

Timeseries:

```
(base) jupyter@tensorflow-2-4-20210310-160115:~$ git clone https://github.com/GoogleCloudPlatform/training-data-analyst
Cloning into 'training-data-analyst'...
remote: Enumerating objects: 74, done.
remote: Counting objects: 100% (74/74), done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 45594 (delta 48), reused 47 (delta 21), pack-reused 45520
Receiving objects: 100% (45594/45594), 479.75 MiB | 31.07 MiB/s, done.
Resolving deltas: 100% (28661/28661), done.
Checking out files: 100% (9198/9198), done.
(base) jupyter@tensorflow-2-4-20210310-160115:~$
```

Jupyter Notebook:

The screenshot shows a Jupyter Notebook interface. On the left is a file browser with a single file named "cta_ridership.csv" selected. The main area contains a code cell with the following content:

```
[10]: # Initialize plotting
register_matplotlib_converters() # Addresses a warning
sns.set(rc={'figure.figsize':(16,4)})

[11]: # Explore total rides over time
sns.lineplot(data=df, x=df.index, y=df[target]).set_title('Total Rides')
fig = plt.show()
```

Below the code cell is a line plot titled "Total Rides". The x-axis is labeled "service_date" and ranges from approximately 2001 to 2020. The y-axis is labeled "total_rides" and ranges from 0.25 to 2.00. The plot shows a highly volatile time series with a general upward trend over the years.

Query:

Google Cloud Platform time-series Search products and resources

Explorer + ADD DATA EDITOR 2 CLEVER... DEMO * UNSAVE... 3 COMPOSE NEW QUERY

Type to search

Viewing pinned projects.

clever-grid-307300 demo cta_ridership cta_ridership_model

```
CREATE OR REPLACE MODEL `demo.cta_ridership_model` OPTIONS(MODEL_TYPE='ARIMA',
    TIME_SERIES_TIMESTAMP_COL='service_date',
    TIME_SERIES_DATA_COL='total_rides',
    HOLIDAY_REGION='us') AS
SELECT
    service_date, total_rides
FROM
    `demo.cta_ridership`
```

Query results

Query complete (1 min 9 sec elapsed, 4.4 MB (ML) processed)

Job information Results Execution details

This statement will create a new model named clever-grid-307300: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

Result for query

```
SELECT *
FROM ML.EVALUATE(MODEL `demo.cta_ridership_model`)
```

Google Cloud Platform time-series Search products and resources

Explorer DEMO * UNSAVE... 3 * UNSAVE... 4 COMPOSE NEW QUERY

Type to search

Viewing pinned projects.

demo cta_ridership cta_ridership_model

```
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
2	1	1	4	false	-84345.76278035615	168703.5255607123	2.1226282591786644E9	WEEKLY
3	4	1	1	true	-84346.86918283005	168707.7383656601	2.1232853081307085E9	WEEKLY
4	1	1	3	true	-84347.97278479983	168707.94556959966	2.1239599007139666E9	WEEKLY

Rows per page: 100 1 - 42 of 42 First page < > >> Last page

JOB HISTORY QUERY HISTORY SAVED QUERIES

Result for query

```
SELECT
```

```
*
```

```
FROM ML.FORECAST(MODEL `demo.cta_ridership_model`,  
STRUCT(7 AS horizon))
```

The screenshot shows the Google Cloud Platform BigQuery interface. The top navigation bar includes 'Google Cloud Platform', 'time-series', a search bar, and user profile icons. The main area is titled 'Explorer' and shows a query editor with the following code:

```
1 SELECT  
2 *  
3 FROM  
4 ML.FORECAST(MODEL `demo.cta_ridership_model`,  
STRUCT(7 AS horizon))
```

The 'RUN' button is highlighted. Below the code, the 'Query results' section displays the following data:

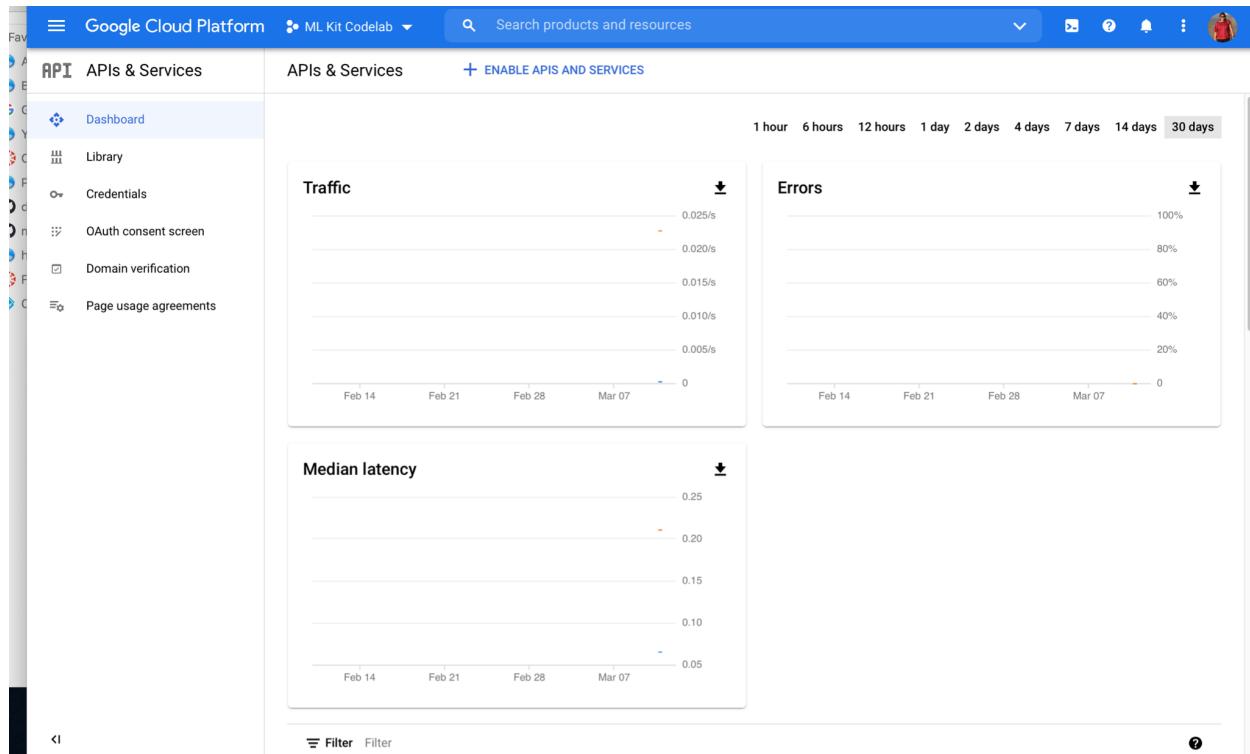
Row	forecast_timestamp	forecast_value	standard_error	confidence_level	prediction_interval_lower_bound	prediction_interval_upper_bound	confidence_interval_lower_bound
1	2020-01-01 00:00:00 UTC	662436.4424369269	46059.49014554253	0.95	572322.980240453	752549.9046334007	572322.980240453
2	2020-01-02 00:00:00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

At the bottom of the results table, there are tabs for 'JOB HISTORY', 'QUERY HISTORY', and 'SAVED QUERIES'.

IOS APP: Dashboard:

The screenshot shows the Firebase Project Overview dashboard for the 'ML Kit Codelab' project. On the left sidebar, there are several sections: Build, Authentication, Cloud Firestore, Realtime Database, Storage, Hosting, Functions, Machine Learning, Release & Monitor, Analytics, and Engage. Under Analytics, 'Blaze' is selected, indicated by a blue background. The main content area displays the 'ML Kit Codelab' dashboard. At the top, it shows '2 apps' (com.google.firebaseio and automl iOS app), '105' users in the last 30 minutes, and a 'Blaze plan' button. Below this is a 'Build' section with a 'Storage' card showing 'Storage (current)' at 61.3MB. A line chart tracks storage usage from March 5 to March 11, showing a significant jump starting on March 7. To the right of the storage card is a promotional banner for 'Store and sync app data in milliseconds' featuring a magnifying glass over a database icon.

The screenshot shows the 'Project settings' page for the 'ML Kit Codelab' project. The left sidebar is identical to the dashboard. The main content area is titled 'Project settings' and includes tabs for General, Cloud Messaging, Integrations, Service accounts, Data privacy, and Users and permissions. The 'General' tab is selected. It contains sections for 'Your project' and 'Public settings'. In 'Your