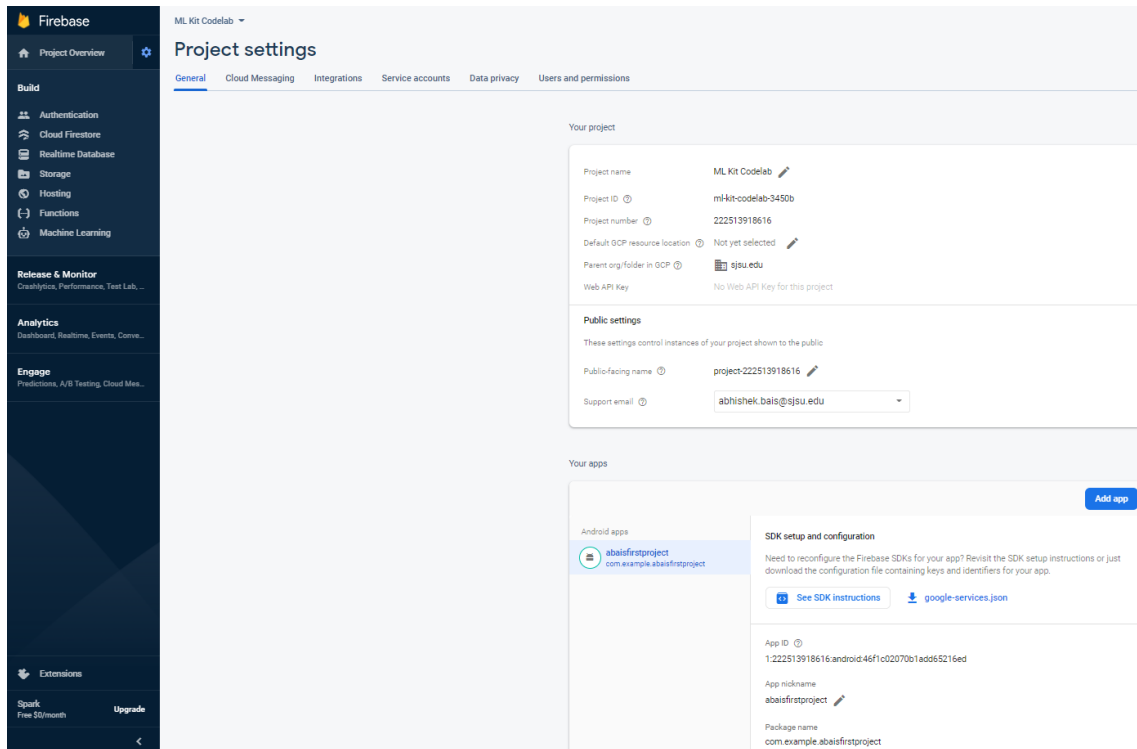
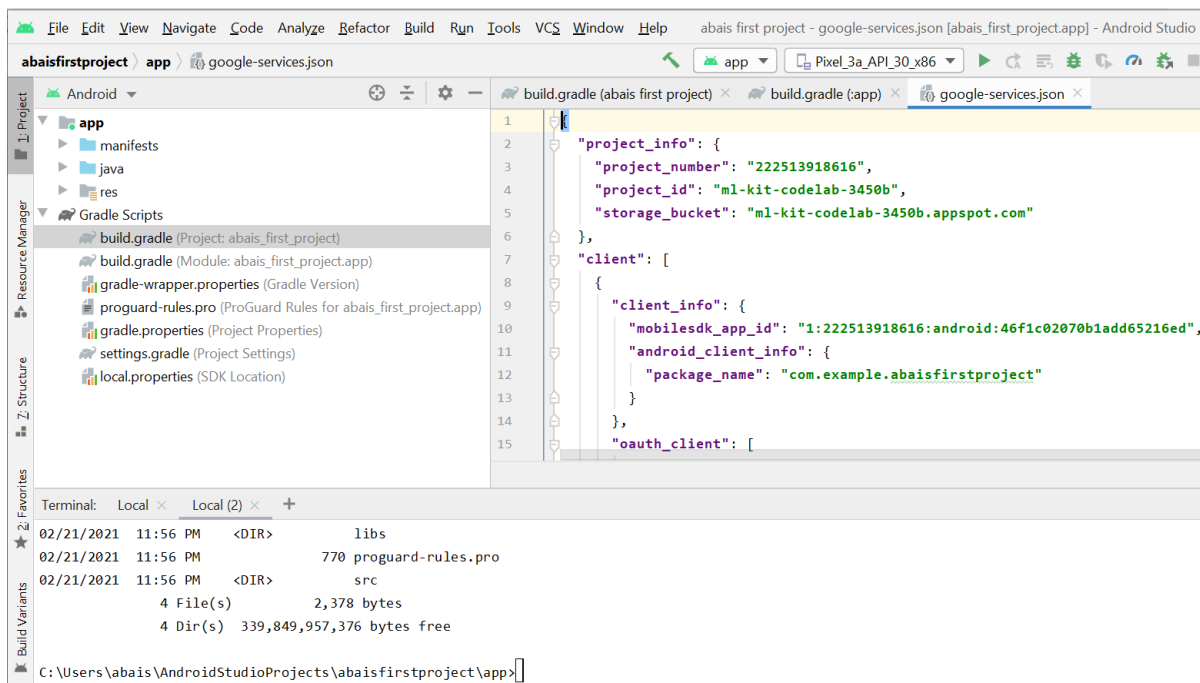


# CMPE 258 – HW2 Abhishek Bais Part 2 Execute automl vision

## a) Firebase setup



## b) Copy over the google-services JSON file



## c) From firebase console – train a model -> import `flower_photos.zip`

Google Cloud Platform My First Project Search products and resources

Vision

abaisflowers LABEL STATS EXPORT DATA

IMPORT IMAGES TRAIN EVALUATE TEST & USE

All images 1,000

Labeled 1,000

Unlabeled 0

Filter Filter images

Select all

Filter labels +

daisy 200

dandelion 200

roses 200

sunflowers 200

tulips 200

ADD NEW LABEL

tulips(1) daisy(1) dandelion(1) roses(1)

tulips(1) tulips(1) daisy(1) daisy(1)

## d) Evaluate model

abaisflowers LABEL STATS EXPORT DATA

IMPORT IMAGES TRAIN EVALUATE TEST & USE

Model abaisflowers\_20210222012303

Confidence threshold 0.5

Filter labels

All labels 0.98193

daisy 0.98569

dandelion 0.99761

roses 0.96504

sunflowers 0.99761

tulips 0.95648

All labels

Total images 900

Test items 100

Precision 97.65%

Recall 83%

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

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 common labels in the dataset.

True Label	Predicted Label	sunflowers	dandelion	tulips	roses	daisy
sunflowers		95%	-	-	5%	-
dandelion		-	95%	-	-	5%

Name	Dataset	Objective	Type	Images	Labels	Avg precision	Last updated	Deployed	Status
abaisflowers_20210222012303 ICN2337396793907609600	abaisflowers	Image classification	Mobile Best Trade-Off	1,000	5	0.982	Feb 22, 2021, 2:49:15 AM	No	Success: Training model

## e) Deploy model

Name	Dataset	Objective	Type	Images	Labels	Avg precision	Last updated	Deployed	Status
abaisflowers_20210222012303 ICN2337396793907609600	abaisflowers	Image classification	Mobile Best Trade-Off	1,000	5	0.982	Feb 22, 2021, 9:15:36 AM	Yes 1 node	Success: Deploying model

## f) Test model

[←](#) abaisflowers [LABEL STATS](#) [EXPORT DATA](#)

IMPORT IMAGES TRAIN EVALUATE **TEST & USE**

Your model is deployed and is available for online prediction requests. [Learn more](#)


Notice for beta users: The v1beta1 API endpoint is scheduled for deletion after GA release. If your beta models have not been [re](#)

### Test your model

UPLOAD IMAGES

Up to 10 images can be uploaded at a time

Test your model



Item 1 of many

Filter labels

sunflowers	1.000
dandelion	0.000
tulips	0.000
roses	0.000
daisy	0.000

Predictions

1 object

sunflowers 0.71464956 0.71

## g) Add model to app, select image and check label and confidence

abaisfirstproject - build.gradle [abais\_first\_project.app] - Android Studio

abaisfirstproject app build.gradle

dependencies {

implementation fileTree(dir: 'libs', include: ['\*.jar'])

implementation 'androidx.appcompat:appcompat:1.0.0'

implementation 'androidx.constraintlayout:constraintlayout:1.0.0'

implementation 'com.google.android.material:material:1.0.0'

implementation 'androidx.annotation:annotation:1.0.0'

implementation 'androidx.legacy:legacy-support-v13:1.0.0'

}

Build: finished at 2/25/2021 10:36 PM with 2 warnings

The 'kotlin-android-extensions' Gradle plugin is deprecated. Please use this migrat

The specified Android SDK Build Tools version (28.0.3) is ignored, as it is below th

Task :app:compressDebugAssets

Task :app:packageDebug

Task :app:assembleDebug

BUILD SUCCESSFUL in 3s

29 actionable tasks: 3 executed, 26 up-to-date

Build Analyzer results available

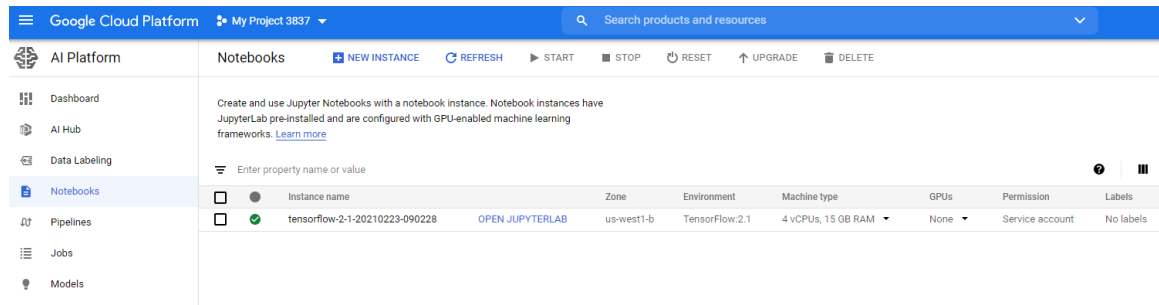
Source: Local model

Latency: 574ms

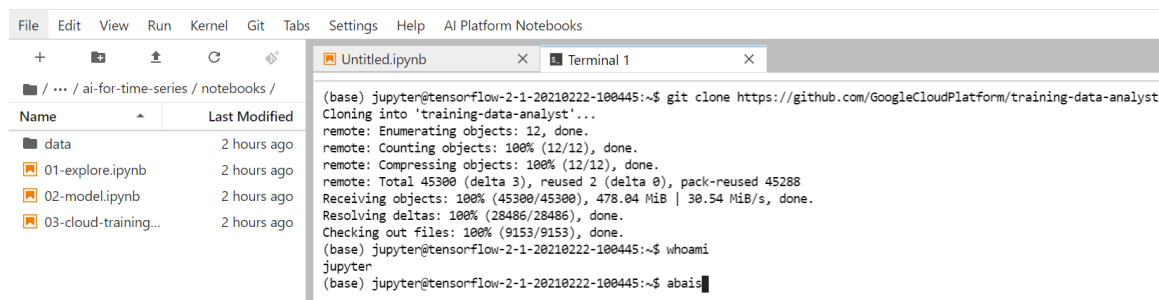
Label: roses, Confidence: 0.84

## Part 2 Time Series forecasting

### a) Create a notebook instance



### b) Download lab materials



### c) Explore and visualize data 01-explore.ipynb file screenshots

#### Create a project, load the data

```
In [10]: # Enter your project and region. Then run the cell to make sure the
# Cloud SDK uses the right project for all the commands in this notebook.

PROJECT = 'My First Project' # REPLACE WITH YOUR PROJECT NAME
REGION = 'us-central-1' # REPLACE WITH YOUR REGION e.g. us-central1

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

```
In [11]: target = 'total_rides' # The variable you are predicting
target_description = 'Total Rides' # A description of the target variable
features = {'day_type': 'Day Type'} # Weekday = W, Saturday = A, Sunday/Holiday = U
ts_col = 'service_date' # The name of the column with the date field

raw_data_file = 'https://data.cityofchicago.org/api/views/6iiy-9s97/rows.csv?accessType=DOWNLOAD'
processed_file = 'cta_ridership.csv' # Which file to save the results to
```

#### Load data

```
In [12]: # Import CSV file

df = pd.read_csv(raw_data_file, index_col=[ts_col], parse_dates=[ts_col])
```

```
In [13]: # Model data prior to 2020

df = df[df.index < '2020-01-01']
```

## Explore the data

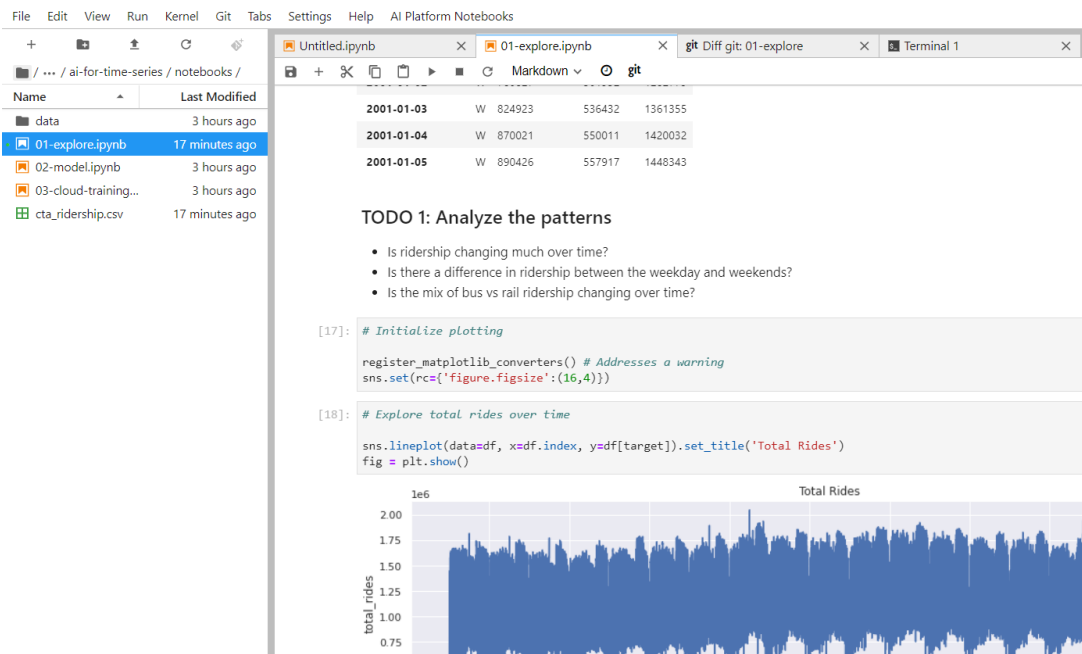
### Explore data

```
In [16]: # Print the top 5 rows
df.head()
```

```
Out[16]:
```

	day_type	bus	rail_boardings	total_rides
service_date				
2001-01-01	U	297192	126455	423647
2001-01-02	W	780827	501952	1282779
2001-01-03	W	824923	536432	1361355
2001-01-04	W	870021	550011	1420032
2001-01-05	W	890426	557917	1448343

## Analyze the patterns



# Review summary statistics

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

+

📁

📄

🔄

🔍

... / ai-for-time-series / notebooks /

Name	Last Modified
cnr_export	5 hours ago
data	12 hours ago
lstm_export	5 hours ago
trainer	8 hours ago
01-explore.ipynb	9 hours ago
02-model.ipynb	an hour ago
03-cloud-training.ipynb	a minute ago
cta_ridership.csv	9 hours ago

Untitled.ipynb X 01-explore.ipynb X 03-cloud-training.ipynb X 02-model.ipynb X

📄 + 🔍 📄 📄 📄 Code ⌵ git

2002 2004 2006 2008 2010 2012 2014 2016

service\_date

### TODO 2: Review summary statistics

- How many records are in the dataset?
- What is the average # of riders per day?

```
[21]: df[target].describe().apply(lambda x: round(x))
```

```
[21]: count    6939
      mean   1368761
      std    391443
      min    222071
      25%   1005394
      50%   1548343
      75%   1668947
      max   2049519
      Name: total_rides, dtype: int64
```

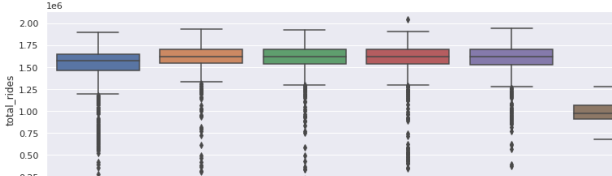
### TODO 3: Explore seasonality

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

```
[22]: # 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])
```



# Perform auto correlation

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

+

📁

📄

🔄

🔍

... / ai-for-time-series / notebooks /

Name	Last Modified
cnr_export	5 hours ago
data	12 hours ago
lstm_export	5 hours ago
trainer	8 hours ago
01-explore.ipynb	9 hours ago
02-model.ipynb	an hour ago
03-cloud-training.ipynb	2 minutes ago
cta_ridership.csv	9 hours ago

Untitled.ipynb X 01-explore.ipynb X 03-cloud-training.ipynb X 02-model.ipynb X

📄 + 🔍 📄 📄 📄 Code ⌵ git

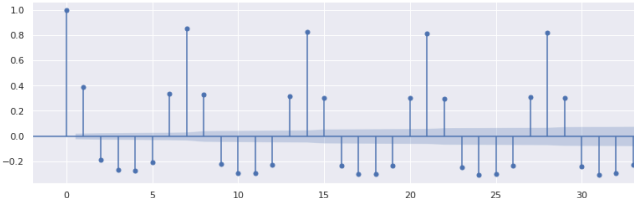
### Auto-correlation

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

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

```
[25]: 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

```
[26]: 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:

- Create a query that groups data into a time series
- Visualize data
- Decompose time series into trend and seasonal components

## d) Create a table for BigQuery with cta\_ridership.csv data

The screenshot shows the Google Cloud Platform BigQuery console. At the top, there's a blue header with 'My First Project' and a search bar. Below the header, there are tabs for 'FEATURES & INFO', 'SHORTCUT', and 'HIDE PREVIEW FEATURES'. The main interface is divided into two panels. The left panel, titled 'Explorer', shows a tree view of the project 'lucid-ceremony-305523' with a sub-entry 'abais\_demo\_bigquery' highlighted. The right panel, titled 'EDITOR', shows the details for the dataset 'lucid-ceremony-305523:abais\_demo\_bigquery'. It includes a 'Description' field (None), a 'Dataset info' section with a table of metadata, and a 'RUN' button.

Dataset ID	lucid-ceremony-305523:abais_demo_bigquery
Created	Feb 22, 2021, 1:53:42 PM
Default table expiration	Never
Last modified	Feb 22, 2021, 1:53:42 PM
Data location	US

## e) Create a time-series query and run it

The screenshot shows the Google Cloud Platform BigQuery console with a query editor and results. The query editor has a toolbar with 'RUN', 'SAVE', 'SCHEDULE', and 'MORE' buttons. The query text is as follows:

```
1 CREATE OR REPLACE MODEL
2   `abais_demo_bigquery.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   `abais_demo_bigquery.cta_ridership`
```

Below the query editor, the 'Query results' section shows the status of the query execution. It indicates that the query is complete, with 1 min 2 sec elapsed and 4.4 MB (ML) processed. There are tabs for 'Job information', 'Results', and 'Execution details'. A message at the bottom states: 'This statement will create a new model named lucid-ceremony-305523:abais\_demo\_bigquery.cta\_ridership\_model. Depending on the type of model, this may take several hours to complete.'

**f) Query model with evaluation statistics in asc order of AIC**

RUN
 SAVE ▾
 SCHEDULE
 MORE ▾

```

1 SELECT
2   *
3 FROM
4   ML.EVALUATE(MODEL `abais_demo_bigquery.cta_ridership_model`)
5

```

---

**Query results**

SAVE RESULTS
 EXPLORE DATA ▾

Query complete (0.5 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
								YFARI Y

Rows per page: 100 ▾    1 - 42 of 42

### g) Run a forecast query

RUN
 SAVE ▾
 SCHEDULE
 MORE ▾
 This query will process 23.4 KiB when r

```

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

```

---

**Query results**
 SAVE RESULTS
 EXPLORE DATA ▾

Query complete (0.3 sec elapsed, 23.4 KB processed)

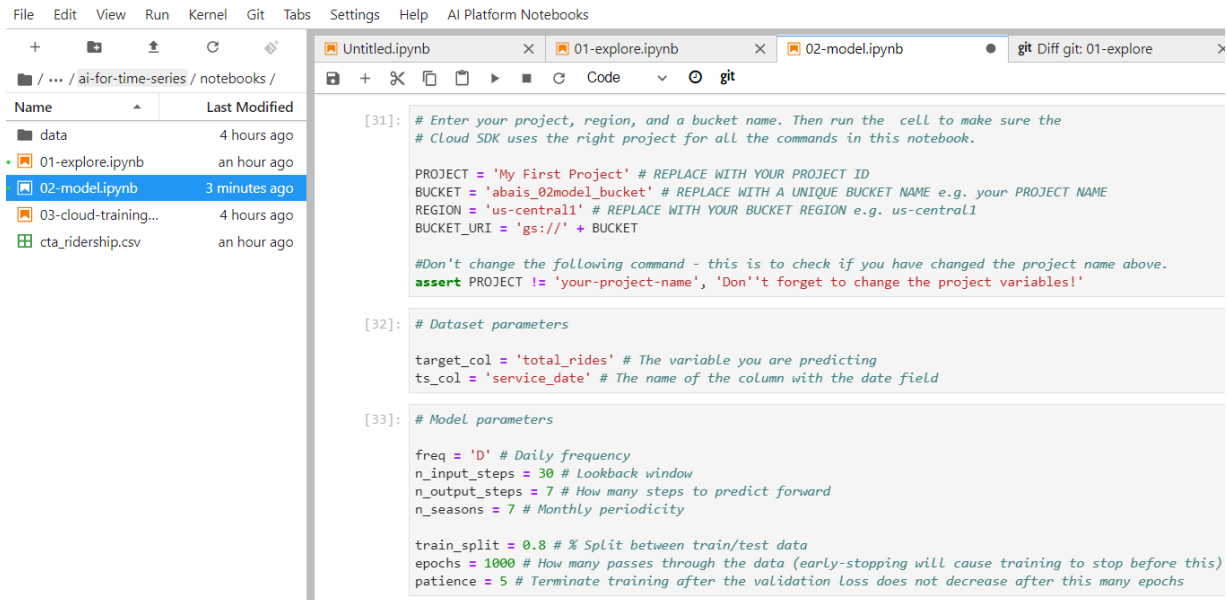
Job information    Results    JSON    Execution details

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



## h) Build a custom forecasting model 02-model.ipynb screenshots

### Setup PROJECT, BUCKET



The screenshot shows the Google Colab interface with the notebook '02-model.ipynb' open. The left sidebar shows a file explorer with 'data', '01-explore.ipynb', '02-model.ipynb' (selected), '03-cloud-training...', and 'cta\_ridership.csv'. The main area shows the notebook content with three code cells:

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

PROJECT = 'My First Project' # REPLACE WITH YOUR PROJECT ID
BUCKET = 'abais_02model_bucket' # REPLACE WITH A UNIQUE BUCKET NAME e.g. your PROJECT NAME
REGION = 'us-central1' # REPLACE WITH YOUR BUCKET REGION e.g. us-central1
BUCKET_URI = 'gs://' + BUCKET

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

```
[32]: # Dataset parameters

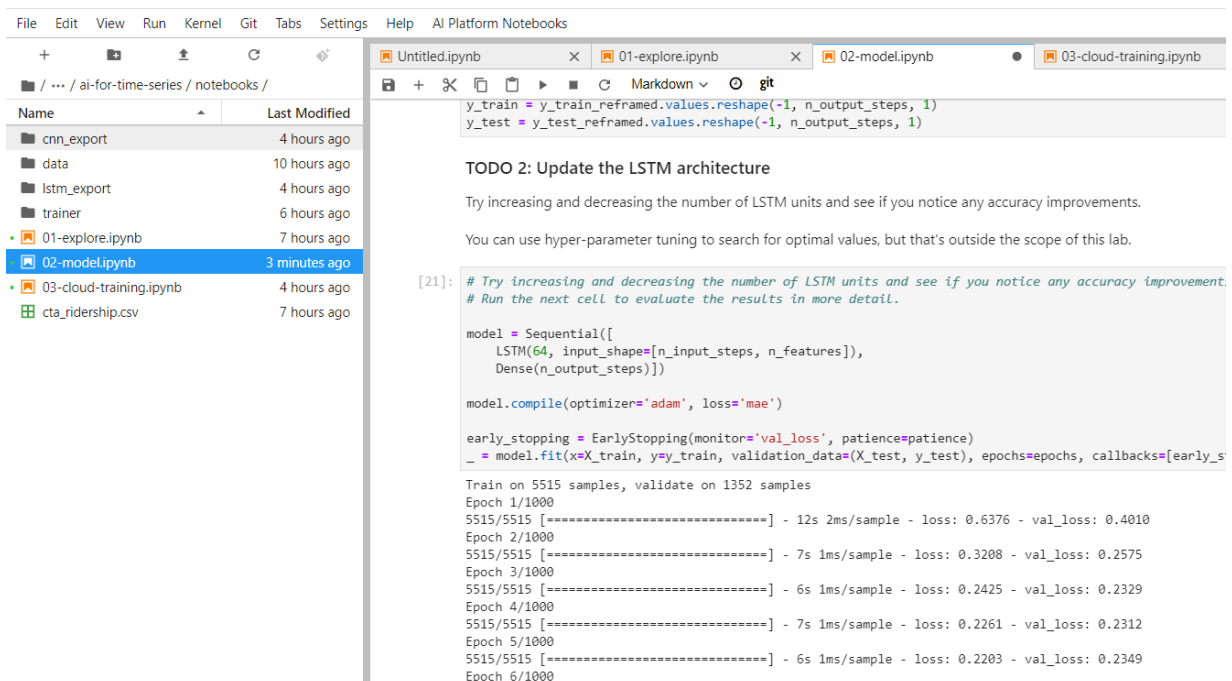
target_col = 'total_rides' # The variable you are predicting
ts_col = 'service_date' # The name of the column with the date field
```

```
[33]: # Model parameters

freq = 'D' # Daily frequency
n_input_steps = 30 # Lookback window
n_output_steps = 7 # How many steps to predict forward
n_seasons = 7 # Monthly periodicity

train_split = 0.8 # % Split between train/test data
epochs = 1000 # How many passes through the data (early-stopping will cause training to stop before this)
patience = 5 # Terminate training after the validation loss does not decrease after this many epochs
```

### Update LSTM architecture



The screenshot shows the Google Colab interface with the notebook '02-model.ipynb' open. The left sidebar shows a file explorer with 'cnn\_export', 'data', 'lstm\_export', 'trainer', '01-explore.ipynb', '02-model.ipynb' (selected), '03-cloud-training.ipynb', and 'cta\_ridership.csv'. The main area shows the notebook content with two code cells:

```
y_train = y_train_reframed.values.reshape(-1, n_output_steps, 1)
y_test = y_test_reframed.values.reshape(-1, n_output_steps, 1)
```

**TODO 2: Update the LSTM architecture**

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

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

```
[21]: # Try increasing and decreasing the number of LSTM units and see if you notice any accuracy improvement.
# 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)])

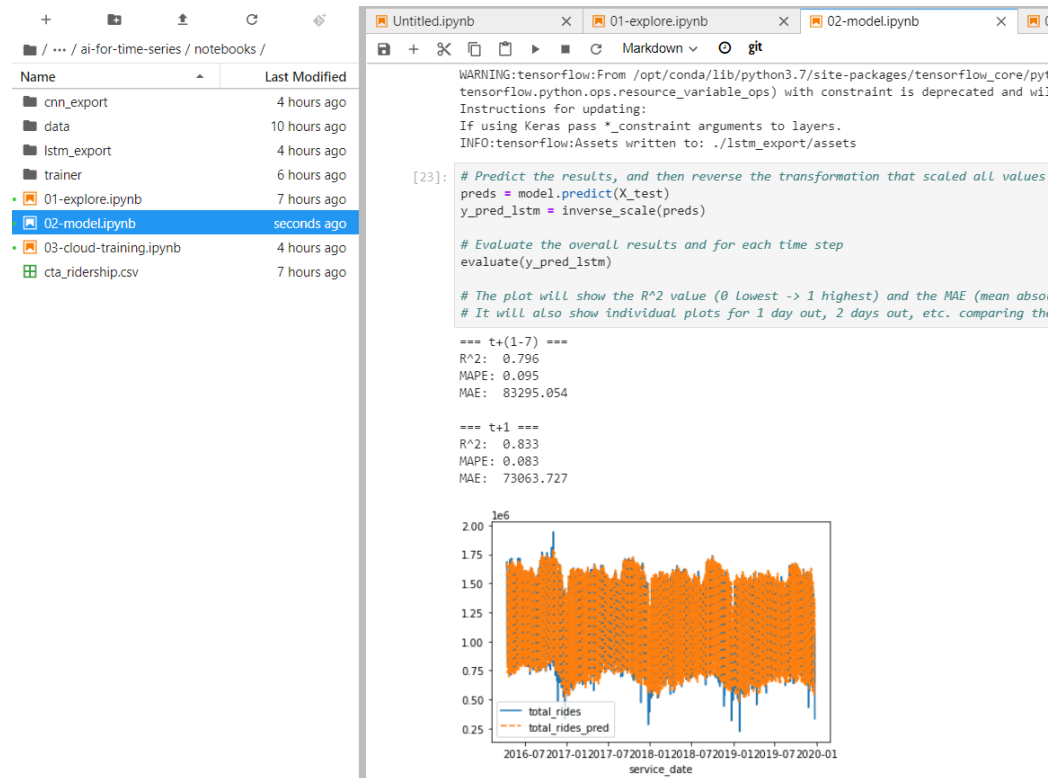
model.compile(optimizer='adam', loss='mae')

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_s
```

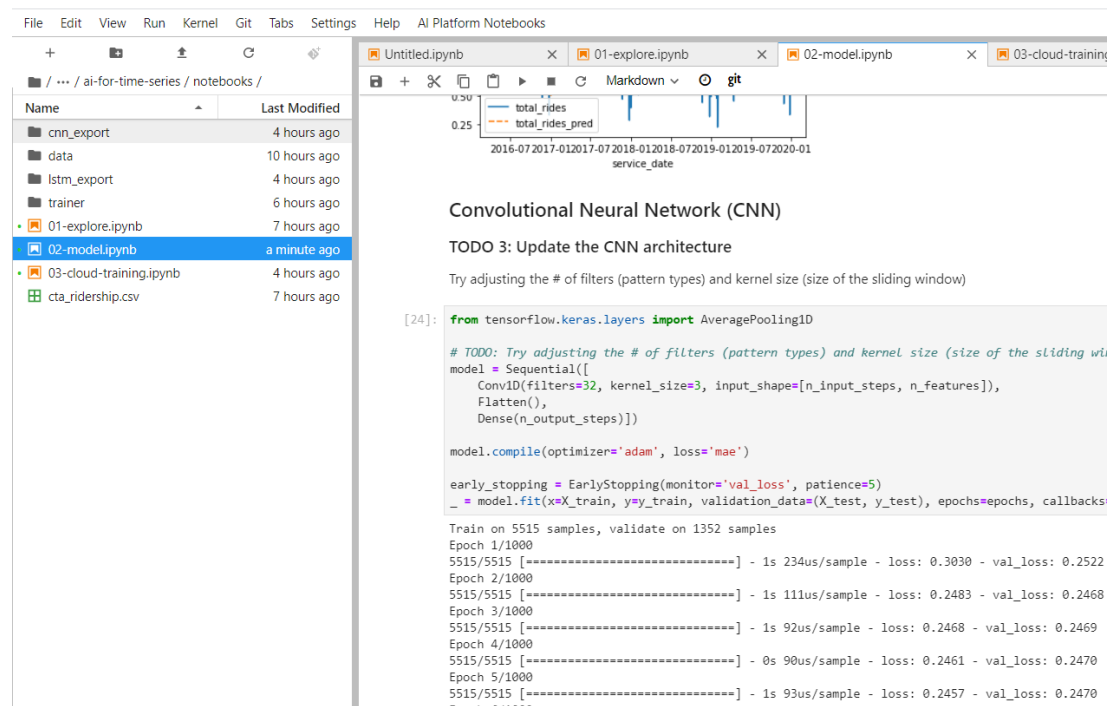
Train on 5515 samples, validate on 1352 samples

```
Epoch 1/1000
5515/5515 [=====] - 12s 2ms/sample - loss: 0.6376 - val_loss: 0.4010
Epoch 2/1000
5515/5515 [=====] - 7s 1ms/sample - loss: 0.3208 - val_loss: 0.2575
Epoch 3/1000
5515/5515 [=====] - 6s 1ms/sample - loss: 0.2425 - val_loss: 0.2329
Epoch 4/1000
5515/5515 [=====] - 7s 1ms/sample - loss: 0.2261 - val_loss: 0.2312
Epoch 5/1000
5515/5515 [=====] - 6s 1ms/sample - loss: 0.2203 - val_loss: 0.2349
Epoch 6/1000
-
```

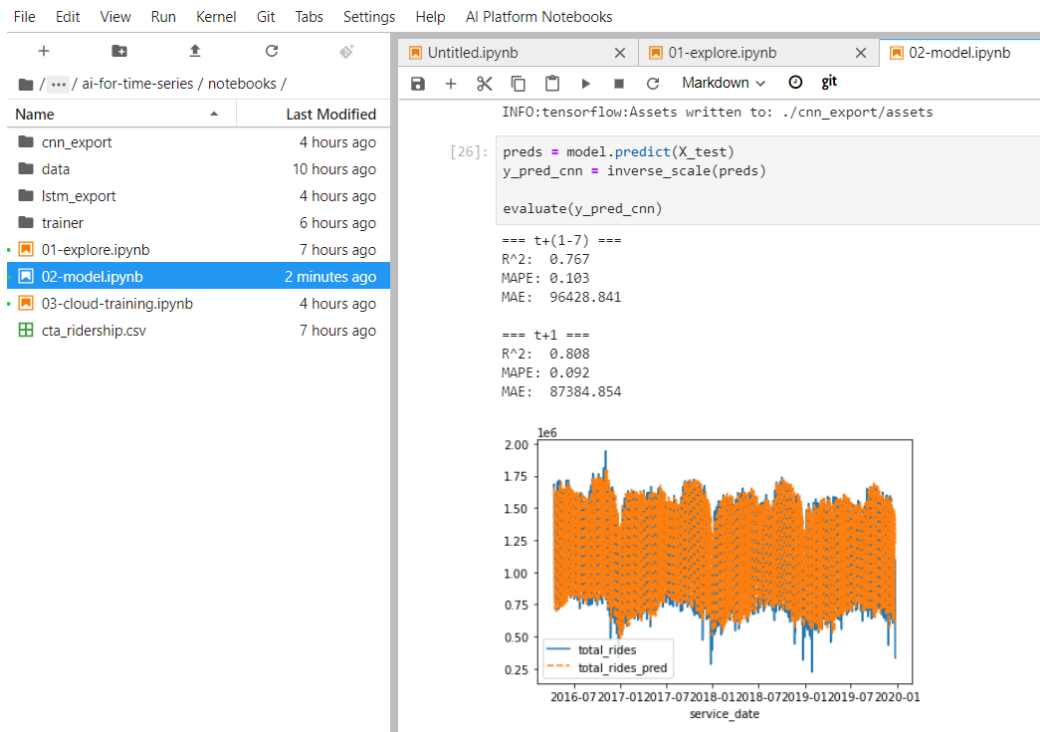
## Evaluate results for each timestamp



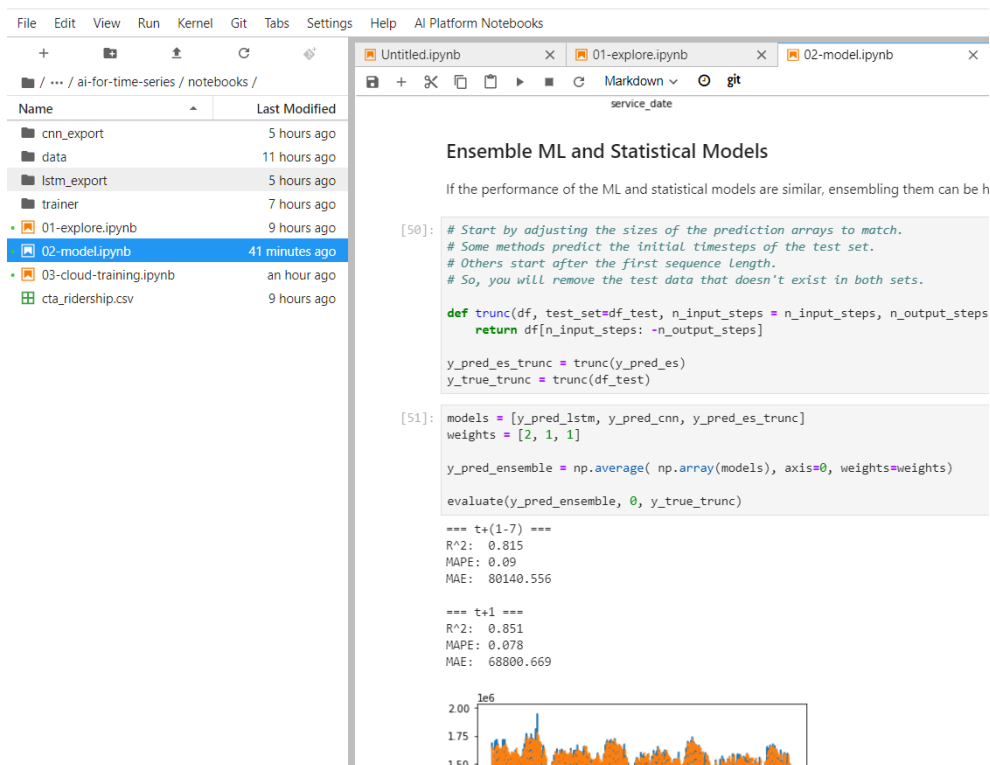
## Update CNN architecture



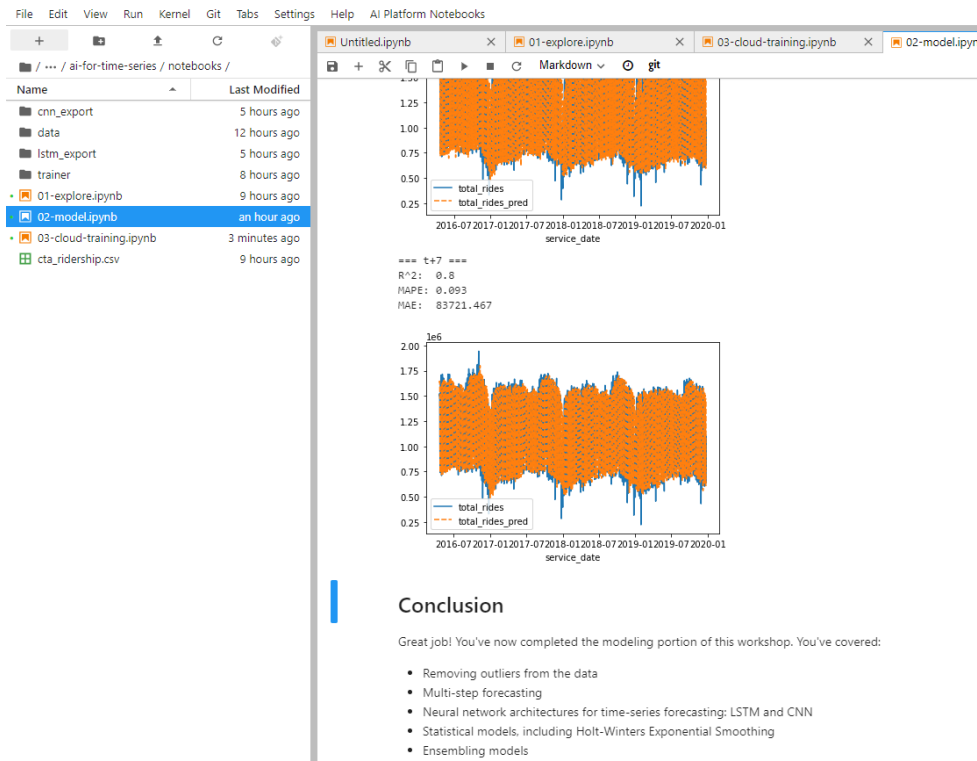
## Evaluate the CNN model



## Ensemble ML and statistical models



## Study the conclusions



## i) Train, predict in the cloud - 03-cloud-training.ipynb screenshots

### Setup PROJECT, BUCKET

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

+ ... / ai-for-time-series / notebooks /

Name Last Modified
data 13 hours ago
trainer 12 hours ago
01-explore.ipynb 13 hours ago
02-model.ipynb 13 hours ago
03-cloud-training.ipynb 7 minutes ago

Terminal 1 03-cloud-training.ipynb

from google.cloud import storage
from sklearn.preprocessing import StandardScaler

[3]: # Check the TensorFlow version installed

tf.__version__

[3]: '2.1.3'

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

PROJECT = 'My Project 3837' # REPLACE WITH YOUR PROJECT ID
BUCKET = 'abais-bucket-3837' # REPLACE WITH A UNIQUE BUCKET NAME e.g. your PROJECT NAME
REGION = 'us-central1' # REPLACE WITH YOUR BUCKET REGION e.g. us-central1
BUCKET_URI = 'gs://' + BUCKET

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

[5]: # Dataset parameters

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

[6]: # Model parameters

freq = 'D' # Daily frequency
n_input_steps = 30 # Lookback window
n_output_steps = 7 # How many steps to predict forward
n_seasons = 7 # Monthly periodicity

train_split = 0.8 # % Split between train/test data
epochs = 1000 # How many passes through the data (early-stopping will cause training to stop before this)
patience = 5 # Terminate training after the validation loss does not decrease after this many epochs

lstm_units = 64
input_layer_name = 'lstm_input'

[7]: # Training parameters
```

## Load and Preview the data

+

📁

📄

🔄

🔍

... / ai-for-time-series / notebooks /

Name	Last Modified
data	13 hours ago
trainer	12 hours ago
01-explore.ipynb	13 hours ago
02-model.ipynb	13 hours ago
03-cloud-training.ipynb	7 minutes ago

Terminal 1

03-cloud-training.ipynb

Code

git

```
bucket = storage_client.create_bucket(BUCKET)
print('Created bucket: ' + BUCKET)

Bucket exists, lets not recreate it.
```

### Load and preview the data

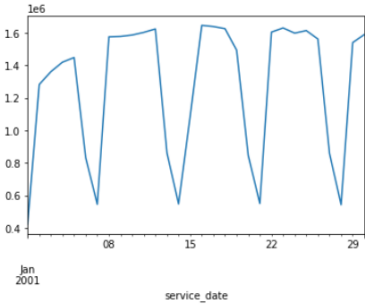
Pre-processing on the original dataset has been done for you and made available on

```
[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()
```



## Process the data

+

📁

📄

🔄

🔍

... / ai-for-time-series / notebooks /

Name	Last Modified
data	13 hours ago
trainer	12 hours ago
01-explore.ipynb	13 hours ago
02-model.ipynb	13 hours ago
03-cloud-training.ipynb	9 minutes ago

Terminal 1

03-cloud-training.ipynb

Code

git

### 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)]
df_train.head()
```

```
[12]:
```

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

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



# Prepare the model

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

+

/ ... / ai-for-time-series / notebooks /

Name	Last Modified
data	13 hours ago
trainer	13 hours ago
01-explore.ipynb	13 hours ago
02-model.ipynb	13 hours ago
03-cloud-training.ipynb	10 minutes ago

Terminal 1

03-cloud-training.ipynb

```
mkdir $TRAINER_DIR
touch $TRAINER_DIR/__init__.py

mkdir: cannot create directory 'trainer': File exists

[21]: # Copy numpy arrays to npy files

np.save(TRAINER_DIR + '/x_train.npy', X_train)
np.save(TRAINER_DIR + '/x_test.npy', X_test)
np.save(TRAINER_DIR + '/y_train.npy', y_train)
np.save(TRAINER_DIR + '/y_test.npy', y_test)

Prepare model code

[22]: # Write training code out to a file that will be submitted to the training job
# Note: f-strings are supported in Python 3.6 and above

model_template = f"""import argparse
import numpy as np
import os
import tempfile

from google.cloud import storage
from tensorflow import keras
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, LSTM
from tensorflow.keras.callbacks import EarlyStopping

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

# Export the model
export_path = os.path.join(model_dir, '{EXPORT_DIR}')
model.save(export_path)

if __name__ == '__main__':
    main()
"""

with open(os.path.join(TRAINER_DIR, 'model.py'), 'w') as f:
    f.write(model_template.format(**globals()))

[23]: # Copy the model and training data files to a GCS bucket

!gsutil -m cp -r trainer $BUCKET_URI

Copying file://trainer/__init__.py [Content-Type=text/x-python]...
Copying file://trainer/model.py [Content-Type=text/x-python]...
Copying file://trainer/y_test.npy [Content-Type=application/octet-stream]...
Copying file://trainer/x_train.npy [Content-Type=application/octet-stream]...
Copying file://trainer/x_test.npy [Content-Type=application/octet-stream]...
Copying file://trainer/y_train.npy [Content-Type=application/octet-stream]...
- [6/6 files][ 1.9 MiB/ 1.9 MiB] 100% Done
Operation completed over 6 objects/1.9 MiB.

[24]: # List the contents of the bucket to ensure they were copied properly

!gsutil ls $BUCKET_URI/$TRAINER_DIR

gs://abais-bucket-3837/trainer/__init__.py
gs://abais-bucket-3837/trainer/model.py
gs://abais-bucket-3837/trainer/x_test.npy
gs://abais-bucket-3837/trainer/x_train.npy
gs://abais-bucket-3837/trainer/y_test.npy
gs://abais-bucket-3837/trainer/y_train.npy
```

+

/ ... / ai-for-time-series / notebooks /

Name	Last Modified
data	13 hours ago
trainer	13 hours ago
01-explore.ipynb	13 hours ago
02-model.ipynb	13 hours ago
03-cloud-training.ipynb	10 minutes ago

Terminal 1

03-cloud-training.ipynb

```
early_stopping = EarlyStopping(monitor='val_loss', patience=patience)
_ = model.fit(x=X_train, y=y_train, validation_data=(X_test, y_test), epoch

# Export the model
export_path = os.path.join(model_dir, '{EXPORT_DIR}')
model.save(export_path)

if __name__ == '__main__':
    main()
"""

with open(os.path.join(TRAINER_DIR, 'model.py'), 'w') as f:
    f.write(model_template.format(**globals()))

[23]: # Copy the model and training data files to a GCS bucket

!gsutil -m cp -r trainer $BUCKET_URI

Copying file://trainer/__init__.py [Content-Type=text/x-python]...
Copying file://trainer/model.py [Content-Type=text/x-python]...
Copying file://trainer/y_test.npy [Content-Type=application/octet-stream]...
Copying file://trainer/x_train.npy [Content-Type=application/octet-stream]...
Copying file://trainer/x_test.npy [Content-Type=application/octet-stream]...
Copying file://trainer/y_train.npy [Content-Type=application/octet-stream]...
- [6/6 files][ 1.9 MiB/ 1.9 MiB] 100% Done
Operation completed over 6 objects/1.9 MiB.

[24]: # List the contents of the bucket to ensure they were copied properly

!gsutil ls $BUCKET_URI/$TRAINER_DIR

gs://abais-bucket-3837/trainer/__init__.py
gs://abais-bucket-3837/trainer/model.py
gs://abais-bucket-3837/trainer/x_test.npy
gs://abais-bucket-3837/trainer/x_train.npy
gs://abais-bucket-3837/trainer/y_test.npy
gs://abais-bucket-3837/trainer/y_train.npy
```

## Submit a training job

+

📁

📄

🔄

🔍

/ ... / ai-for-time-series / notebooks /

Name	Last Modified
data	13 hours ago
trainer	13 hours ago
01-explore.ipynb	13 hours ago
02-model.ipynb	13 hours ago
03-cloud-training.ipynb	11 minutes ago

Terminal 1

03-cloud-training.ipynb

gs://abais-bucket-3837/trainer/y\_train.npy

### Submit training job

```
[25]: # Re-run this if you need to create a new training job

timestamp = str(datetime.datetime.now().time())
JOB_NAME = 'caip_training_' + str(int(time.time()))

[26]: # Set training job parameters

MODULE_NAME = TRAINER_DIR + '.model'
TRAIN_DIR = os.getcwd() + '/' + TRAINER_DIR
JOB_DIR = BUCKET_URI

[27]: # Submit the training job

!gcloud ai-platform jobs submit training $JOB_NAME \
  --scale-tier basic \
  --package-path $TRAIN_DIR \
  --module-name $MODULE_NAME \
  --job-dir $JOB_DIR \
  --region $REGION \
  --runtime-version $RUNTIME_VERSION \
  --python-version $PYTHON_VERSION

Job [caip_training_1614143464] submitted successfully.
Your job is still active. You may view the status of your job with the command

$ gcloud ai-platform jobs describe caip_training_1614143464

or continue streaming the logs with the command

$ gcloud ai-platform jobs stream-logs caip_training_1614143464
jobId: caip_training_1614143464
state: QUEUED

[28]: # Check the job status

!gcloud ai-platform jobs describe $JOB_NAME

createTime: '2021-02-24T05:11:19Z'
etag: gozqkeBsVq==
jobId: caip_training_1614143464
state: PREPARING
trainingInput:
```

## Monitor the training job

My Project 3837

Search products and resources

🔍

🔗

🔔

🔔

🔔

🔔

🔔

🔔

Logs Explorer

OPTIONS

REFINE SCOPE

Project

SHARE LINK

LAST 1 HOUR

PAGE LAYOUT

New features are available in the Logs Explorer.

Dismiss

Learn more

Query preview

resource.type="ml\_job"

Save

Stream logs

Run query

Log fields

Search fields and values

RESOURCE TYPE

Cloud ML Job

Clear X

SEVERITY

Info

1,433

Warning

3

Error

1

LOG NAME

master-replica-0

1,427

ml.googleapis.com/caip\_training\_1614143464

10

PROJECT\_ID

my-project-3837-305716

1,437

JOB\_ID

caip\_training\_1614143464

1,437

TASK\_NAME

master-replica-0

1,427

service

10

Histogram

1000

500

0

Feb 23, 8:31 PM

9:00 PM

Feb 23, 9:32 PM

Query results

Jump to now

Actions

Configure

SEVERITY	TIMESTAMP	PST	SUMMARY
Info	2021-02-23 21:21:36.317	PST	master-replica-0 "Sets are not currently considered sequences, but this may change in the futur..."
Info	2021-02-23 21:21:36.320	PST	master-replica-0 "From /opt/conda/lib/python3.7/site-packages/tensorflow/python/training/tracki..."
Info	2021-02-23 21:21:41.410	PST	master-replica-0 "Assets written to: gs://abais-bucket-3837/tf_export/assets"
Info	2021-02-23 21:21:42.523	PST	master-replica-0 "Module completed; cleaning up."
Info	2021-02-23 21:21:42.523	PST	master-replica-0 "Clean up finished."
Info	2021-02-23 21:21:42.524	PST	master-replica-0 "Task completed successfully."
Info	2021-02-23 21:21:52.346	PST	service Tearing down training program.
Info	2021-02-23 21:22:34.156	PST	service Finished tearing down training program.
Info	2021-02-23 21:22:34.406	PST	service Job completed successfully.

textPayload: "Job completed successfully."

insertId: "645qmb9e"

resource: (2)

timestamp: "2021-02-24T05:22:34.406642736Z"

severity: "INFO"

labels: (2)

logName: "projects/my-project-3837-305716/logs/ml.googleapis.com%2Fcaip\_training\_1614143464"

receiveTimestamp: "2021-02-24T05:22:35.821905872Z"

Showing logs for last 1 hour ending at 2/23/21, 9:31 PM.

Extend time by: 1 hour

Edit time

my/home?project=my-project-3837-305716



## Create a Model Version, Deploy the Model

The screenshot shows a JupyterLab environment. On the left is a file browser pane showing the directory structure: `/ ... / ai-for-time-series / notebooks /`. It lists files: `data` (3 days ago), `trainer` (3 days ago), `01-explore.ipynb` (3 days ago), `02-model.ipynb` (3 days ago), and `03-cloud-training.ipynb` (a minute ago). On the right is a terminal window titled `Terminal 1` showing the execution of a Jupyter Notebook. The terminal output includes the creation of a model and the printing of its path.

```

[41]: # Create model if it doesn't already exist

!gcloud ai-platform models create $MODEL_NAME --regions $REGION

Using endpoint [https://ml.googleapis.com/]
Created ai platform model [projects/my-project-3837-305716/models/cta_ridership_4].

[65]: # Create the model version
export_path = BUCKET_URI + '/' + EXPORT_DIR
abais_version = 'version_' + str(int(time.time()))
print(export_path)
print(abais_version)

gs://abais-bucket-3837/tf_export
version_1614359512
  
```

## Make Predictions on deployed model

```
print('Input:')
print(X_test.tolist()[ :5])
pred_val = predict_json(PROJECT, REGION, MODEL_NAME, X_test.tolist(), version=abais_version)

print()

print('Predictions:')
print(pred_val[ :5])
```

[illegible]

Predictions:

```
[[0.659684241, 0.626064, 0.676753581, -1.0021131, -1.80011857, 0.605064034, 0.744683266], [0.608709455, 0.735271394, -1.0335294, -1.83254, 0.614551127, 0.688534737, 0.642241955], [0.718176951, -1.01311612, -1.82813942, 0.576377749, 0.706874132, 0.580318511, 0.614244103], [1.0662595, -1.86067057, 0.567883909, 0.750643671, 0.643208325, 0.624556184, 0.671228171], [-1.81234813, 0.592724057, 0.749779522, 0.7131
```



```
# Print prediction and compare to actual value
```

```
print('Predicted riders:', int(round(inverse_scale(np.array([pred_val[0][0]]).reshape(1,1))[0][0])))  
print('Actual riders: ', int(round(inverse_scale(np.array([y_test[0]]))[0][0])))
```

Predicted riders: 1646008

Actual riders: 1647321

## Cleanup

```
# Delete model version resource
```

```
!echo gcloud ai-platform versions delete {abais_version} --model {MODEL_NAME} --quiet
```

```
# Delete model resource
```

```
!echo gcloud ai-platform models delete {MODEL_NAME} --quiet
```

```
gcloud ai-platform versions delete version_1614359512 --model cta_ridership_4 --quiet
```

```
gcloud ai-platform models delete cta_ridership_4 --quiet
```

## Storage Buckets

### For My Project 3837

Google Cloud Platform

My Project 3837

Search products and resources

Storage

Storage browser

CREATE BUCKET

DELETE

REFRESH

Browser

Filter

Filter buckets

Monitoring

Settings

Bucket sorting and filtering are available in the Storage browser. Now you can filter your buckets by any value and sort by any column.

<input type="checkbox"/>	Name <span>↑</span>	Created	Location type	Location	Default storage class <span>?</span>	Updated <span>?</span>	Public access <span>?</span>	Access control <span>?</span>
<input type="checkbox"/>	abais-bucket-3837	Feb 23, 2021, 9:09:45 AM	Multi-region	us (multiple re...	Standard	Feb 23, 2021, 9:09:45 AM	Subject to object ACLs	Fine-grained
<input type="checkbox"/>	abais-bucket-3837-model_2	Feb 26, 2021, 10:55:43 AM	Multi-region	us (multiple re...	Standard	Feb 26, 2021, 10:55:43 AM	Subject to object ACLs	Fine-grained
<input type="checkbox"/>	cloud-ai-platform-dfd01098-f913-4ab7-a810-65469afcc776	Feb 23, 2021, 9:29:24 PM	Region	us-central1 (lo...	Regional	Feb 23, 2021, 9:29:24 PM	Subject to object ACLs	Fine-grained
<input type="checkbox"/>	project-3837-305716	Feb 23, 2021, 9:25:29 PM	Region	us-central1 (lo...	Standard	Feb 23, 2021, 9:25:29 PM	Subject to object ACLs	Fine-grained

### For My First Project

Google Cloud Platform

My First Project

Search products and resources

Storage

Browser

Monitoring

Settings

Storage browser

CREATE BUCKETDELETEREFRESH

Filter Filter buckets

Bucket sorting and filtering are available in the Storage browser. Now you can filter your buckets by any value and sort by any column.

<input type="checkbox"/> Name ↑	Created	Location type	Location	Default storage class	Updated	Public access	Access control
<input type="checkbox"/> abais_02cloud_bucket	Feb 22, 2021, 2:20:59 PM	Multi-region	us (multiple re...	Standard	Feb 22, 2021, 2:20:59 PM	Subject to object ACLs	Fine-grained
<input type="checkbox"/> abais_02model_bucket	Feb 22, 2021, 2:12:41 PM	Multi-region	us (multiple re...	Standard	Feb 22, 2021, 2:12:41 PM	Subject to object ACLs	Fine-grained
<input type="checkbox"/> abais_custom_bucket	Feb 22, 2021, 1:31:08 AM	Region	us-central1 (io...	Standard	Feb 22, 2021, 1:31:08 AM	Subject to object ACLs	Fine-grained
<input type="checkbox"/> abais_custom_bucket-1	Feb 22, 2021, 8:59:56 AM	Region	us-central1 (io...	Standard	Feb 22, 2021, 8:59:56 AM	Subject to object ACLs	Fine-grained
<input type="checkbox"/> abais_hello_custom_lucid-ceremony-305523	Feb 21, 2021, 8:04:02 PM	Region	us-central1 (io...	Standard	Feb 21, 2021, 8:04:02 PM	Subject to object ACLs	Fine-grained
<input type="checkbox"/> abaisflowers_bucket	Feb 22, 2021, 9:41:12 AM	Region	us-central1 (io...	Standard	Feb 22, 2021, 9:41:12 AM	Subject to object ACLs	Fine-grained
<input type="checkbox"/> cloud-ai-platform-acb62e23-66da-410e-b124-7ea4f31488...	Feb 20, 2021, 4:11:52 PM	Region	us-central1 (io...	Regional	Feb 20, 2021, 4:11:52 PM	Subject to object ACLs	Fine-grained
<input type="checkbox"/> lucid-ceremony-305523	Feb 20, 2021, 4:50:40 PM	Region	us-central1 (io...	Standard	Feb 20, 2021, 4:50:40 PM	Subject to object ACLs	Fine-grained
<input type="checkbox"/> lucid-ceremony-305523-lcm	Feb 21, 2021, 4:33:21 PM	Region	us-central1 (io...	Standard	Feb 21, 2021, 4:33:21 PM	Subject to object ACLs	Fine-grained