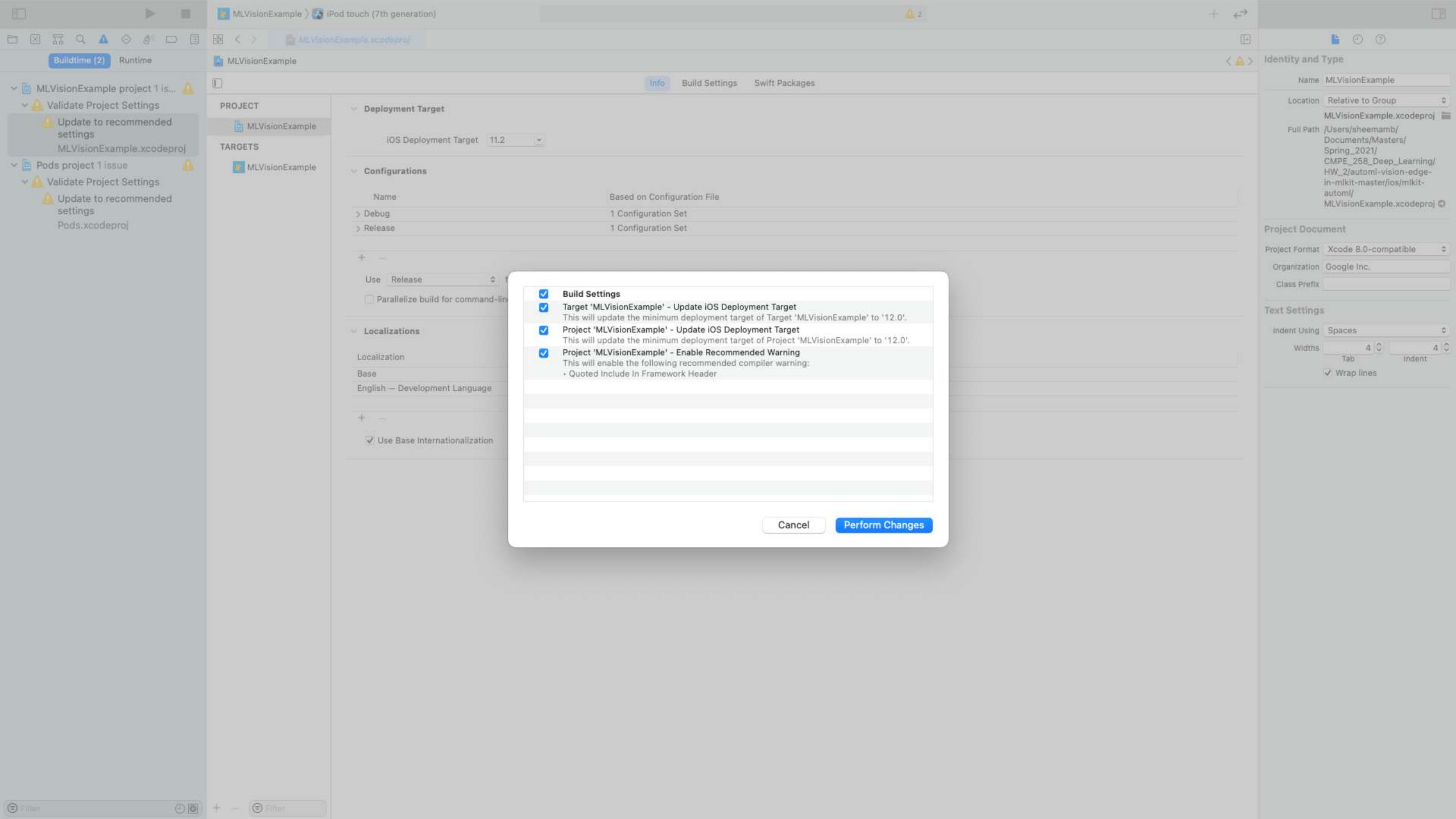
mlkit-automl — vim Podfile — 146×38

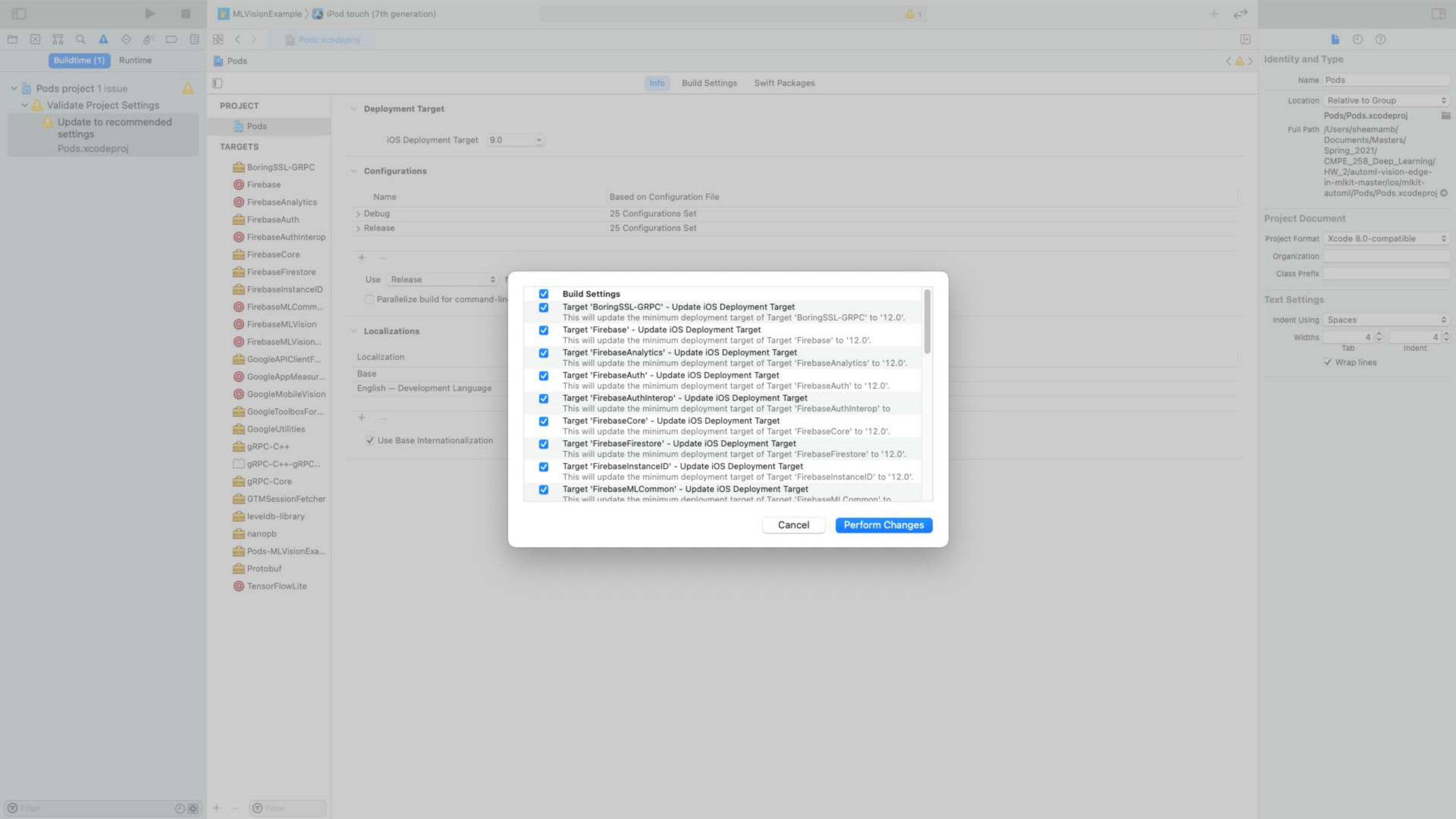
```
source 'https://github.com/CocoaPods/Specs.git'

platform :ios, '9.0'
use_frameworks!

target 'MLVisionExample' do
  pod 'FirebaseMLVision', '0.16.0'
  pod 'FirebaseMLCommon', '0.16.0'
  pod 'FirebaseMLVisionAutoML', '0.16.0'
  pod 'Firebase/Analytics'
  pod 'Firebase/Auth'
  pod 'Firebase/Firestore'
end
~
~
~
```

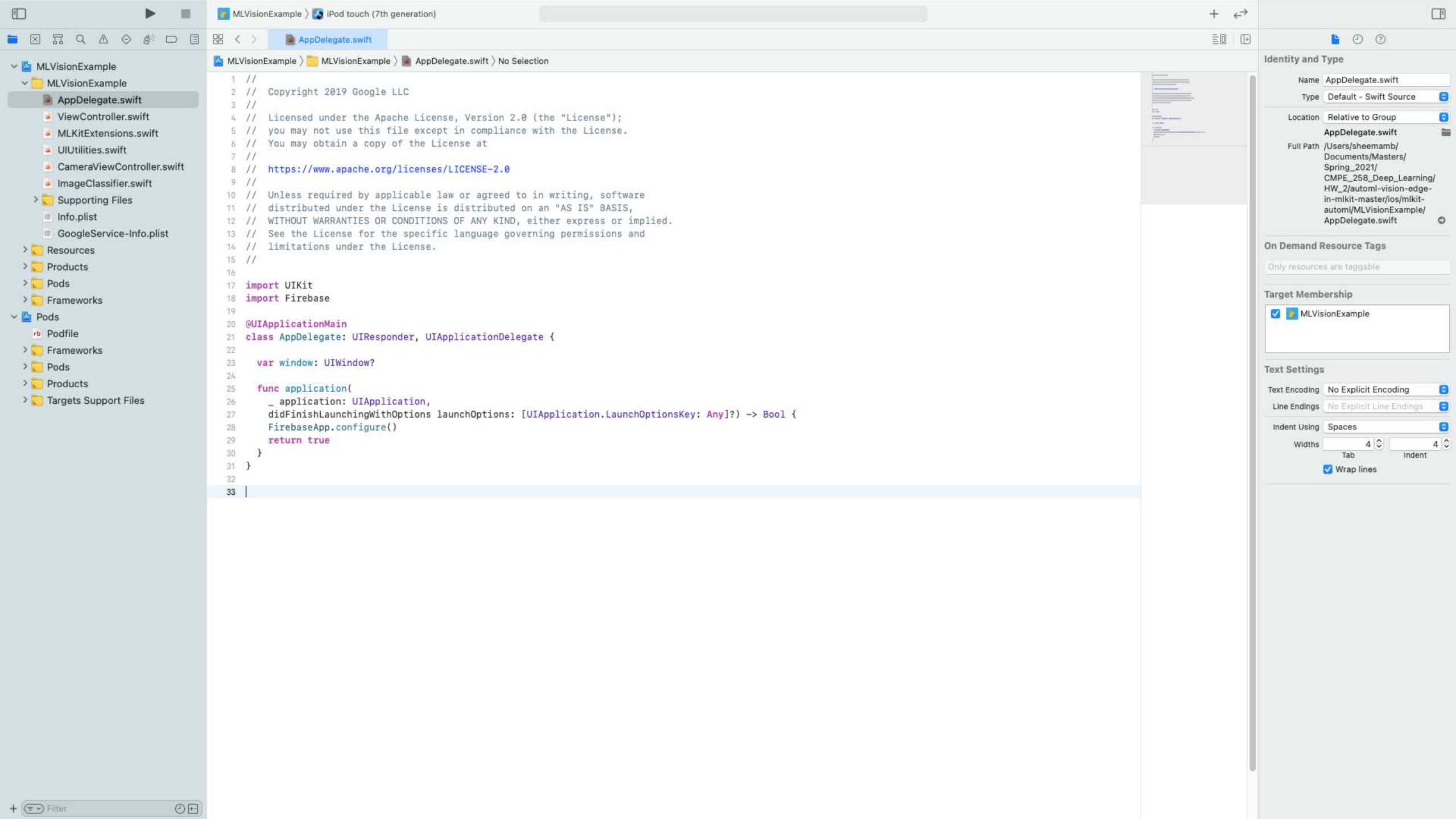
mlkit-automl — -zsh — 146×38

```
Installing GoogleToolboxForMac (2.2.0)
Installing GoogleUtilities (6.0.0)
Installing Protobuf (3.7.0)
Installing TensorFlowLite (1.13.1)
Installing nanopb (0.3.9011)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `MLVisionExample.xcworkspace` for this project from now on.
Pod installation complete! There are 4 dependencies from the Podfile and 16 total pods installed.

[!] FirebaseMLCommon has been deprecated

[!] FirebaseMLVision has been deprecated

[!] FirebaseMLVisionAutoML has been deprecated in favor of MLKitImageLabelingAutoML
(base) sheemamb@Sheemas-MacBook-Pro mlkit-automl % vim Podfile
(base) sheemamb@Sheemas-MacBook-Pro mlkit-automl % pod install
Analyzing dependencies
Downloading dependencies
Installing BoringSSL-GRPC (0.0.3)
Installing Firebase 6.0.0
Installing FirebaseAuth (6.0.0)
Installing FirebaseAuthInterop (1.1.0)
Installing FirebaseFirestore (1.3.2)
Installing gRPC-C++ (0.0.9)
Installing gRPC-Core (1.21.0)
Installing leveldb-library (1.22.1)
Generating Pods project
Integrating client project
Pod installation complete! There are 6 dependencies from the Podfile and 23 total pods installed.

[!] FirebaseMLCommon has been deprecated

[!] FirebaseMLVision has been deprecated

[!] FirebaseMLVisionAutoML has been deprecated in favor of MLKitImageLabelingAutoML
(base) sheemamb@Sheemas-MacBook-Pro mlkit-automl % open MLVisionExample.xcworkspace
```


MLVisionExample
iPod touch (7th generation)
Pods.xcodeproj
Buildtime (2)
Runtime
MLVisionExample project 1 is...
Validate Project Settings
Update to recommended settings
MLVisionExample.xcodeproj
Pods project 1 issue
Validate Project Settings
Update to recommended settings
Pods.xcodeproj
Pods
General
Signing & Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
PROJECT
Pods
TARGETS
BoringSSL-GRPC
Firebase
FirebaseAnalytics
FirebaseAuth
FirebaseAuthInterop
FirebaseCore
FirebaseFirestore
FirebaseInstanceID
FirebaseMLComm...
FirebaseMLVision
FirebaseMLVision...
GoogleAPIClientF...
GoogleAppMeasur...
GoogleMobileVision
GoogleToolboxFor...
GoogleUtilities
gRPC-C++
gRPC-C++-gRPC...
gRPC-Core
GTMSessionFetcher
leveldb-library
nanopb
Pods-MLVisionExa...
Protobuf
TensorFlowLite
Identity
Display Name
openssl_grpc
Bundle Identifier
org.cocoapods.openssl-grpc
Version
0.0.3
Build
1
Deployment Info
iOS 8.0
iPhone
iPad
macOS 10.15
Mac
Scale Interface to Match iPad
Show "Designed for iPad" Run Destination
App Extensions
Allow app extension API only
Frameworks and Libraries
Name
Platforms
Embed
Foundation.framework
macOS + iOS
Do Not Embed
Development Assets
Add development assets here
Identity and Type
Name
Pods
Location
Relative to Group
Pods/Pods.xcodeproj
Full Path
/Users/sheemamb/Documents/Masters/Spring_2021/CMPE_258_Deep_Learning/HW_2/automl-vision-edge-in-mlkit-master/ios/mlkit-automl/Pods/Pods.xcodeproj
Project Document
Project Format
Xcode 8.0-compatible
Organization
Class Prefix
Text Settings
Indent Using
Spaces
Widths
4
4
Tab
Indent
Wrap lines


MLVisionExample
iPod touch (7th generation)
Buildtime (2)
Runtime
MLVisionExample project 1 is...
Validate Project Settings
Update to recommended settings
MLVisionExample.xcodeproj
Pods project 1 issue
Validate Project Settings
Update to recommended settings
Pods.xcodeproj
Info
Build Settings
Swift Packages
PROJECT
MLVisionExample
TARGETS
MLVisionExample
Deployment Target
iOS Deployment Target
11.2
Configurations
Name
Based on Configuration File
Debug
1 Configuration Set
Release
1 Configuration Set
Use
Release
Parallelize build for command-line
Localizations
Localization
Base
English — Development Language
Use Base Internationalization
Build Settings
Target 'MLVisionExample' - Update iOS Deployment Target
This will update the minimum deployment target of Target 'MLVisionExample' to '12.0'.
Project 'MLVisionExample' - Update iOS Deployment Target
This will update the minimum deployment target of Project 'MLVisionExample' to '12.0'.
Project 'MLVisionExample' - Enable Recommended Warning
This will enable the following recommended compiler warning:
- Quoted Include In Framework Header
Cancel
Perform Changes
Identity and Type
Name
MLVisionExample
Location
Relative to Group
MLVisionExample.xcodeproj
Full Path
/Users/sheemamb/Documents/Masters/Spring_2021/CMPE_258_Deep_Learning/HW_2/automl-vision-edge-in-mlkit-master/ios/mlkit-automl/MLVisionExample.xcodeproj
Project Document
Project Format
Xcode 8.0-compatible
Organization
Google Inc.
Class Prefix
Text Settings
Indent Using
Spaces
Widths
Tab
Indent
Wrap lines


MLVisionExample
iPod touch (7th generation)
Buildtime (1)
Runtime
Pods project 1 issue
Validate Project Settings
Update to recommended settings
Pods.xcodeproj
Pods
Info
Build Settings
Swift Packages
PROJECT
Pods
TARGETS
BoringSSL-GRPC
Firebase
FirebaseAnalytics
FirebaseAuth
FirebaseAuthInterop
FirebaseCore
FirebaseFirestore
FirebaseInstanceID
FirebaseMLComm...
FirebaseMLVision
FirebaseMLVision...
GoogleAPIClientF...
GoogleAppMeasur...
GoogleMobileVision
GoogleToolboxFor...
GoogleUtilities
gRPC-C++
gRPC-C++-gRPC...
gRPC-Core
GTMSessionFetcher
leveldb-library
nanopb
Pods-MLVisionExa...
Protobuf
TensorFlowLite
Deployment Target
iOS Deployment Target
9.0
Configurations
Name
Based on Configuration File
Debug
25 Configurations Set
Release
25 Configurations Set
Use
Release
Parallelize build for command-lin
Localizations
Localization
Base
English — Development Language
Use Base Internationalization
Identity and Type
Name
Pods
Location
Relative to Group
Pods/Pods.xcodeproj
Full Path
/Users/sheemamb/Documents/Masters/Spring_2021/CMPE_258_Deep_Learning/HW_2/automl-vision-edge-in-mlkit-master/ios/mlkit-automl/Pods/Pods.xcodeproj
Project Document
Project Format
Xcode 8.0-compatible
Organization
Class Prefix
Text Settings
Indent Using
Spaces
Widths
4
Tab
4
Indent
Wrap lines
Build Settings
Target 'BoringSSL-GRPC' - Update iOS Deployment Target
This will update the minimum deployment target of Target 'BoringSSL-GRPC' to '12.0'.
Target 'Firebase' - Update iOS Deployment Target
This will update the minimum deployment target of Target 'Firebase' to '12.0'.
Target 'FirebaseAnalytics' - Update iOS Deployment Target
This will update the minimum deployment target of Target 'FirebaseAnalytics' to '12.0'.
Target 'FirebaseAuth' - Update iOS Deployment Target
This will update the minimum deployment target of Target 'FirebaseAuth' to '12.0'.
Target 'FirebaseAuthInterop' - Update iOS Deployment Target
This will update the minimum deployment target of Target 'FirebaseAuthInterop' to
Target 'FirebaseCore' - Update iOS Deployment Target
This will update the minimum deployment target of Target 'FirebaseCore' to '12.0'.
Target 'FirebaseFirestore' - Update iOS Deployment Target
This will update the minimum deployment target of Target 'FirebaseFirestore' to '12.0'.
Target 'FirebaseInstanceID' - Update iOS Deployment Target
This will update the minimum deployment target of Target 'FirebaseInstanceID' to '12.0'.
Target 'FirebaseMLCommon' - Update iOS Deployment Target
Cancel
Perform Changes


MLVisionExample
iPod touch (7th generation)
AppDelegate.swift
MLVisionExample
MLVisionExample
AppDelegate.swift
ViewController.swift
MLKitExtensions.swift
UIUtilities.swift
CameraViewController.swift
ImageClassifier.swift
Supporting Files
Info.plist
GoogleService-Info.plist
Resources
Products
Pods
Frameworks
Pods
Podfile
Frameworks
Pods
Products
Targets Support Files
MLVisionExample > MLVisionExample > AppDelegate.swift > No Selection
//
// Copyright 2019 Google LLC
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// https://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
//
import UIKit
import Firebase
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
var window: UIWindow?
func application(
_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
FirebaseApp.configure()
return true
}
}
Identity and Type
Name AppDelegate.swift
Type Default - Swift Source
Location Relative to Group
AppDelegate.swift
Full Path /Users/sheemamb/Documents/Masters/Spring_2021/CMPE_258_Deep_Learning/HW_2/automl-vision-edge-in-mlkit-master/ios/mlkit-automl/MLVisionExample/AppDelegate.swift
On Demand Resource Tags
Only resources are taggable
Target Membership
MLVisionExample
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 Tab 4 Indent
Wrap lines

Add Firebase to your iOS app

1. Register app
   iOS bundle ID: com.google.firebase.codelab.mlkit.automl, App nickname: Sheema iOS App
2. Download config file
3. Add Firebase SDK
4. Add initialization code
5. **Next steps**

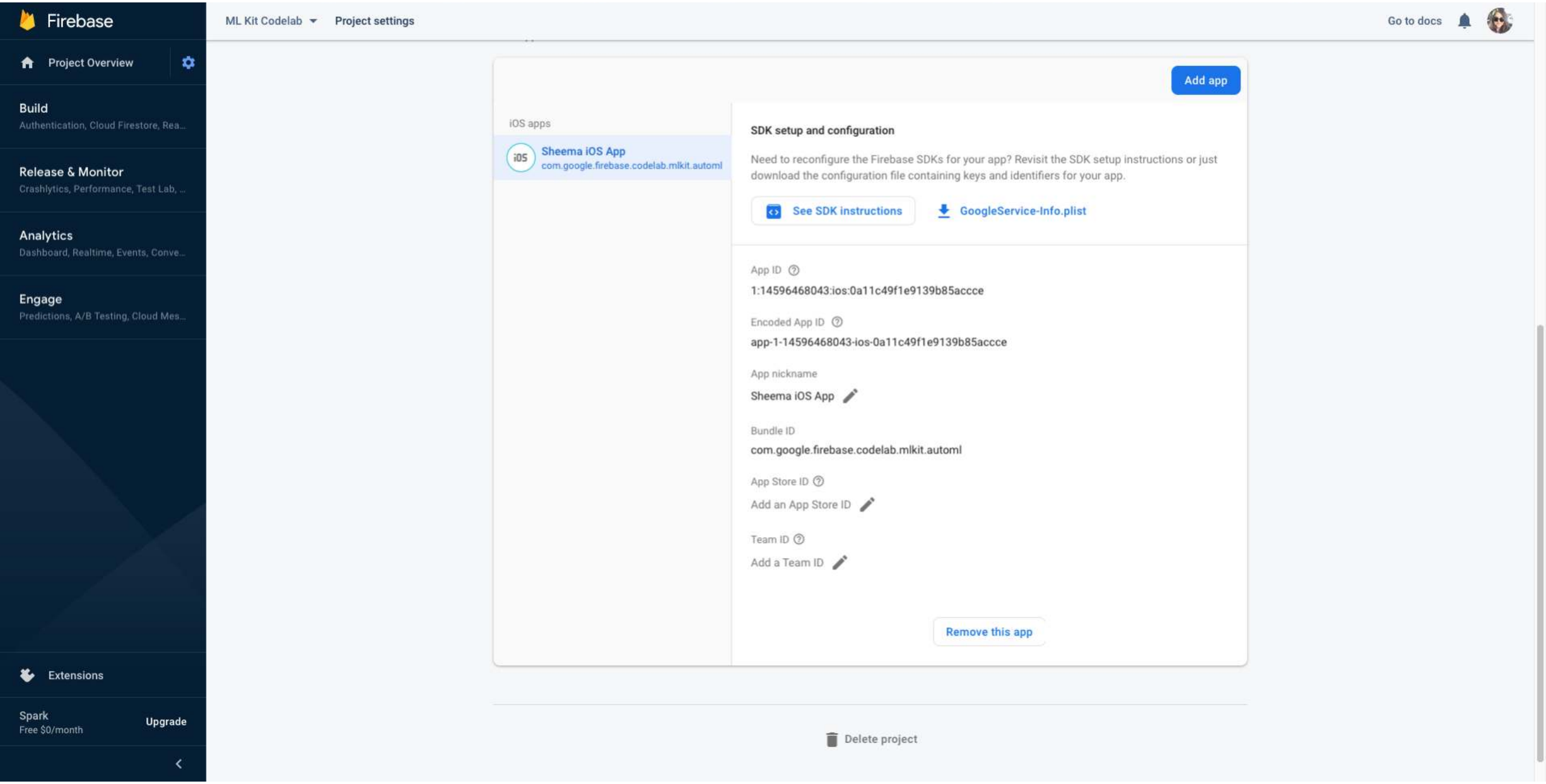
You're all set!

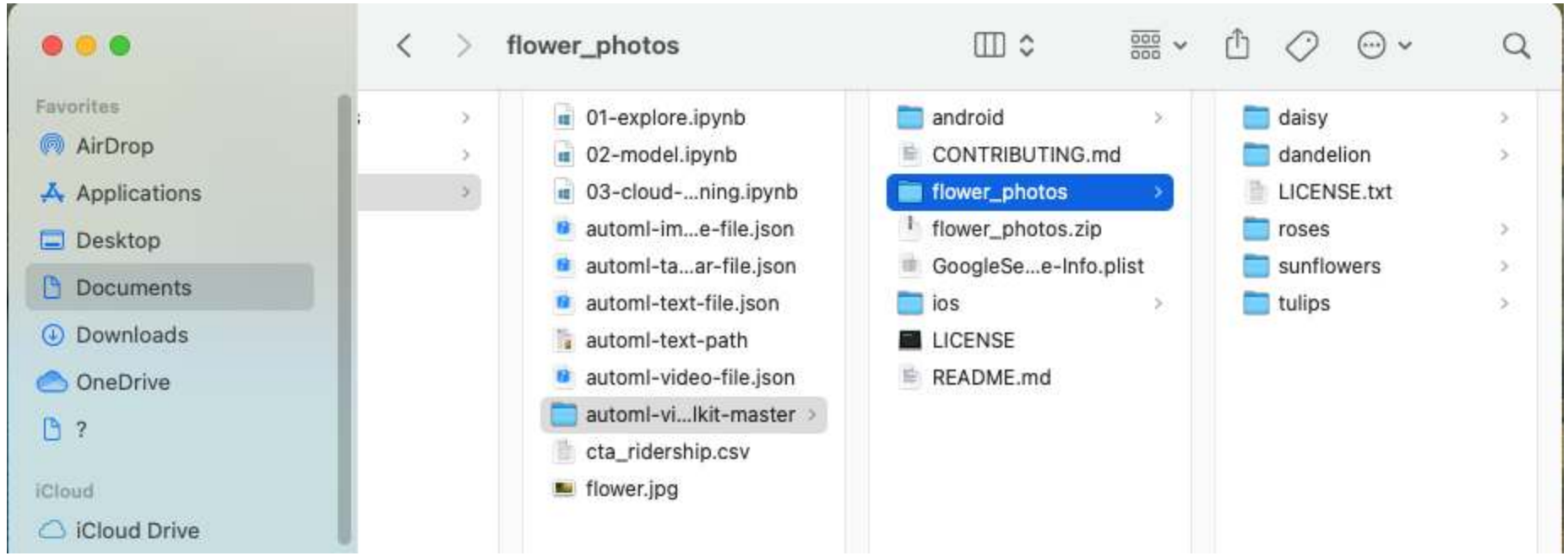
Make sure to check out the documentation to learn how to get started with each Firebase product that you want to use in your app.

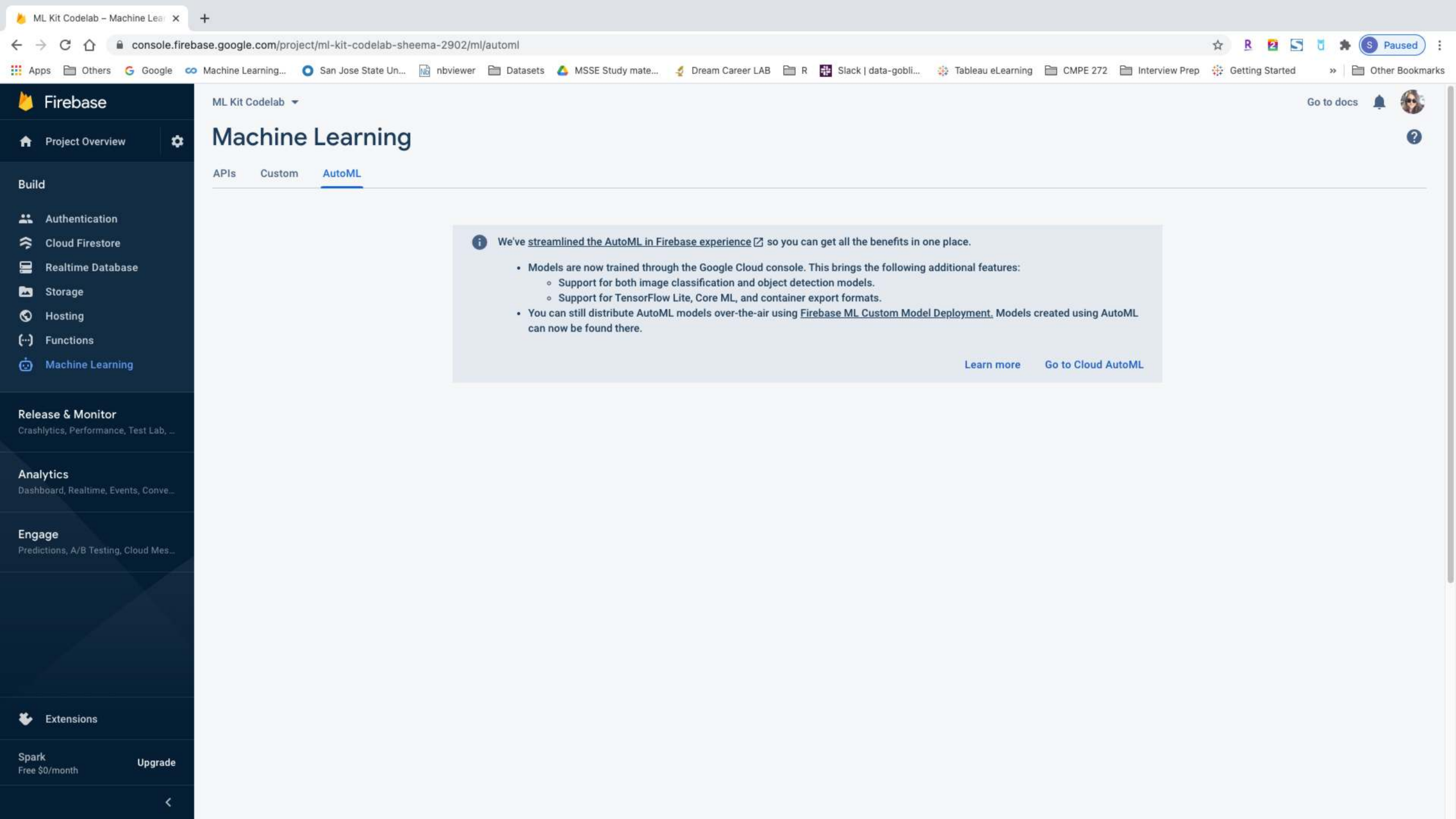
You can also explore sample Firebase apps.

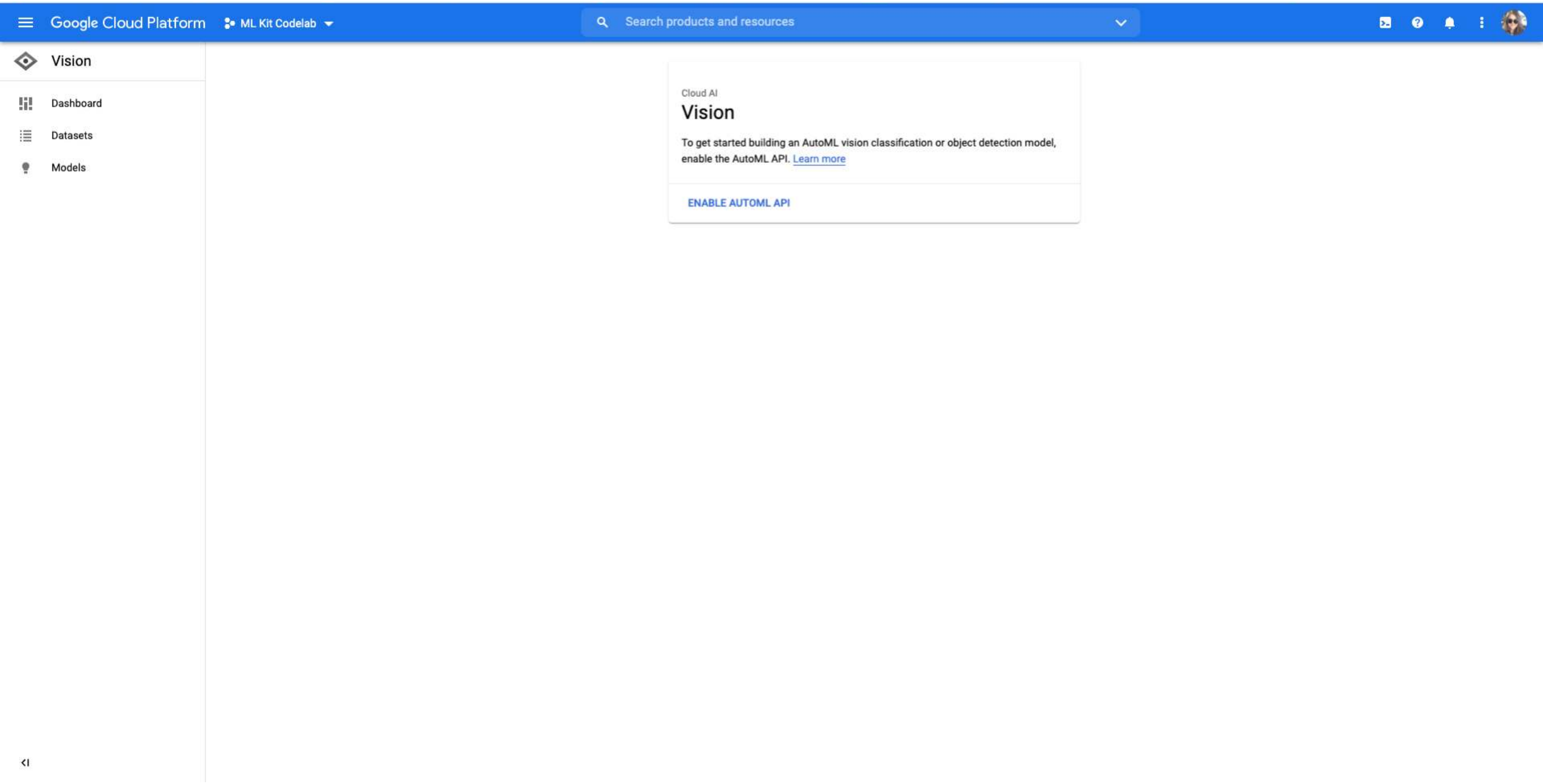
Or, continue to the console to explore Firebase.

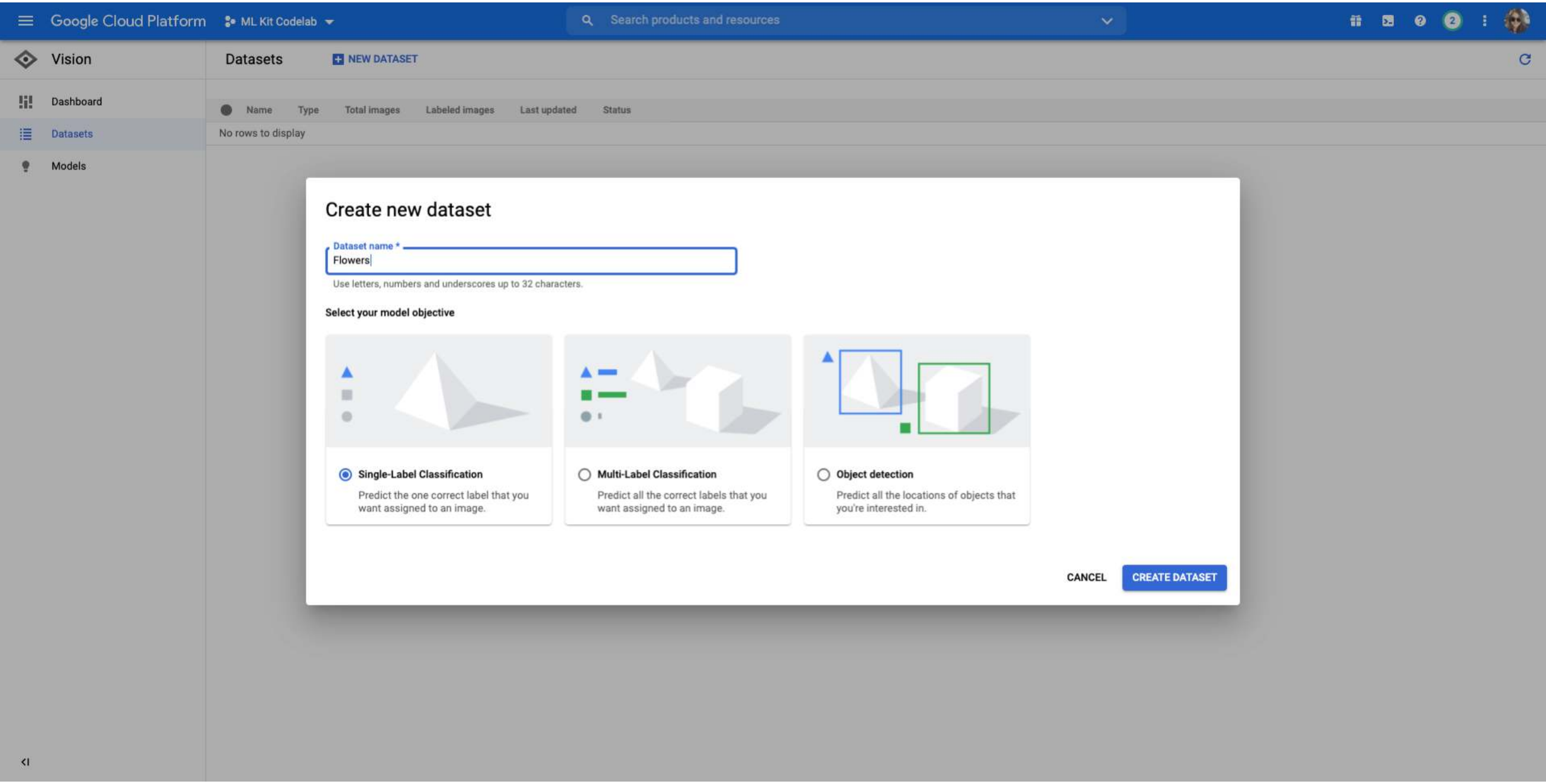
Previous | Continue to console

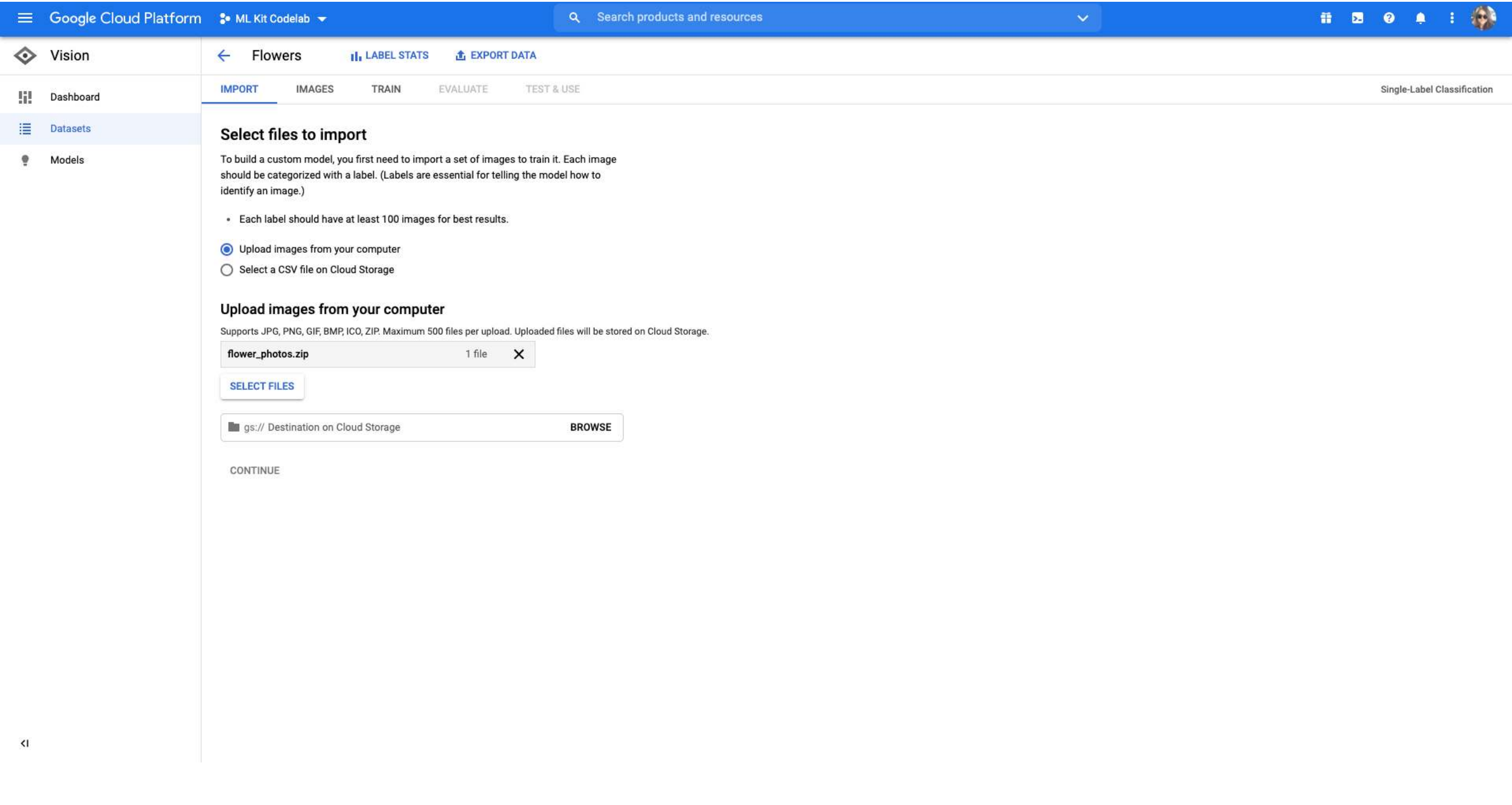

Firebase
ML Kit Codelab
Project settings
Go to docs
Project Overview
Build
Authentication, Cloud Firestore, Rea...
Release & Monitor
Crashlytics, Performance, Test Lab, ...
Analytics
Dashboard, Realtime, Events, Conve...
Engage
Predictions, A/B Testing, Cloud Mes...
Extensions
Spark
Free $0/month
Upgrade
Add app
iOS apps
iOS
Sheema 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
1:14596468043:ios:0a11c49f1e9139b85accce
Encoded App ID
app-1-14596468043-ios-0a11c49f1e9139b85accce
App nickname
Sheema 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
Delete project

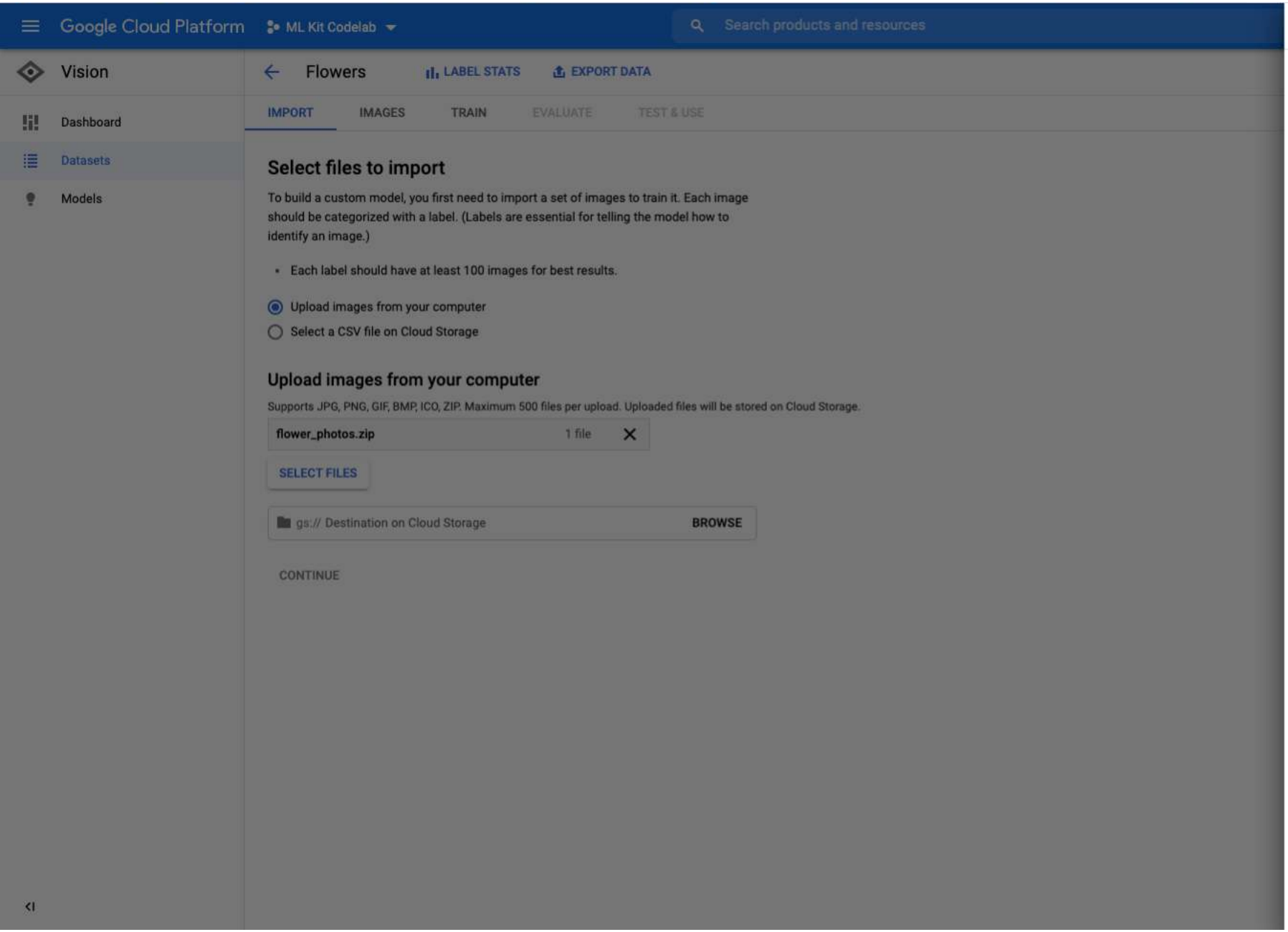

flower_photos
Favorites
AirDrop
Applications
Desktop
Documents
Downloads
OneDrive
?
iCloud
iCloud Drive
01-explore.ipynb
02-model.ipynb
03-cloud-...ning.ipynb
automl-im...e-file.json
automl-ta...ar-file.json
automl-text-file.json
automl-text-path
automl-video-file.json
automl-vi...lkit-master
cta_ridership.csv
flower.jpg
android
CONTRIBUTING.md
flower_photos
flower_photos.zip
GoogleSe...e-Info.plist
ios
LICENSE
README.md
daisy
dandelion
LICENSE.txt
roses
sunflowers
tulips


ML Kit Codelab – Machine Lea
console.firebase.google.com/project/ml-kit-codelab-sheema-2902/ml/automl
Paused
Apps
Others
Google
Machine Learning...
San Jose State Un...
nbviewer
Datasets
MSSE Study mate...
Dream Career LAB
R
Slack | data-gobli...
Tableau eLearning
CMPE 272
Interview Prep
Getting Started
Other Bookmarks
Firebase
ML Kit Codelab
Go to docs
Project Overview
Machine Learning
Build
Authentication
Cloud Firestore
Realtime Database
Storage
Hosting
Functions
Machine Learning
APIs
Custom
AutoML
We've streamlined the AutoML in Firebase experience so you can get all the benefits in one place.
Models are now trained through the Google Cloud console. This brings the following additional features:
Support for both image classification and object detection models.
Support for TensorFlow Lite, Core ML, and container export formats.
You can still distribute AutoML models over-the-air using Firebase ML Custom Model Deployment. Models created using AutoML can now be found there.
Learn more
Go to Cloud AutoML
Release & Monitor
Crashlytics, Performance, Test Lab, ...
Analytics
Dashboard, Realtime, Events, Conve...
Engage
Predictions, A/B Testing, Cloud Mes...
Extensions
Spark
Free $0/month
Upgrade


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Dashboard
Datasets
Models
Cloud AI
Vision
To get started building an AutoML vision classification or object detection model, enable the AutoML API. Learn more
ENABLE AUTOML API


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Datasets
NEW DATASET
Dashboard
Datasets
Models
Name
Type
Total images
Labeled images
Last updated
Status
No rows to display
Create new dataset
Dataset name *
Flowers
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


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Dashboard
Datasets
Models
Flowers
LABEL STATS
EXPORT DATA
IMPORT
IMAGES
TRAIN
EVALUATE
TEST & USE
Single-Label Classification
Select files to import
To build a custom model, you first need to import a set of images to train it. Each image should be categorized with a label. (Labels are essential for telling the model how to identify an image.)
Each label should have at least 100 images for best results.
Upload images from your computer
Select a CSV file on Cloud Storage
Upload images from your computer
Supports JPG, PNG, GIF, BMP, ICO, ZIP. Maximum 500 files per upload. Uploaded files will be stored on Cloud Storage.
flower_photos.zip
1 file
SELECT FILES
gs:// Destination on Cloud Storage
BROWSE
CONTINUE



Create a bucket

- **Name your bucket**

  Pick a **globally unique**, permanent name. Naming guidelines

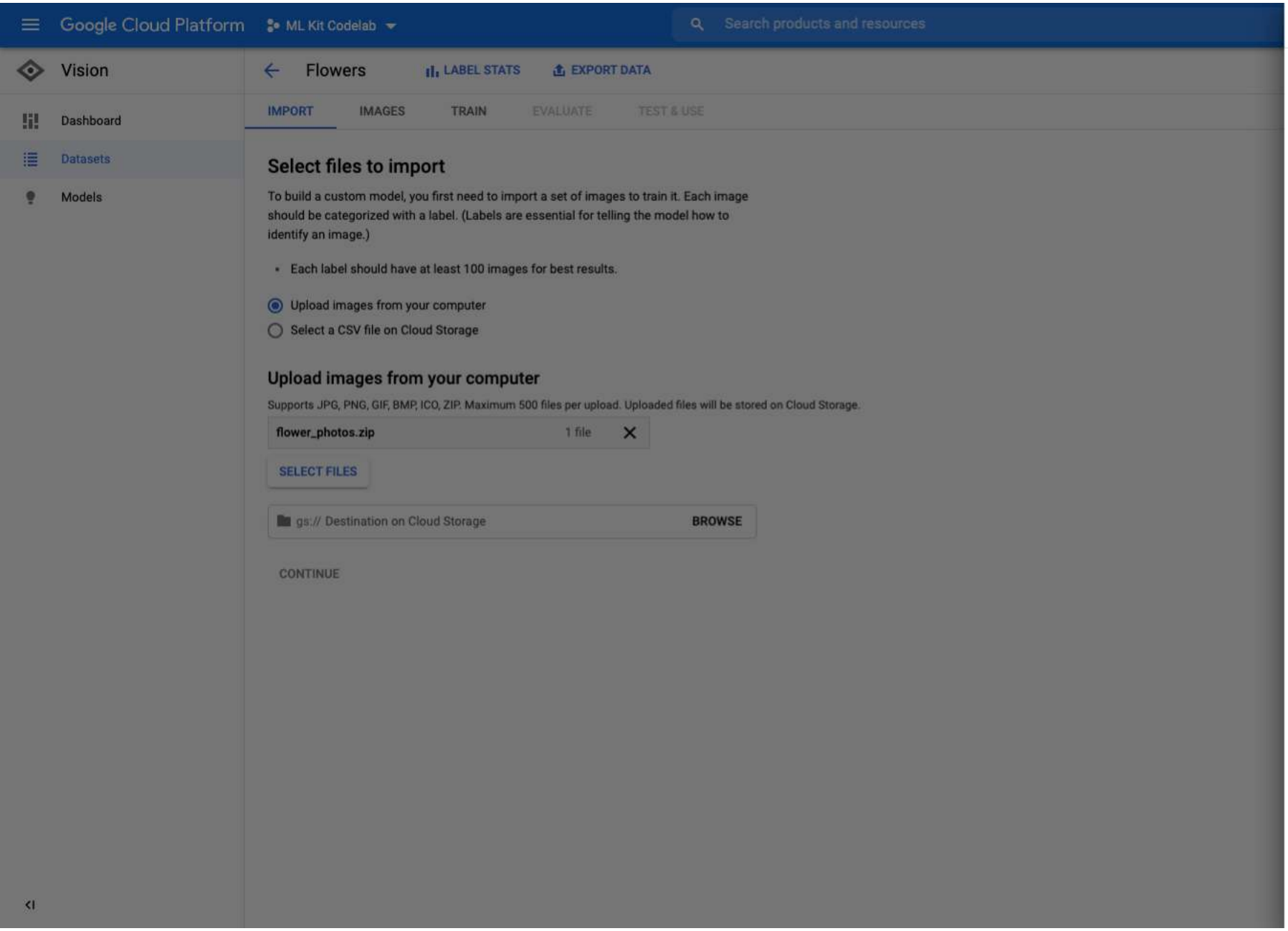
  flower-bucket-2902

  Tip: Don't include any sensitive information

  CONTINUE

- **Choose where to store your data**
- **Choose a default storage class for your data**
- **Choose how to control access to objects**
- **Advanced settings (optional)**

CREATE CANCEL



Create a bucket

Name your bucket

Choose where to store your data

This permanent choice defines the geographic placement of your data and affects cost, performance, and availability. Learn more

Location type

- Region
  Lowest latency within a single region
- Dual-region
  High availability and low latency across 2 regions
- Multi-region
  Highest availability across largest area

Location

us-east1 (South Carolina)

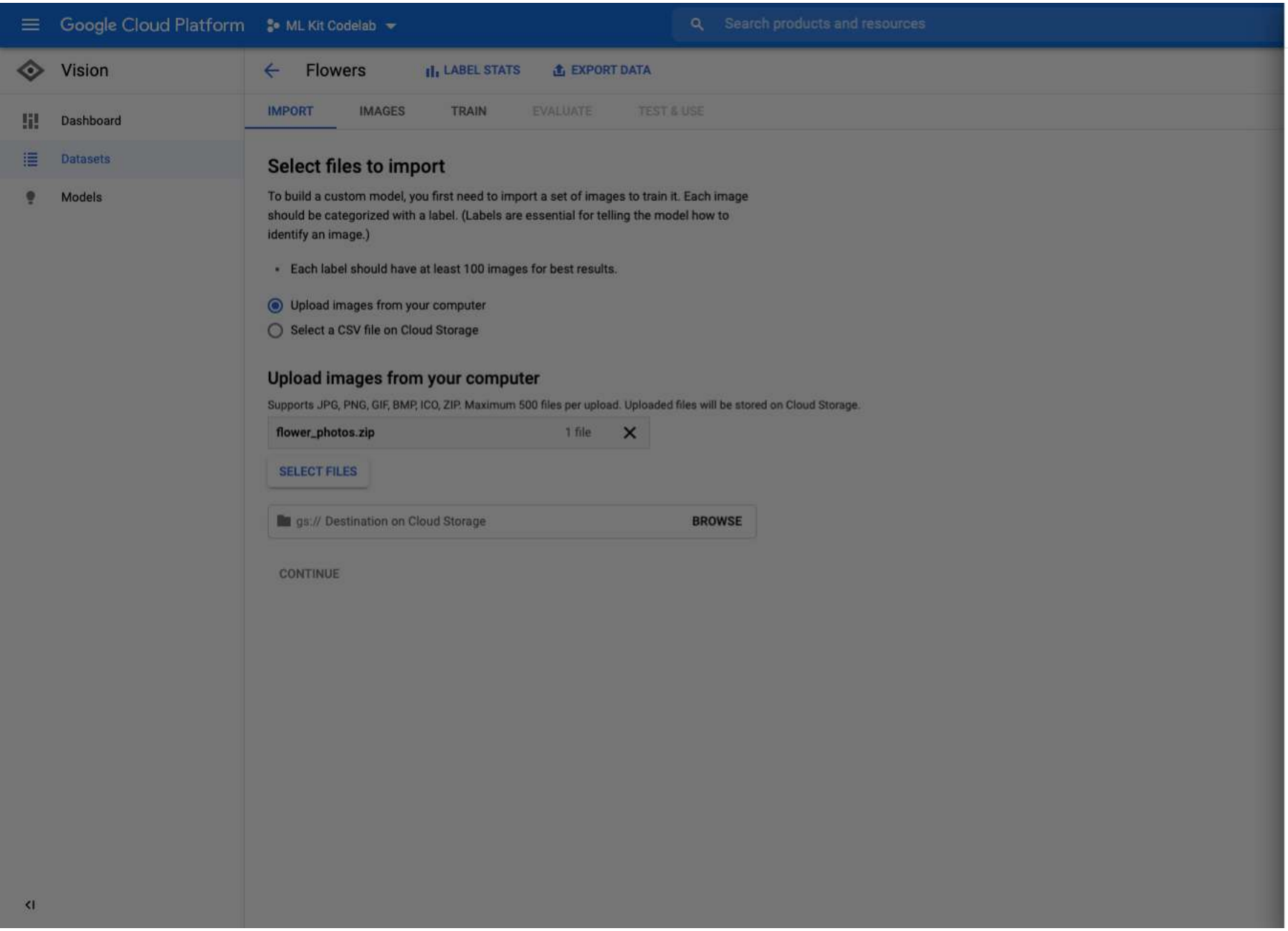
CONTINUE

Choose a default storage class for your data

Choose how to control access to objects

Advanced settings (optional)

CREATE CANCEL



Create a bucket

- Name your bucket
- Choose where to store your data
- Choose a default storage class for your data

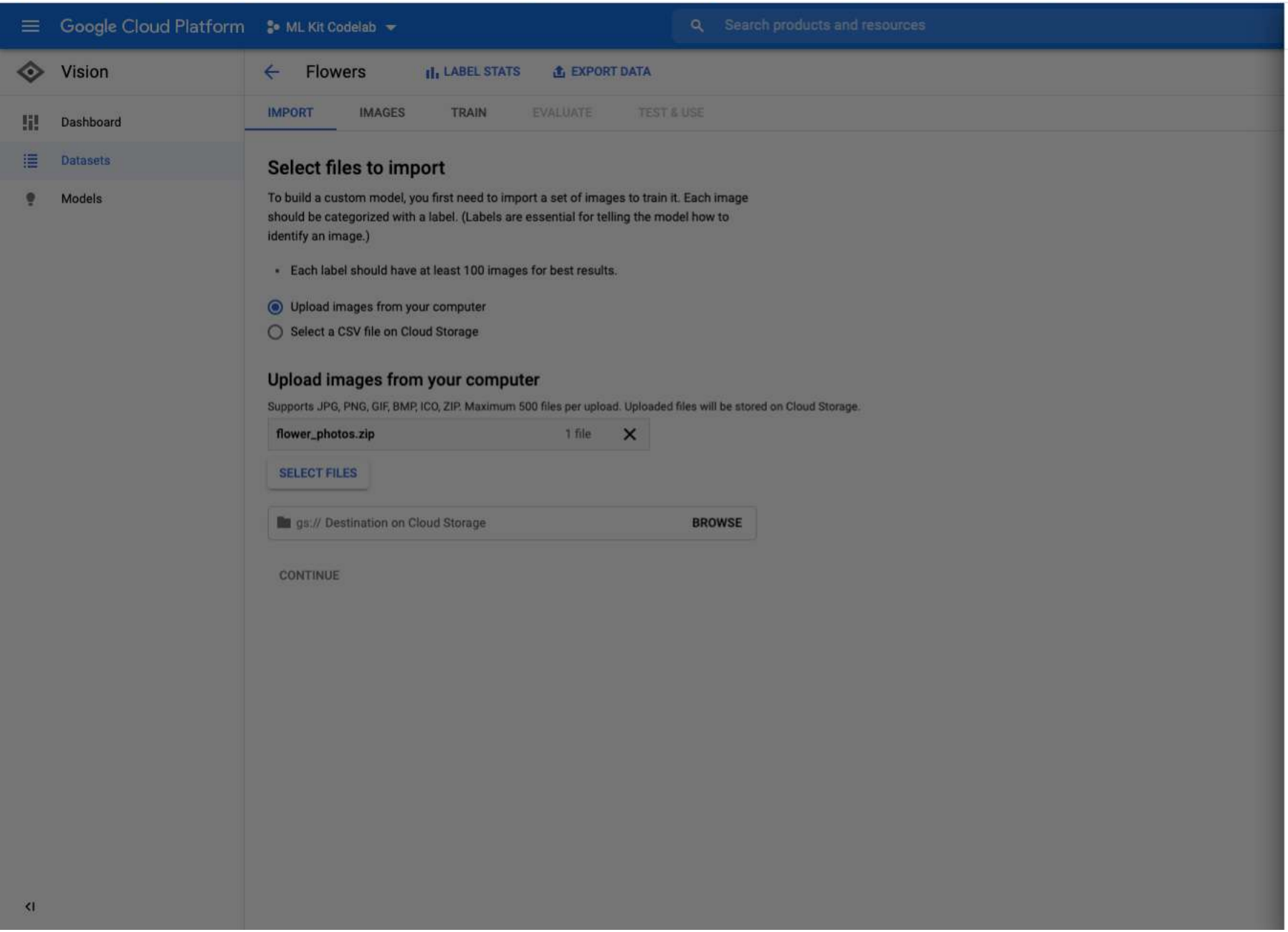
A storage class sets costs for storage, retrieval, and operations. Pick a default storage class based on how long you plan to store your data and how often it will be accessed. Learn more

- Standard
  Best for short-term storage and frequently accessed data
- Nearline
  Best for backups and data accessed less than once a month
- Coldline
  Best for disaster recovery and data accessed less than once a quarter
- Archive
  Best for long-term digital preservation of data accessed less than once a year

CONTINUE

- Choose how to control access to objects
- Advanced settings (optional)

CREATE CANCEL



Create a bucket

- Name your bucket
- Choose where to store your data
- Choose a default storage class for your data
- Choose how to control access to objects

Access control

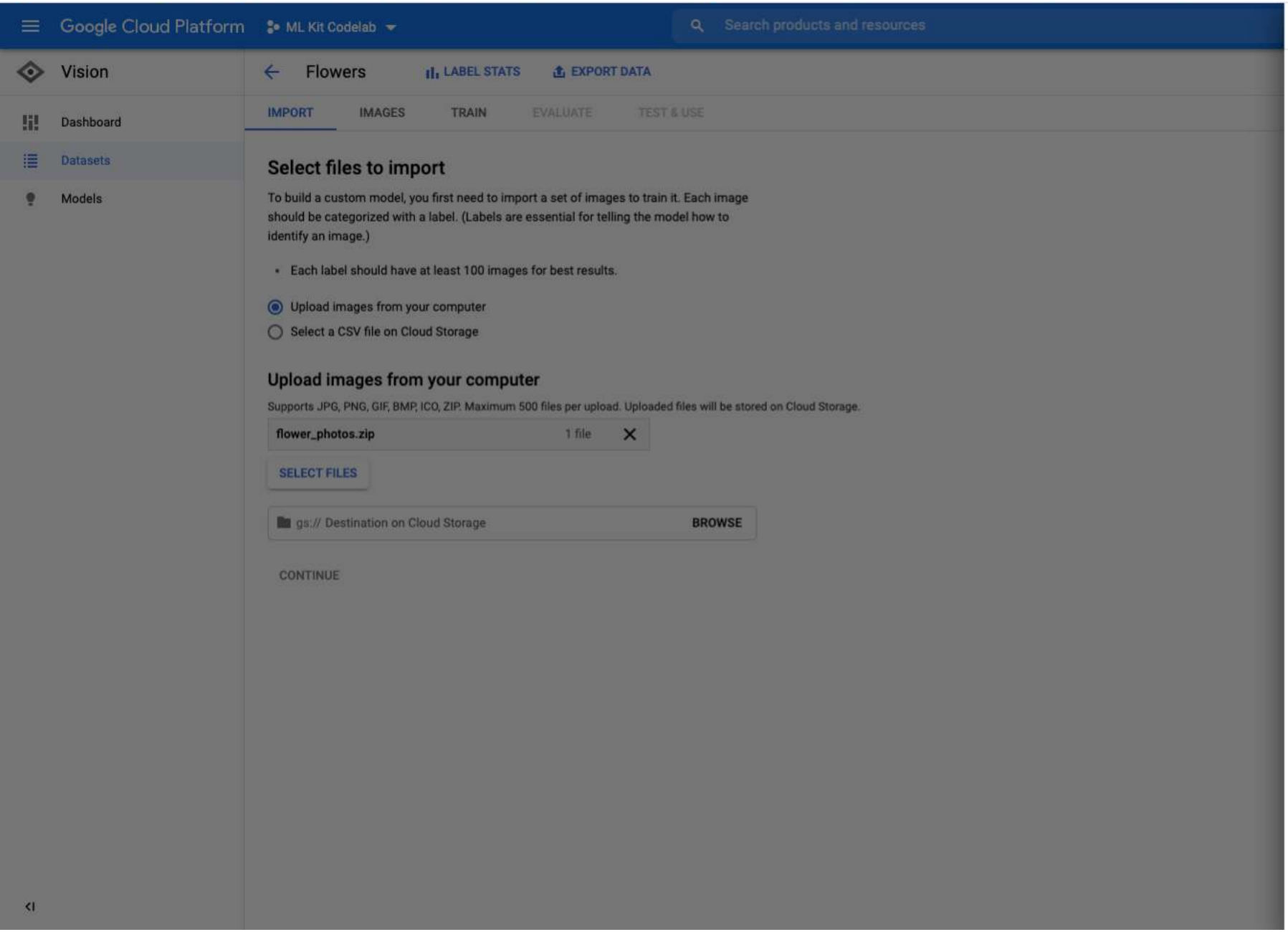
Fine-grained
Specify access to individual objects by using object-level permissions (ACLs) in addition to your bucket-level permissions (IAM). Learn more

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

CONTINUE

- Advanced settings (optional)

CREATE CANCEL



Create a bucket

- Name your bucket
- Choose where to store your data
- Choose a default storage class for your data
- Choose how to control access to objects
- Advanced settings (optional)

Encryption

Google-managed encryption key
No configuration required

Customer-managed encryption key (CMEK)
Manage via Google Cloud Key Management Service

Retention policy

Set a retention policy to specify the minimum duration that this bucket's objects must be protected from deletion or modification after they're uploaded. You might set a policy to address industry-specific retention challenges. Learn more

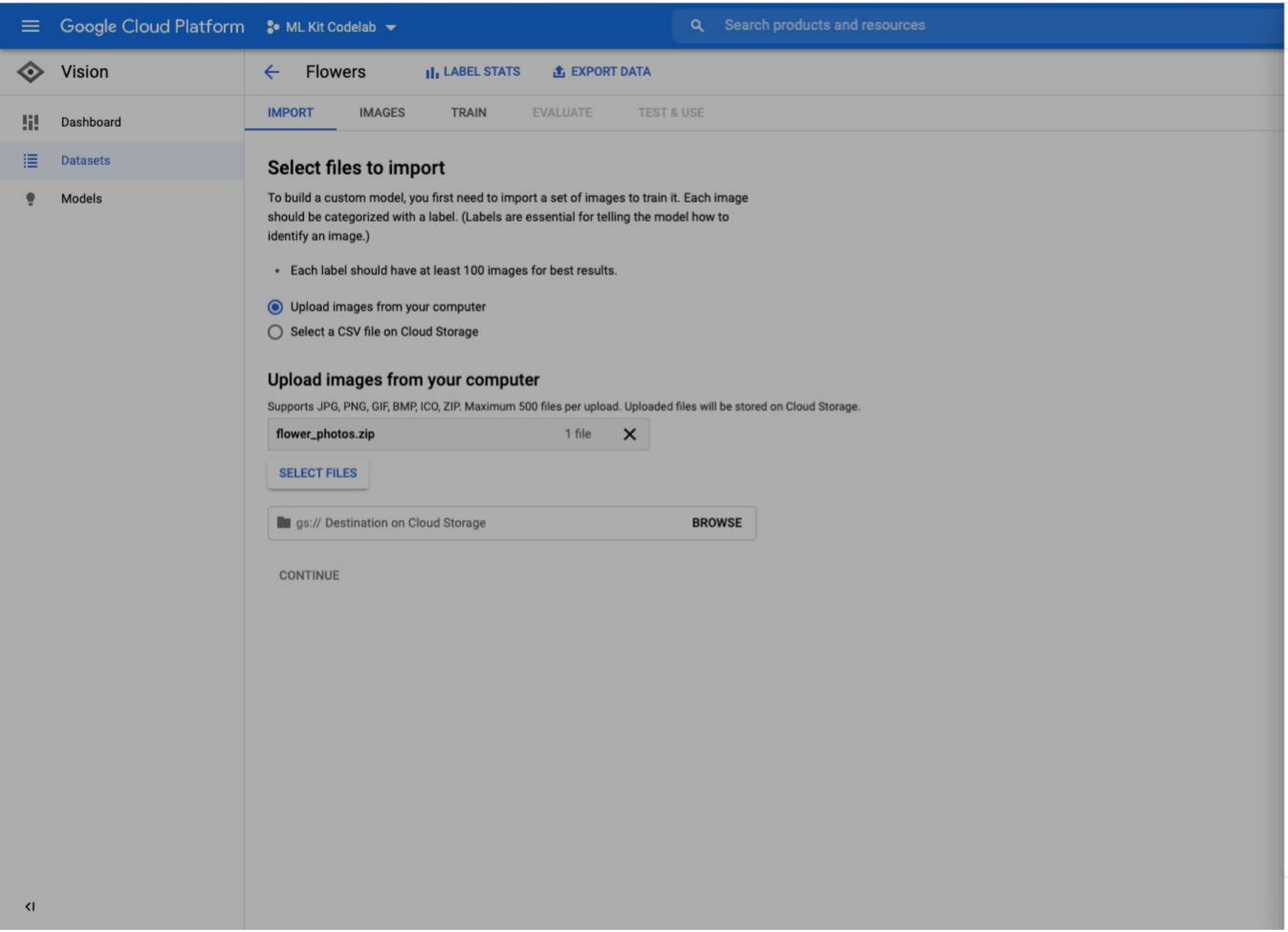
Set a retention policy

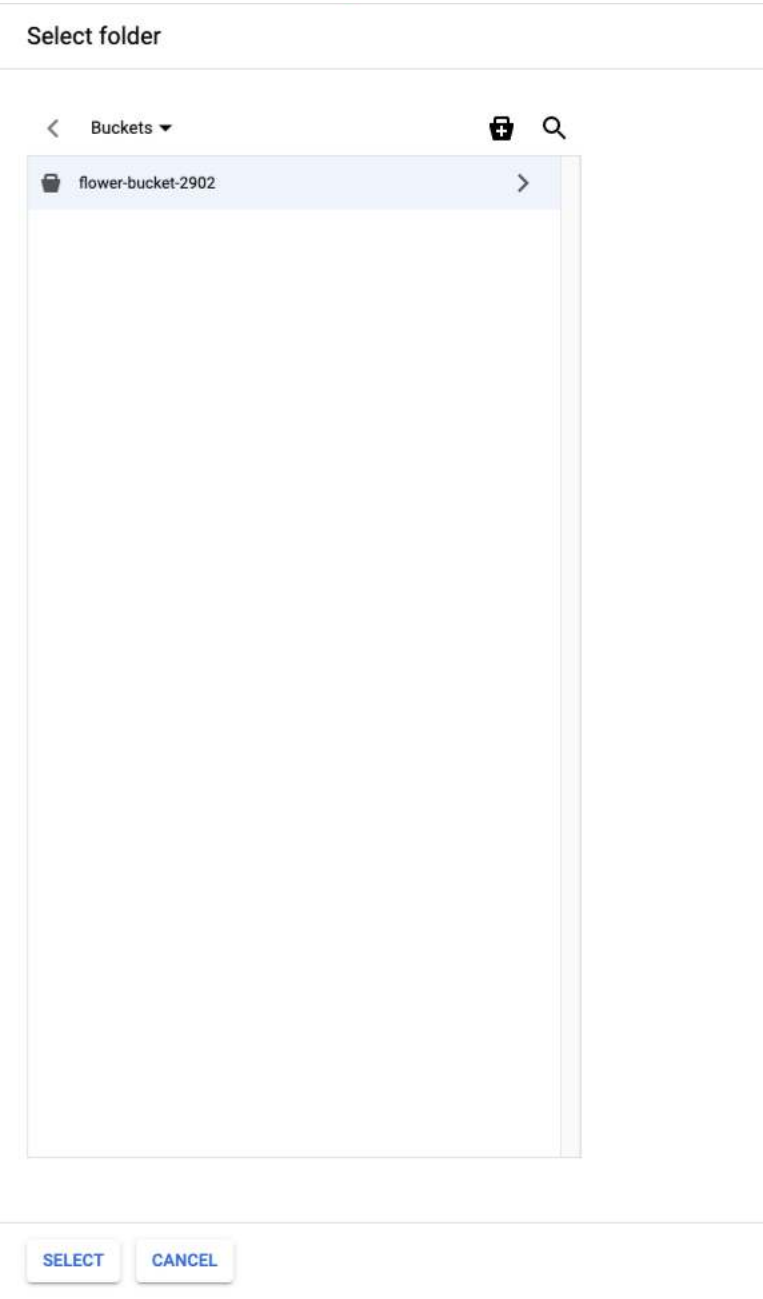
Labels

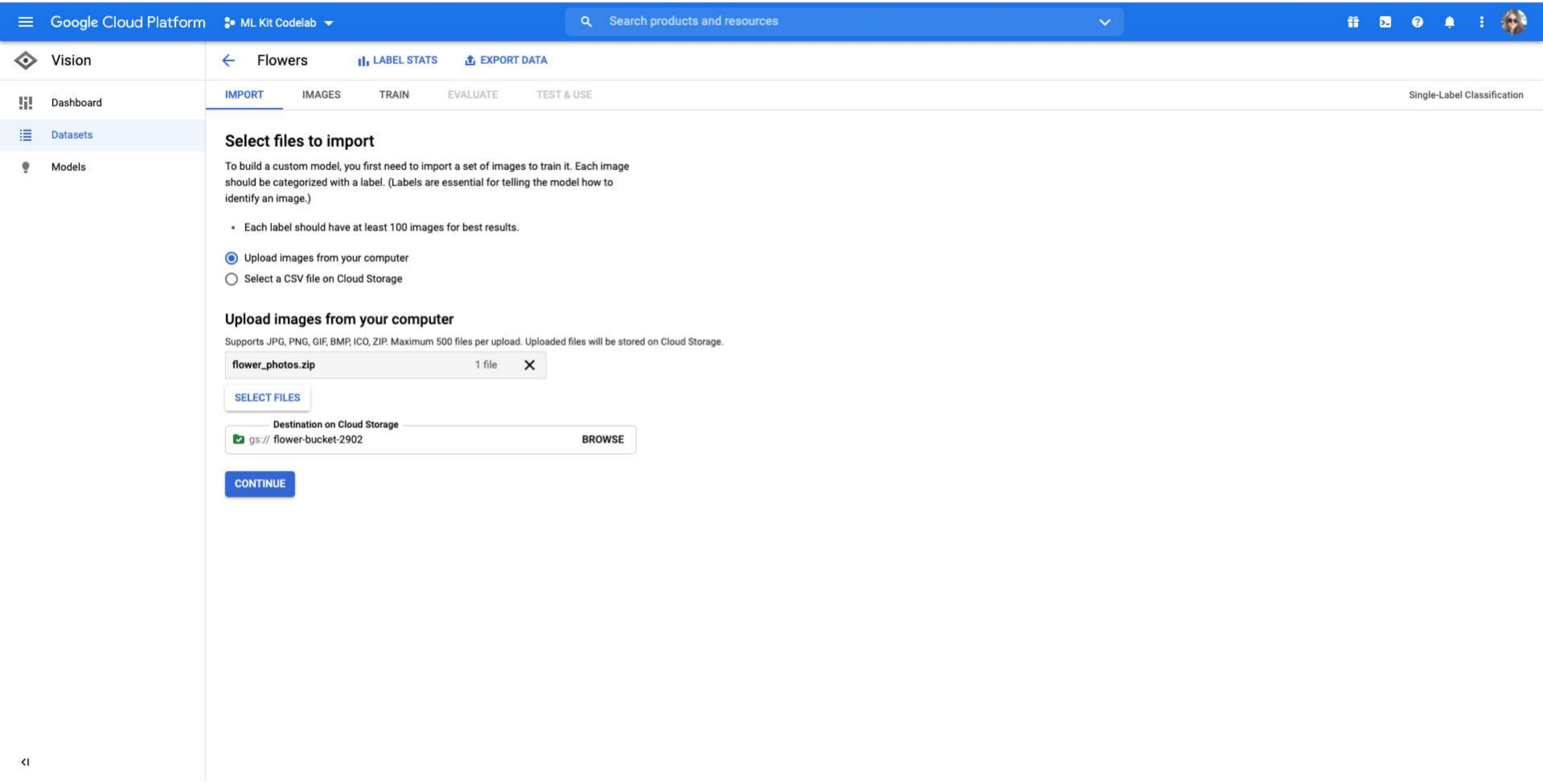
Labels are key:value pairs that allow you to group related buckets together or with other Cloud Platform resources. Learn more

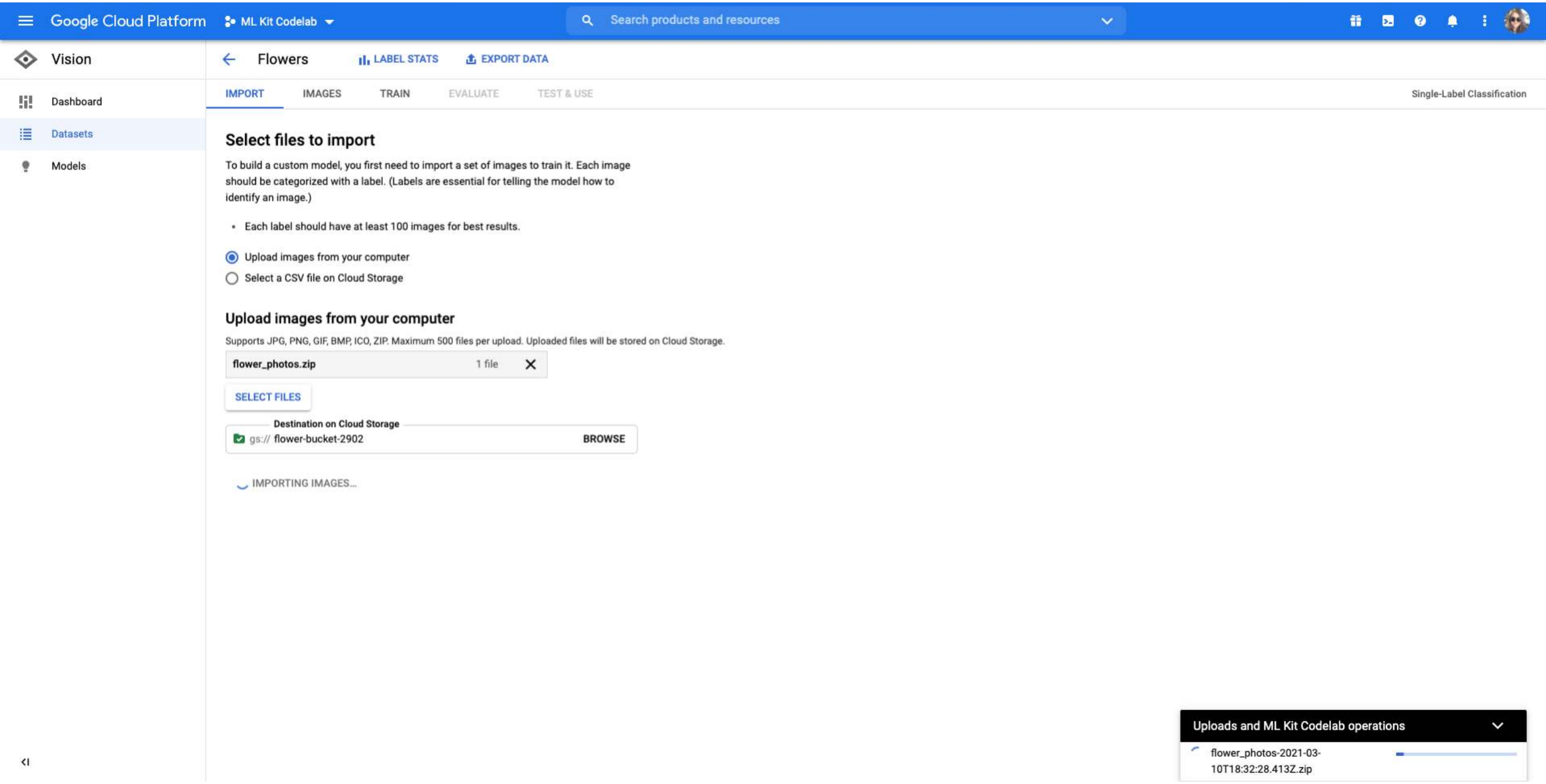
+ ADD LABEL

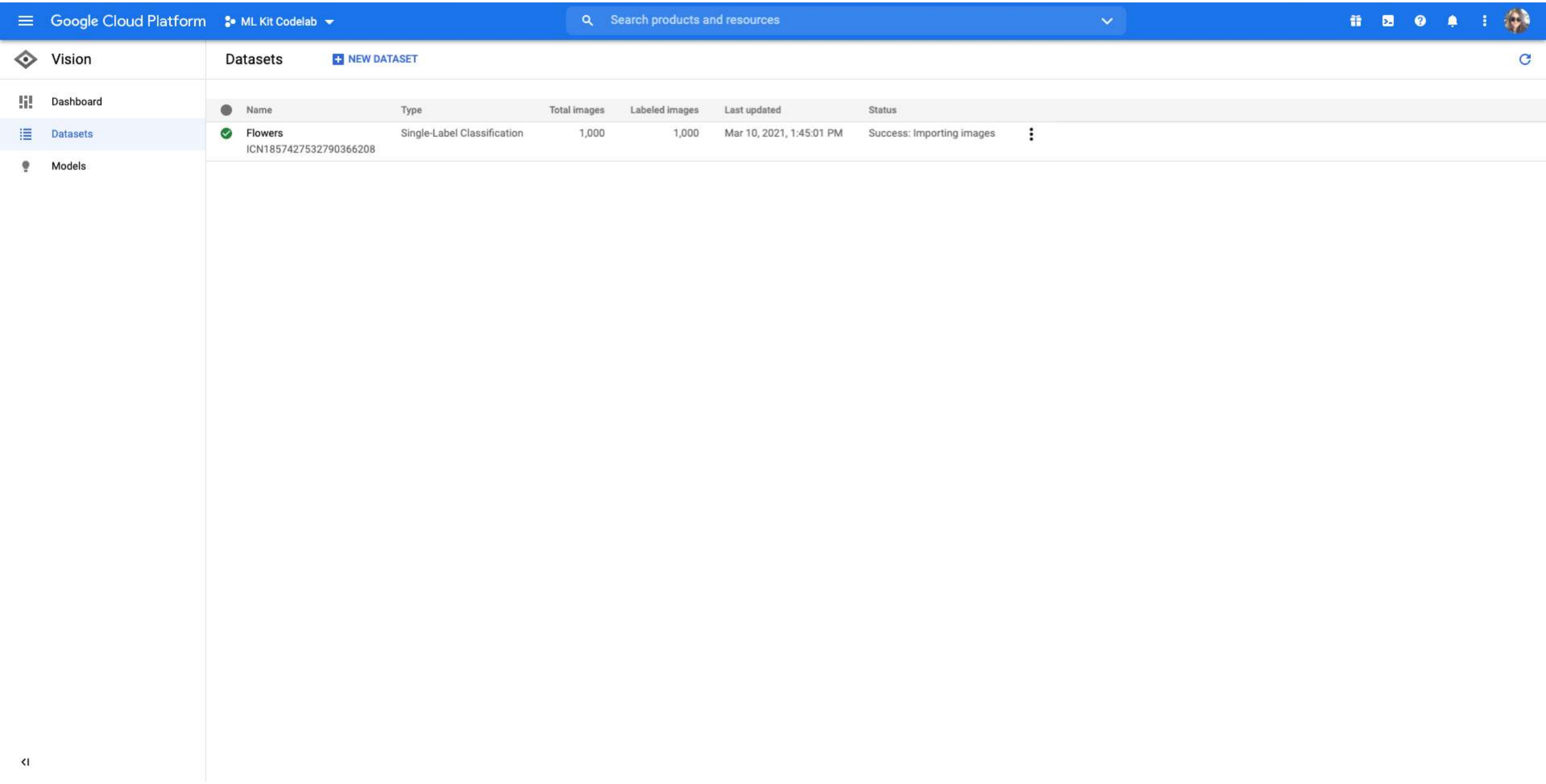
CREATE CANCEL


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Dashboard
Datasets
Models
Flowers
LABEL STATS
EXPORT DATA
IMPORT
IMAGES
TRAIN
EVALUATE
TEST & USE
Select files to import
To build a custom model, you first need to import a set of images to train it. Each image should be categorized with a label. (Labels are essential for telling the model how to identify an image.)
Each label should have at least 100 images for best results.
Upload images from your computer
Select a CSV file on Cloud Storage
Upload images from your computer
Supports JPG, PNG, GIF, BMP, ICO, ZIP. Maximum 500 files per upload. Uploaded files will be stored on Cloud Storage.
flower_photos.zip
1 file
SELECT FILES
gs:// Destination on Cloud Storage
BROWSE
CONTINUE


Select folder
Buckets
flower-bucket-2902
SELECT
CANCEL


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Dashboard
Datasets
Models
Flowers
LABEL STATS
EXPORT DATA
IMPORT
IMAGES
TRAIN
EVALUATE
TEST & USE
Single-Label Classification
Select files to import
To build a custom model, you first need to import a set of images to train it. Each image should be categorized with a label. (Labels are essential for telling the model how to identify an image.)
Each label should have at least 100 images for best results.
Upload images from your computer
Select a CSV file on Cloud Storage
Upload images from your computer
Supports JPG, PNG, GIF, BMP, ICO, ZIP. Maximum 500 files per upload. Uploaded files will be stored on Cloud Storage.
flower_photos.zip
1 file
SELECT FILES
Destination on Cloud Storage
gs:// flower-bucket-2902
BROWSE
CONTINUE


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Dashboard
Datasets
Models
Flowers
LABEL STATS
EXPORT DATA
IMPORT
IMAGES
TRAIN
EVALUATE
TEST & USE
Single-Label Classification
Select files to import
To build a custom model, you first need to import a set of images to train it. Each image should be categorized with a label. (Labels are essential for telling the model how to identify an image.)
Each label should have at least 100 images for best results.
Upload images from your computer
Select a CSV file on Cloud Storage
Upload images from your computer
Supports JPG, PNG, GIF, BMP, ICO, ZIP. Maximum 500 files per upload. Uploaded files will be stored on Cloud Storage.
flower_photos.zip
1 file
SELECT FILES
Destination on Cloud Storage
gs:// flower-bucket-2902
BROWSE
IMPORTING IMAGES...
Uploads and ML Kit Codelab operations
flower_photos-2021-03-10T18:32:28.413Z.zip


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Dashboard
Datasets
Models
Datasets
NEW DATASET
Name
Type
Total images
Labeled images
Last updated
Status
Flowers
ICN1857427532790366208
Single-Label Classification
1,000
1,000
Mar 10, 2021, 1:45:01 PM
Success: Importing images

Google Cloud Platform ML Kit Codelab Search products and resources

Vision

Dashboard

Datasets

Models

Flowers LABEL STATS EXPORT DATA

IMPORT IMAGES TRAIN EVALUATE TEST & USE Single-Label Classification

All images 1,000

Labeled 1,000

Unlabeled 0

Filter labels

daisy 200

dandelion 200

roses 200

sunflowers 200

tulips 200

ADD NEW LABEL

Filter Filter images

Select all


tulips(1)


daisy(1)


dandelion(1)


roses(1)


daisy(1)


dandelion(1)


sunflowers(1)


dandelion(1)


tulips(1)


tulips(1)


daisy(1)


daisy(1)


tulips(1)


roses(1)


dandelion(1)


sunflowers(1)


sunflowers(1)


sunflowers(1)


daisy(1)


daisy(1)

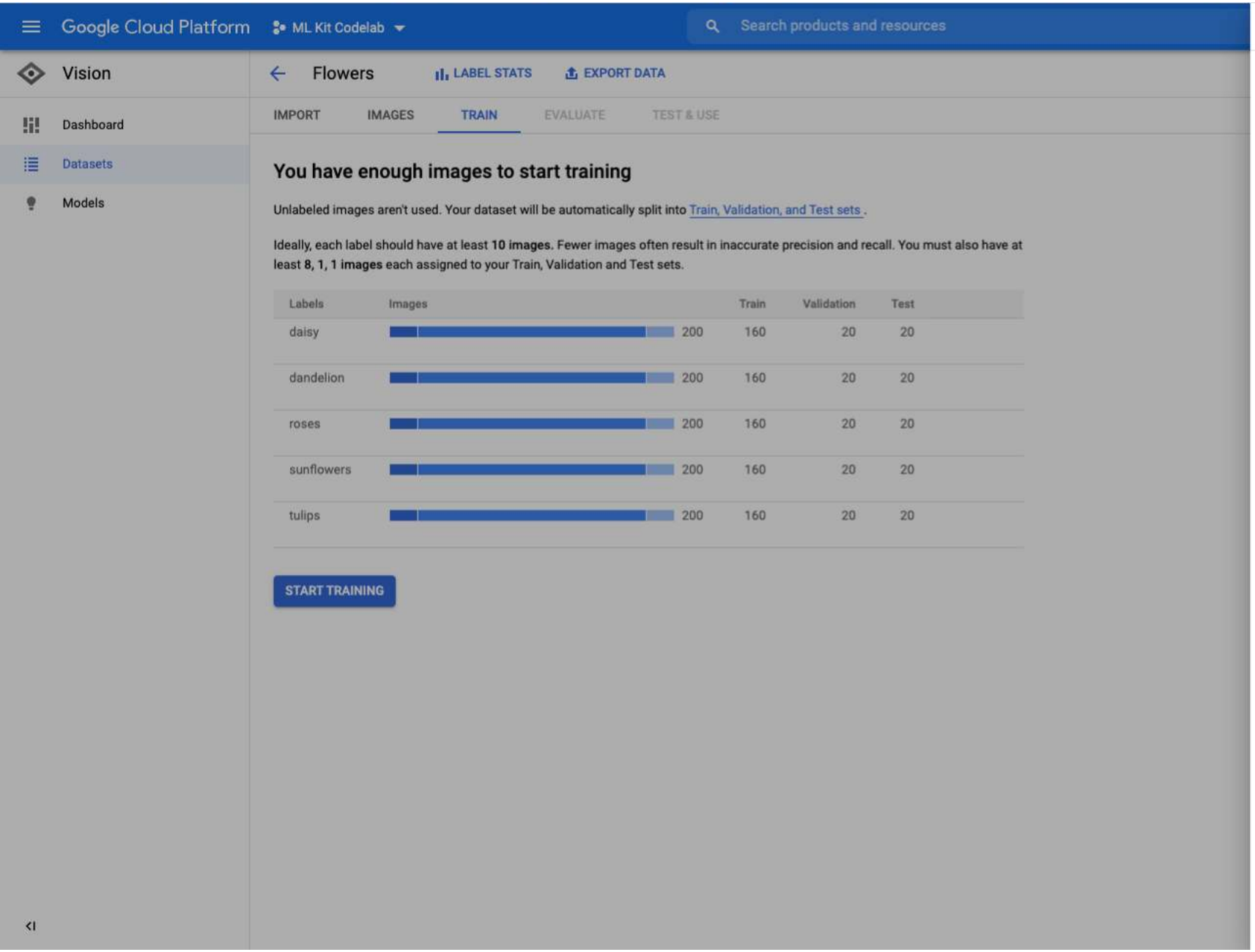

sunflowers(1)

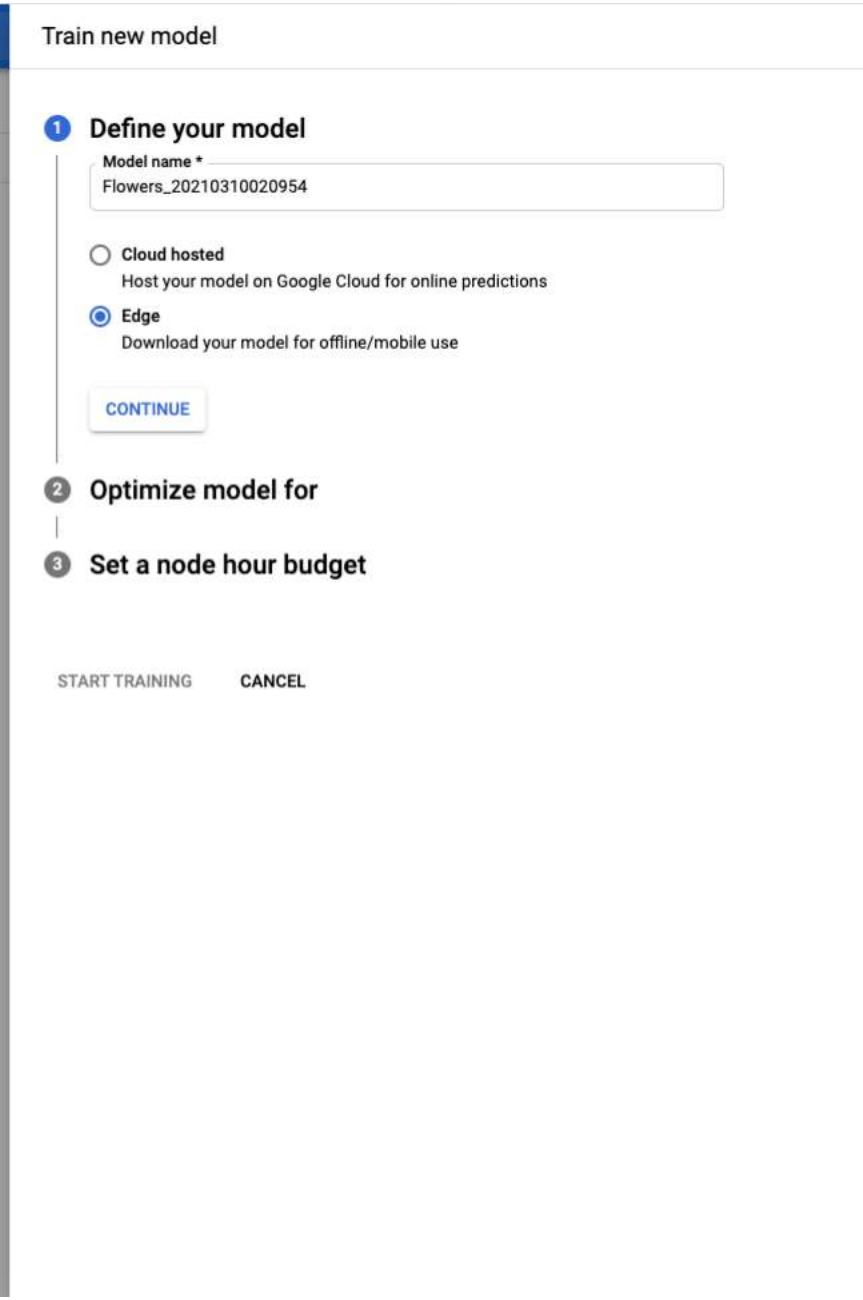

roses(1)

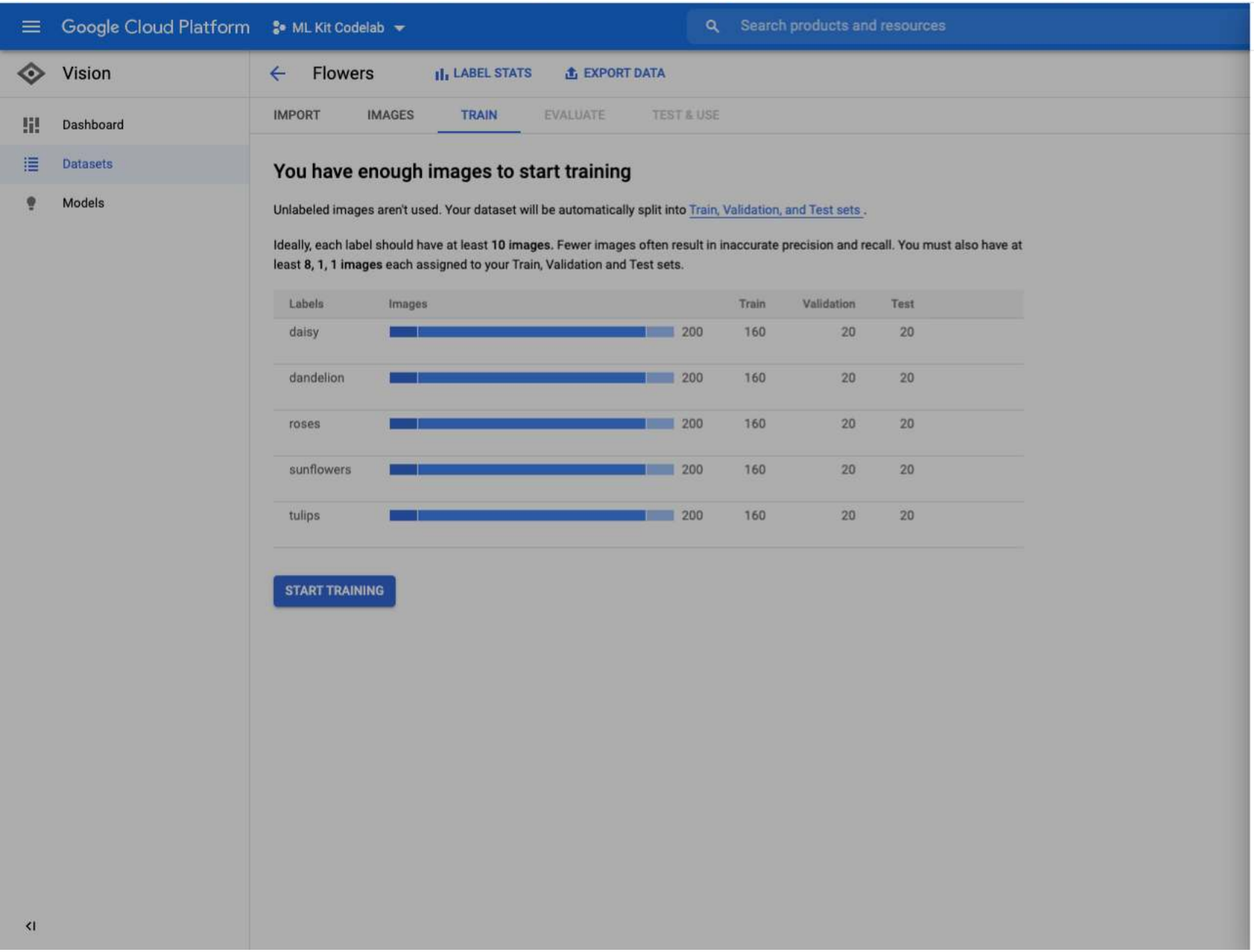

daisy(1)


daisy(1)

Images per page: 50 1 – 50 of many


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Dashboard
Datasets
Models
Flowers
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 200 160 20 20
dandelion 200 160 20 20
roses 200 160 20 20
sunflowers 200 160 20 20
tulips 200 160 20 20
START TRAINING


Train new model
1 Define your model
Model name *
Flowers_20210310020954
Cloud hosted
Host your model on Google Cloud for online predictions
Edge
Download your model for offline/mobile use
CONTINUE
2 Optimize model for
3 Set a node hour budget
START TRAINING
CANCEL



Train new model

Define your model

2 Optimize model for

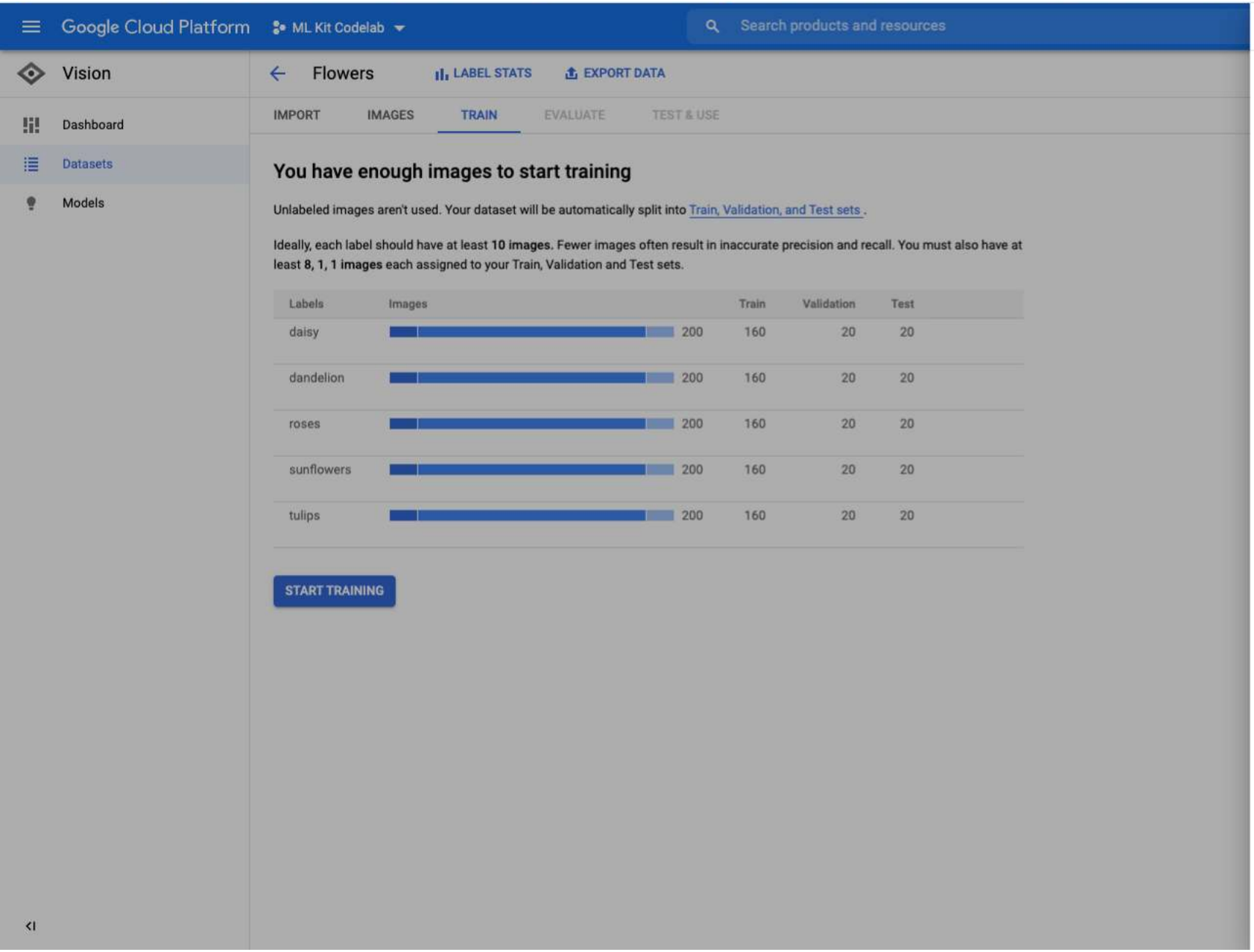
	Goal	Package size	Accuracy	Latency for Google Pixel 2
○	Higher accuracy	6 MB	Higher	360 ms
○	Best trade-off	3.2 MB	Medium	150 ms
◉	Faster predictions	0.6 MB	Lower	56 ms

Please note that prediction latency estimates are for guidance only. Actual latency will depend on your network connectivity.

CONTINUE

3 Set a node hour budget

START TRAINING CANCEL



Train new model

Define your model

Optimize model for

3 Set a node hour budget

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

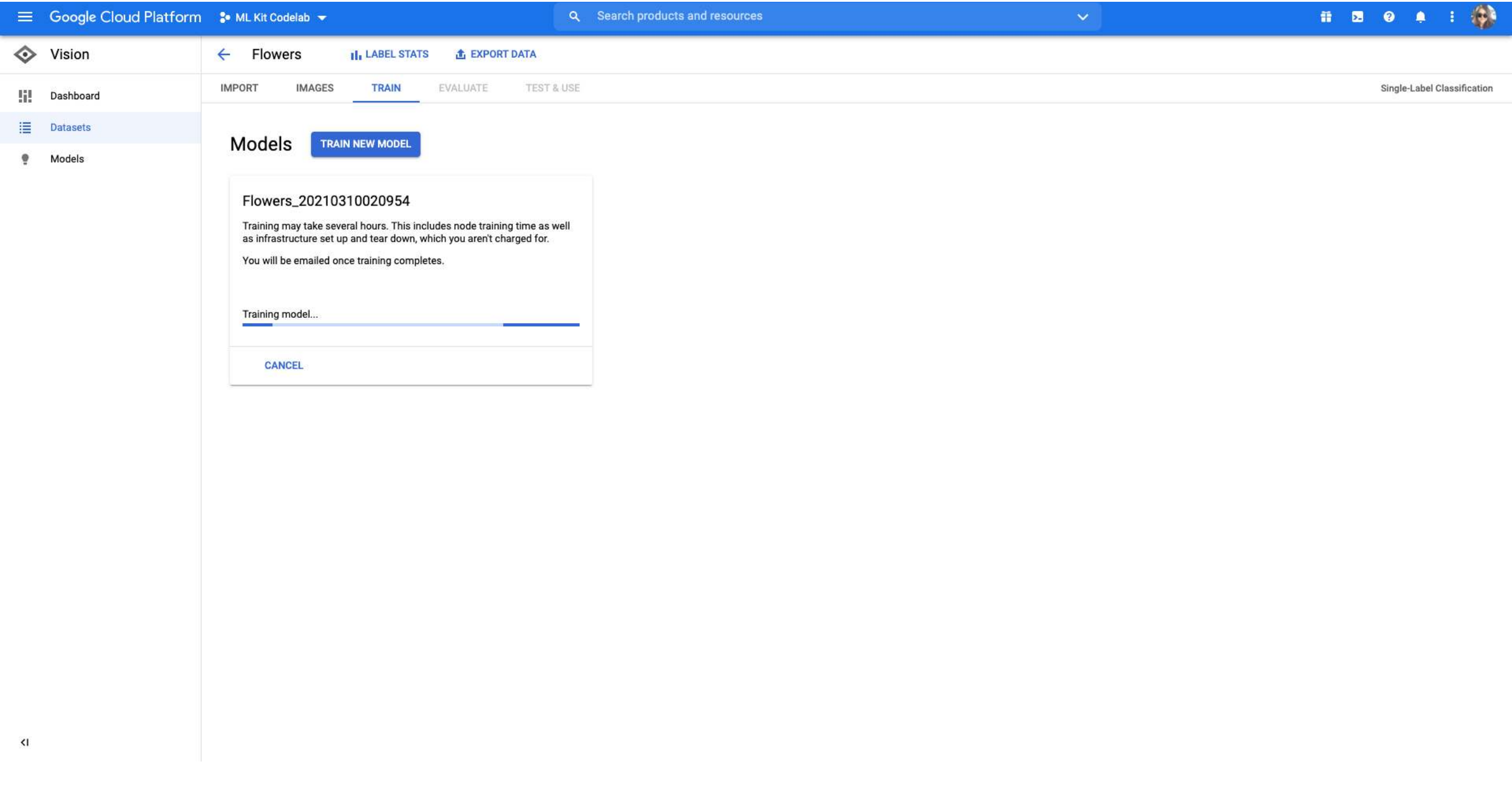
We recommend using 3 node hours for your dataset. However, you can train for as little as 1 node hours. You may also eligible to train with free node hours. Pricing guide

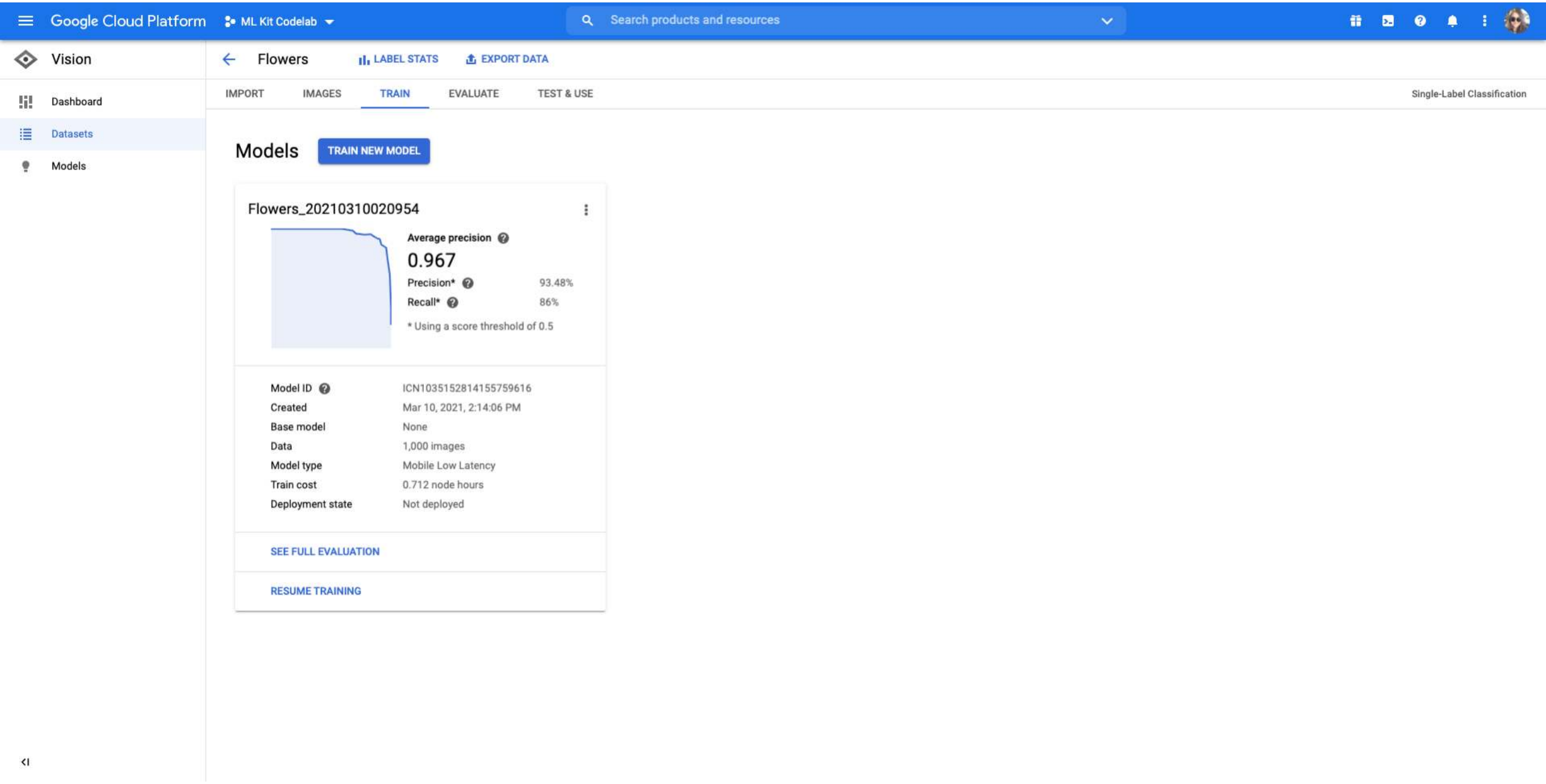
Set your budget *

1 node hours

Estimated completion date: Mar 10, 2021 4 PM
GMT-5

START TRAINING CANCEL


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Dashboard
Datasets
Models
Flowers
LABEL STATS
EXPORT DATA
IMPORT
IMAGES
TRAIN
EVALUATE
TEST & USE
Single-Label Classification
Models
TRAIN NEW MODEL
Flowers_20210310020954
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


Google Cloud Platform
ML Kit Codelab
Search products and resources
Vision
Dashboard
Datasets
Models
Flowers
LABEL STATS
EXPORT DATA
IMPORT
IMAGES
TRAIN
EVALUATE
TEST & USE
Single-Label Classification
Models
TRAIN NEW MODEL
Flowers_20210310020954
Average precision
0.967
Precision*
93.48%
Recall*
86%
* Using a score threshold of 0.5
Model ID
ICN1035152814155759616
Created
Mar 10, 2021, 2:14:06 PM
Base model
None
Data
1,000 images
Model type
Mobile Low Latency
Train cost
0.712 node hours
Deployment state
Not deployed
SEE FULL EVALUATION
RESUME TRAINING

Flowers

LABEL STATS | EXPORT DATA

IMPORT | IMAGES | TRAIN | EVALUATE | TEST & USE

Single-Label Classification

Model: Flowers_20210310020954

Confidence threshold: 0.5

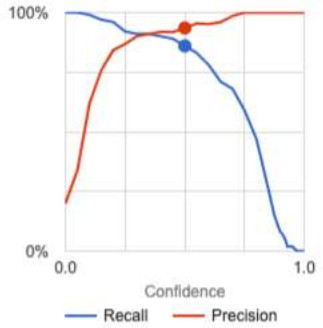
Filter labels

All labels	0.96742
daisy	0.98021
dandelion	0.99511
roses	0.93840
sunflowers	0.96226
tulips	0.94114

All labels

Total images	900
Test items	100
Precision	93.48%
Recall	86%

Use the slider to see which confidence threshold works best for your model on the precision-recall tradeoff curve.
Learn more about these metrics and graphs.

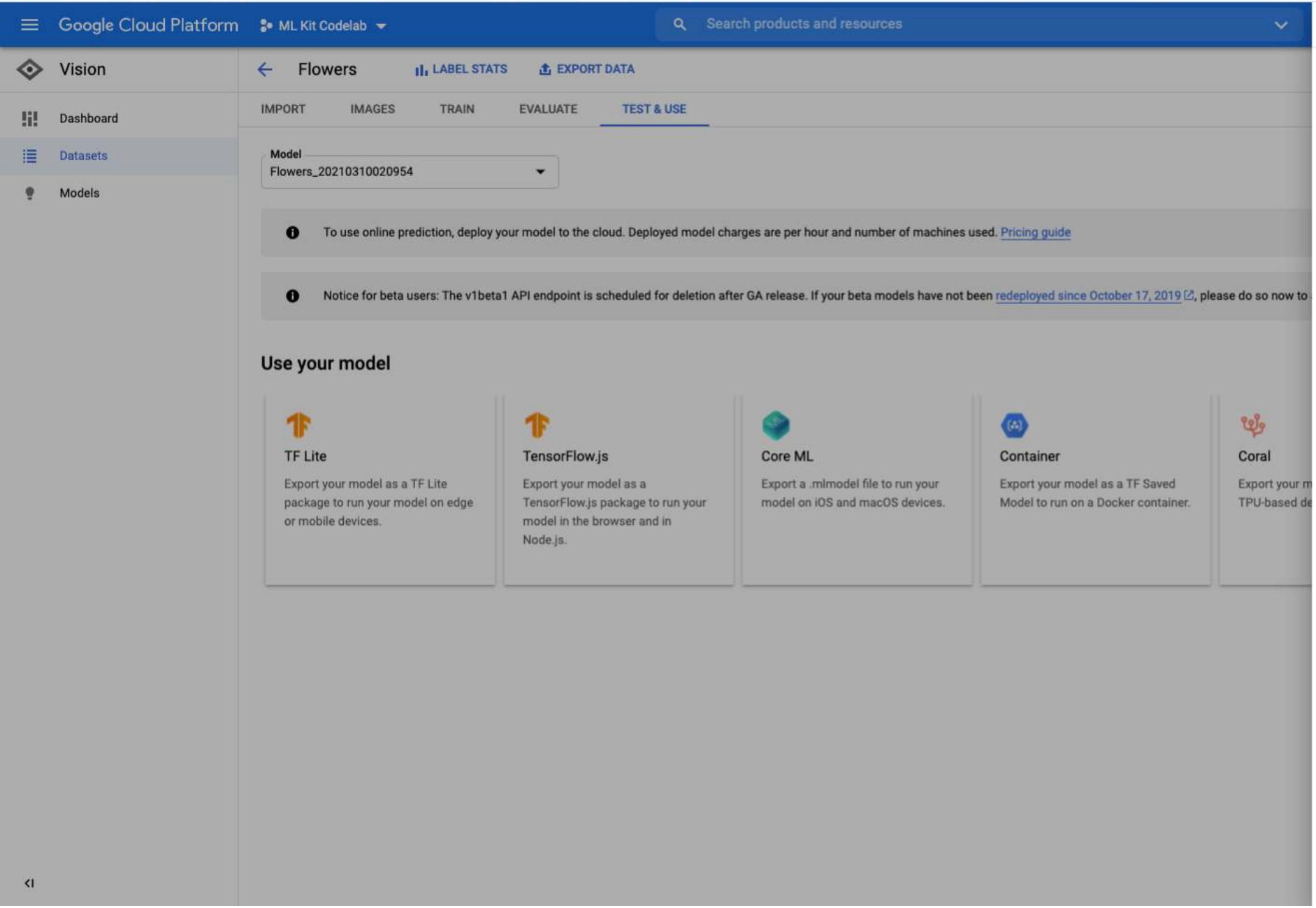


Confusion matrix

Item counts

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray). **Note that this table is limited to the 10 most confused labels.** You can download the entire confusion matrix as a CSV file.

True Label \ Predicted Label	sunflowers	dandelion	tulips	roses	daisy
sunflowers	95%	-	-	5%	-
dandelion	5%	95%	-	-	-
tulips	5%	-	85%	5%	5%
roses	5%	-	10%	85%	-
daisy	5%	5%	-	-	90%



Export TF Lite package

The **Tensorflow Lite (.tflite)** format allows you to run your model on mobile and embedded devices.

1. Export your model as a TF Lite package.

   Destination folder on Cloud Storage

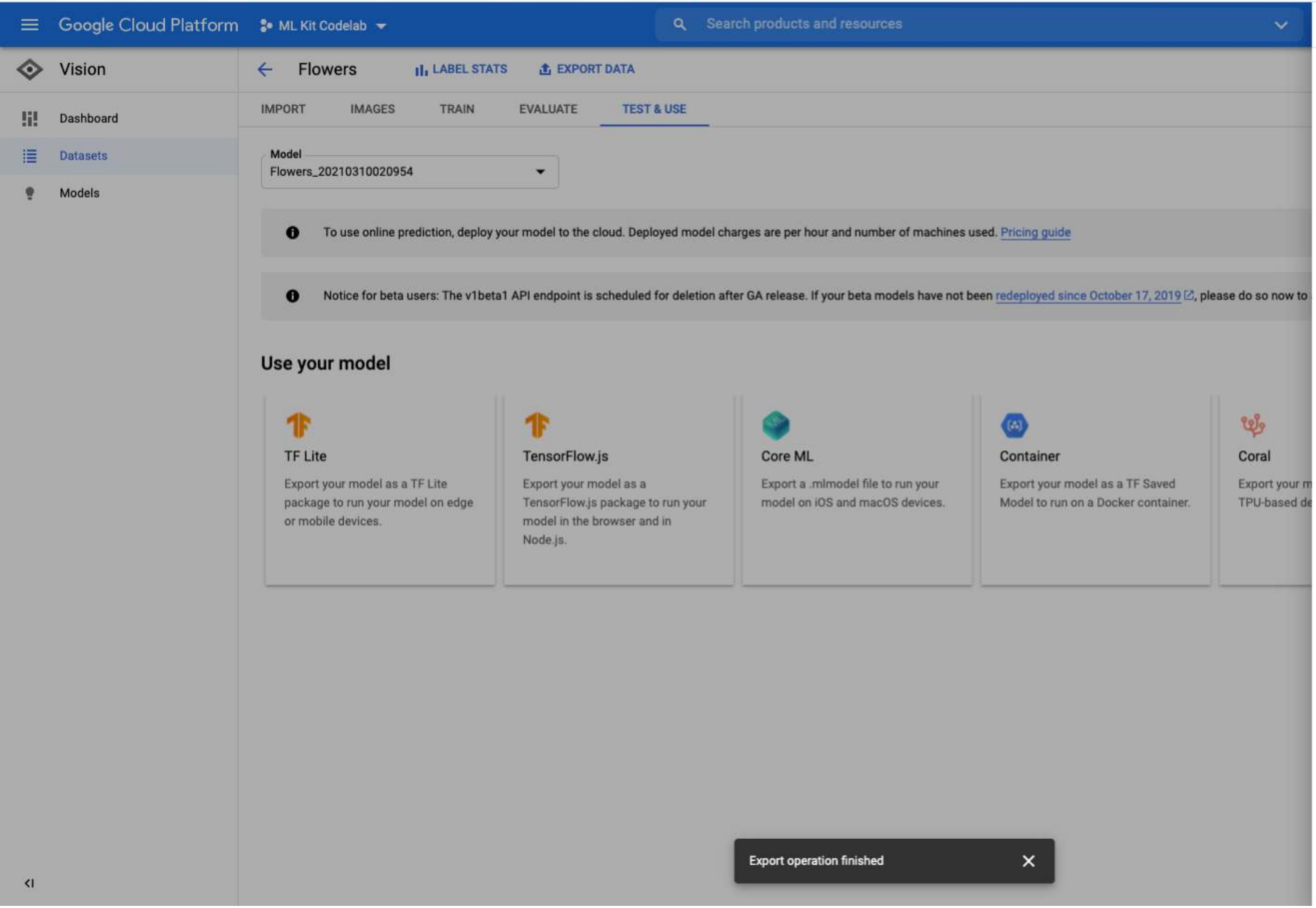
   flower-bucket-2902 **BROWSE**

   EXPORT **OPEN IN GCS**

2. After your model finishes exporting, you can copy your package to your computer using this command:

   ```
   $ gsutil cp -r gs://flower-bucket-2902 ./download_dir
   ```

3. Follow the quickstart to learn how to implement your model on your device.
   Android quickstart iOS quickstart



Export TF Lite package

The **Tensorflow Lite (.tflite)** format allows you to run your model on mobile and embedded devices.

1. Export your model as a TF Lite package.

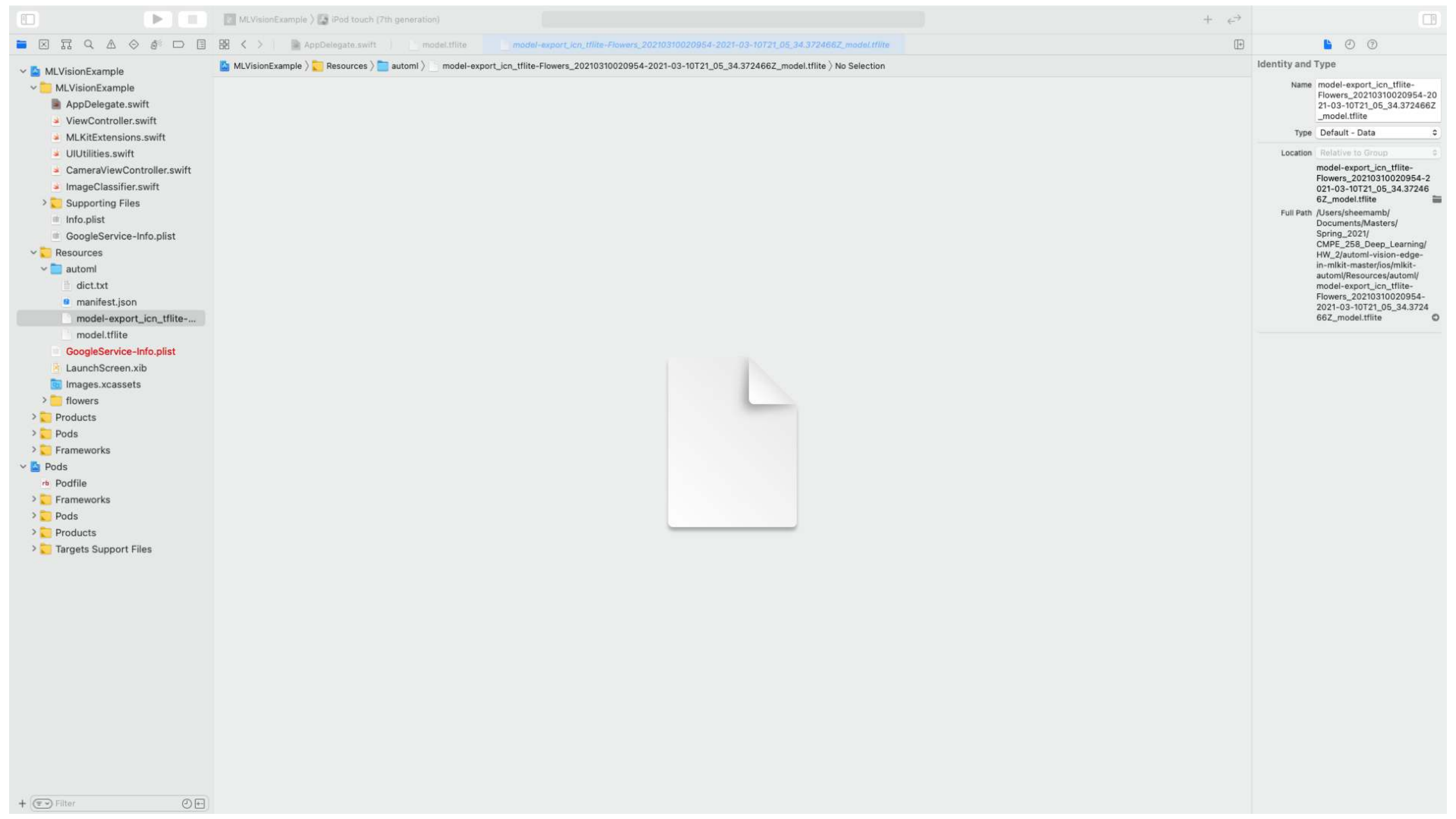
   Destination folder on Cloud Storage: flower-bucket-demo-2902 BROWSE

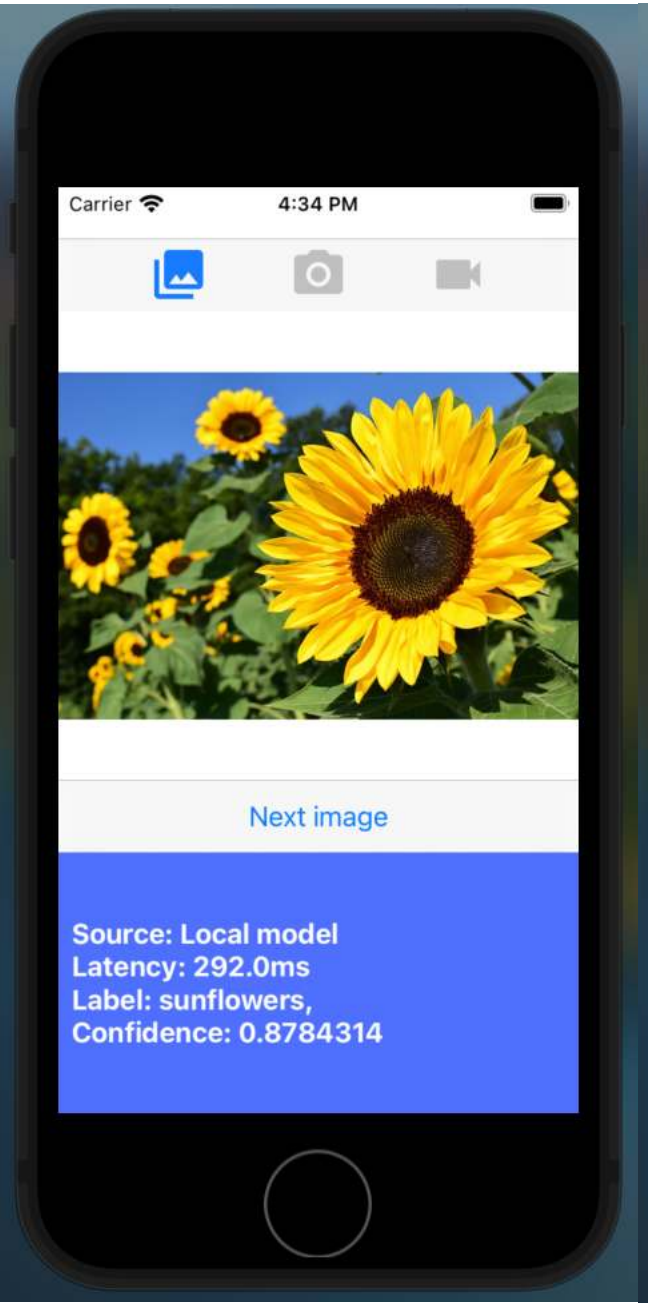
   EXPORT OPEN IN GCS

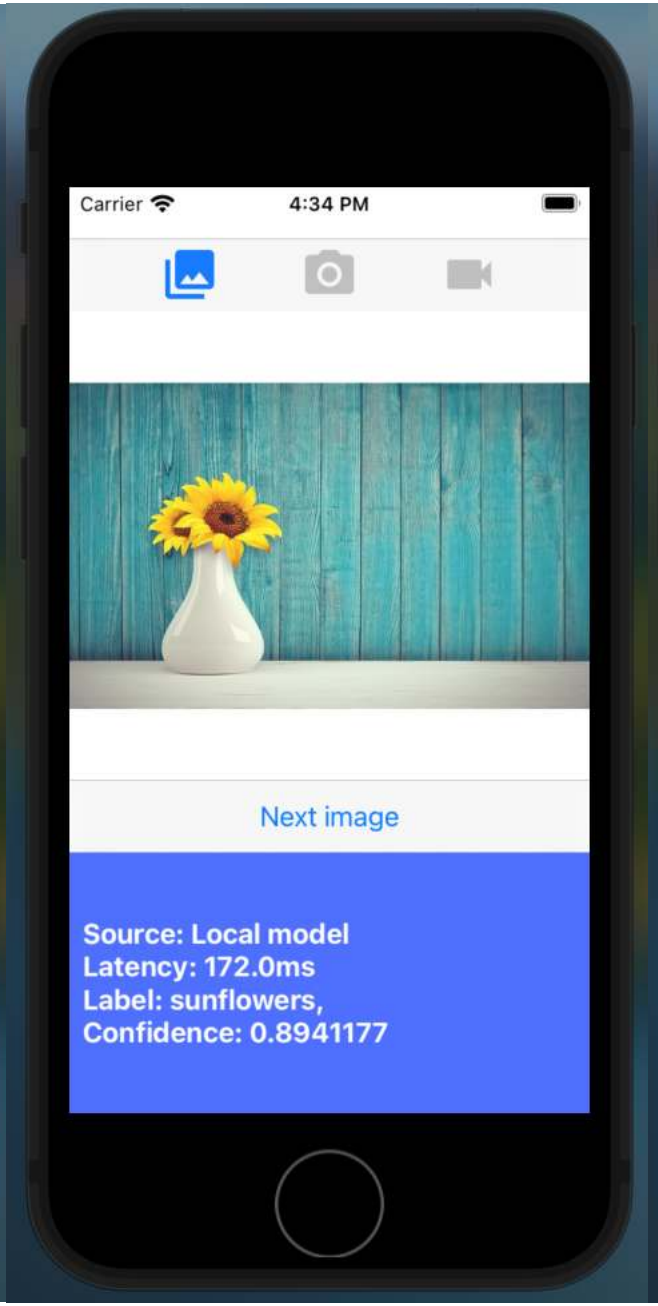
2. After your model finishes exporting, you can copy your package to your computer using this command:

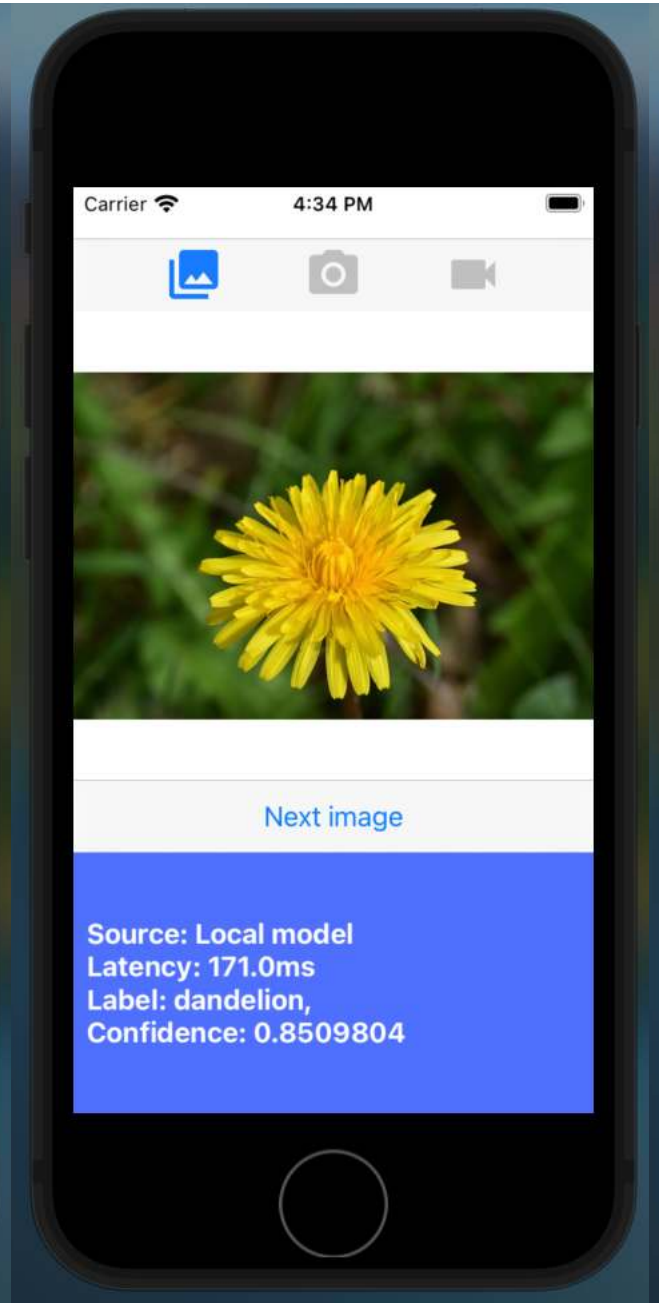
   ```
   $ gsutil cp -r gs://flower-bucket-demo-2902 ./downloa
   ```

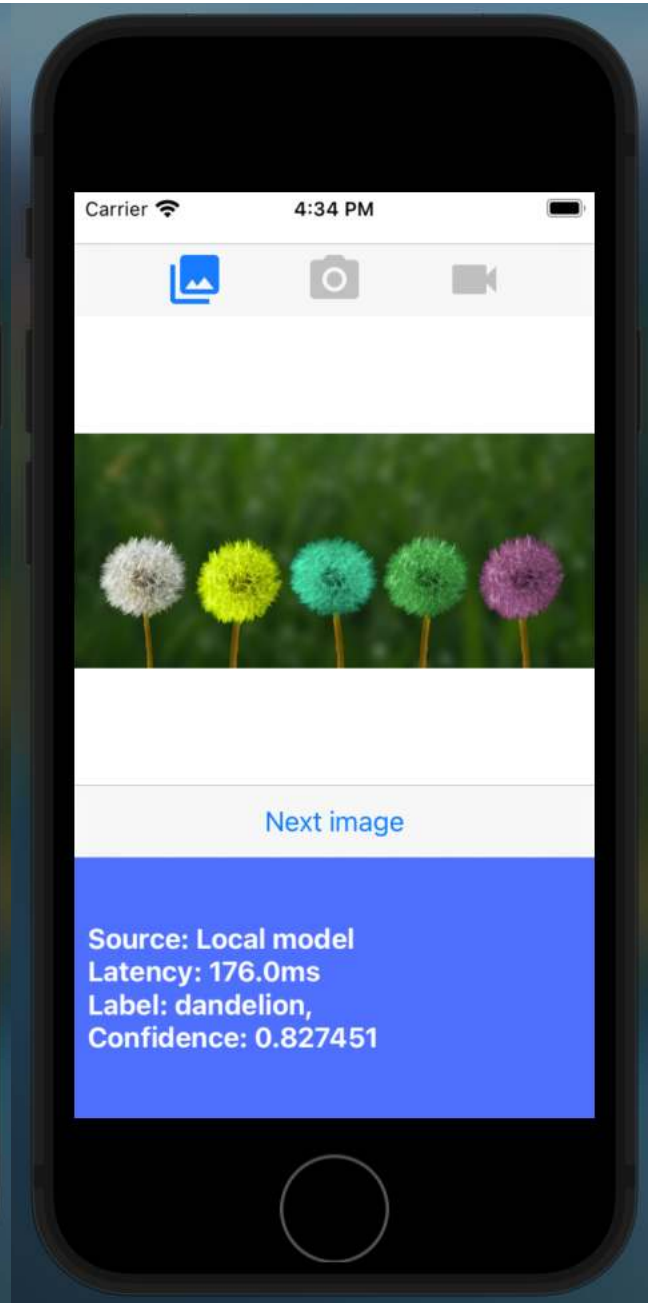
3. Follow the quickstart to learn how to implement your model on your device.
   Android quickstart iOS quickstart


MLVisionExample
iPod touch (7th generation)
AppDelegate.swift
model.tflite
model-export_icn_tflite-Flowers_20210310020954-2021-03-10T21_05_34.372466Z_model.tflite
MLVisionExample
Resources
automl
model-export_icn_tflite-Flowers_20210310020954-2021-03-10T21_05_34.372466Z_model.tflite
No Selection
MLVisionExample
MLVisionExample
AppDelegate.swift
ViewController.swift
MLKitExtensions.swift
UIUtilities.swift
CameraViewController.swift
ImageClassifier.swift
Supporting Files
Info.plist
GoogleService-Info.plist
Resources
automl
dict.txt
manifest.json
model-export_icn_tflite-...
model.tflite
GoogleService-Info.plist
LaunchScreen.xib
Images.xcassets
flowers
Products
Pods
Frameworks
Pods
Podfile
Frameworks
Pods
Products
Targets Support Files
Filter
Identity and Type
Name
model-export_icn_tflite-Flowers_20210310020954-2021-03-10T21_05_34.372466Z_model.tflite
Type
Default - Data
Location
Relative to Group
model-export_icn_tflite-Flowers_20210310020954-2021-03-10T21_05_34.372466Z_model.tflite
Full Path
/Users/sheemamb/Documents/Masters/Spring_2021/CMPE_258_Deep_Learning/HW_2/automl-vision-edge-in-mlkit-master/ios/mlkit-automl/Resources/automl/model-export_icn_tflite-Flowers_20210310020954-2021-03-10T21_05_34.372466Z_model.tflite


Carrier
4:34 PM
Next image
Source: Local model
Latency: 292.0ms
Label: sunflowers,
Confidence: 0.8784314


Carrier
4:34 PM
Next image
Source: Local model
Latency: 172.0ms
Label: sunflowers,
Confidence: 0.8941177


Carrier
4:34 PM
Next image
Source: Local model
Latency: 171.0ms
Label: dandelion,
Confidence: 0.8509804


Carrier
4:34 PM
Next image
Source: Local model
Latency: 176.0ms
Label: dandelion,
Confidence: 0.827451

THANK YOU!