project', fields include Project name (ML Kit Codelab), Project ID (ml-kit-codelab-285f1), Project number (159826202478), Default GCP resource location (Not yet selected), Parent org/folder in GCP (sjsu.edu), and Web API Key (No Web API Key for this project). In 'Public settings', fields include Public-facing name (project-159826202478) and Support email (Not configured). Below these are sections for 'Your apps' (Android apps) and 'SDK setup and configuration'. A prominent 'Add app' button is located in the bottom right corner of the app list area.



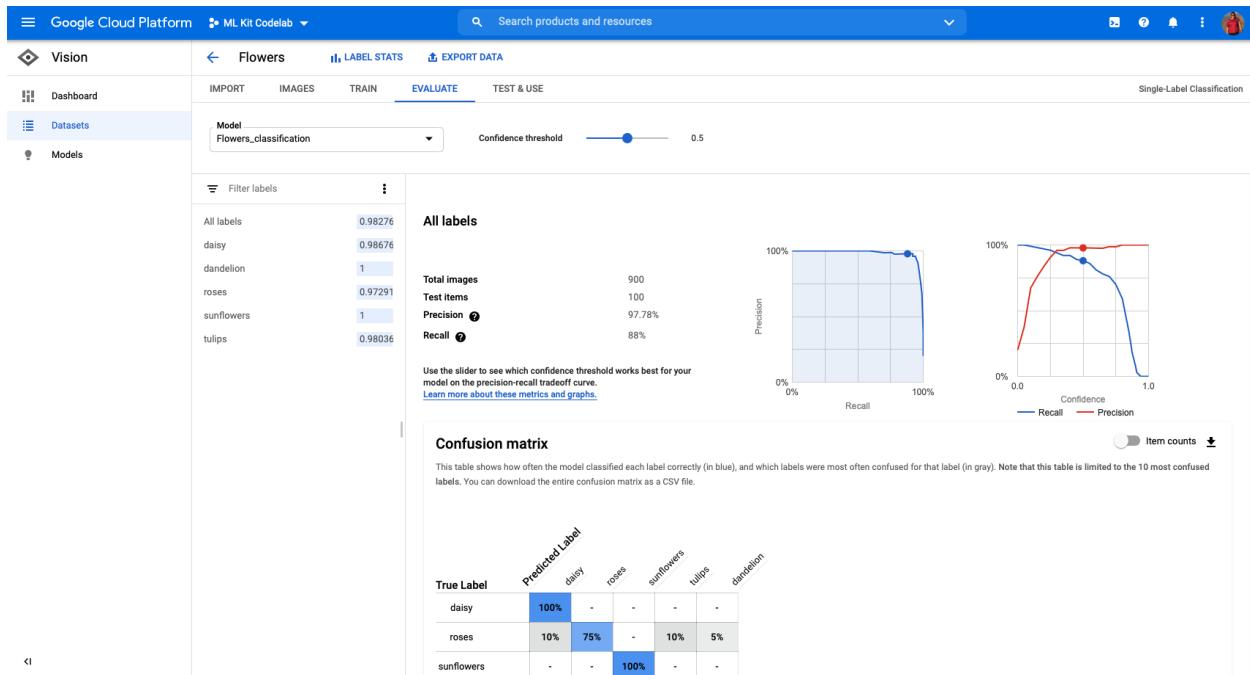
Dataset:

The screenshot shows the Google Cloud Platform Vision dataset interface for the "Flowers" dataset. The left sidebar includes links for Vision, Dashboard, Datasets, and Models. The main area has tabs for IMPORT, IMAGES (selected), TRAIN, EVALUATE, and TEST & USE. A "Single-Label Classification" note is present. The IMAGES tab shows a list of images with their labels:

Label	Count
All images	1,000
Labeled	1,000
Unlabeled	0
daisy	200
dandelion	200
roses	200
sunflowers	200
tulips	200

Below the list are five rows of image thumbnails, each labeled with its category and count. The images include various types of flowers like tulips, daisies, sunflowers, and roses.

Model:



Confusion Matrix:

This screenshot shows the confusion matrix table from the previous interface. It lists the true labels (daisy, roses, sunflowers, tulips, dandelion) and predicted labels (daisy, roses, sunflowers, tulips, dandelion). The table highlights correctly classified images in blue and misclassified images in gray. The diagonal shows 100% accuracy for all classes.

True Label	Predicted Label				
	daisy	roses	sunflowers	tulips	dandelion
daisy	100%	-	-	-	-
roses	10%	75%	-	10%	5%
sunflowers	-	-	100%	-	-
tulips	-	-	-	100%	-
dandelion	-	-	-	-	100%

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				
	daisy	roses	sunflowers	tulips	dandelion
daisy	20	-	-	-	-
roses	2	15	-	2	1
sunflowers	-	-	20	-	-
tulips	-	-	-	20	-
dandelion	-	-	-	-	20

Confusion matrix

Testing:

