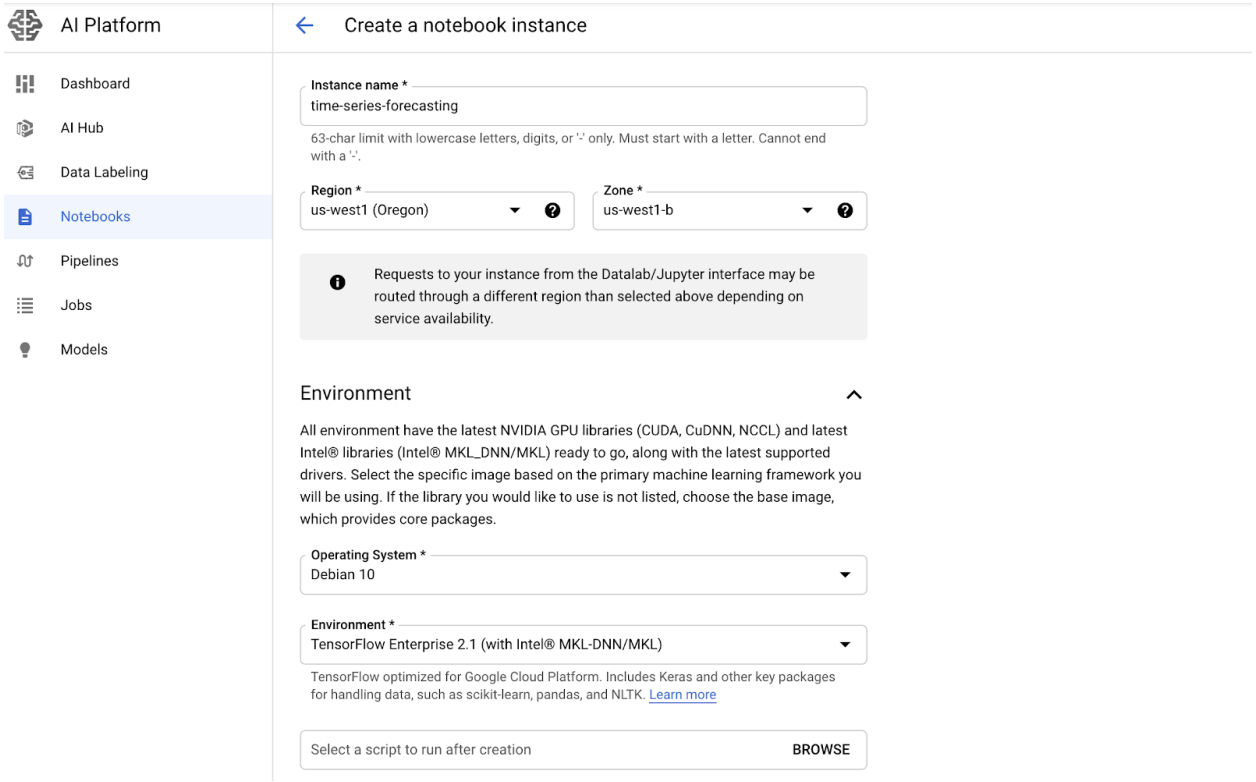
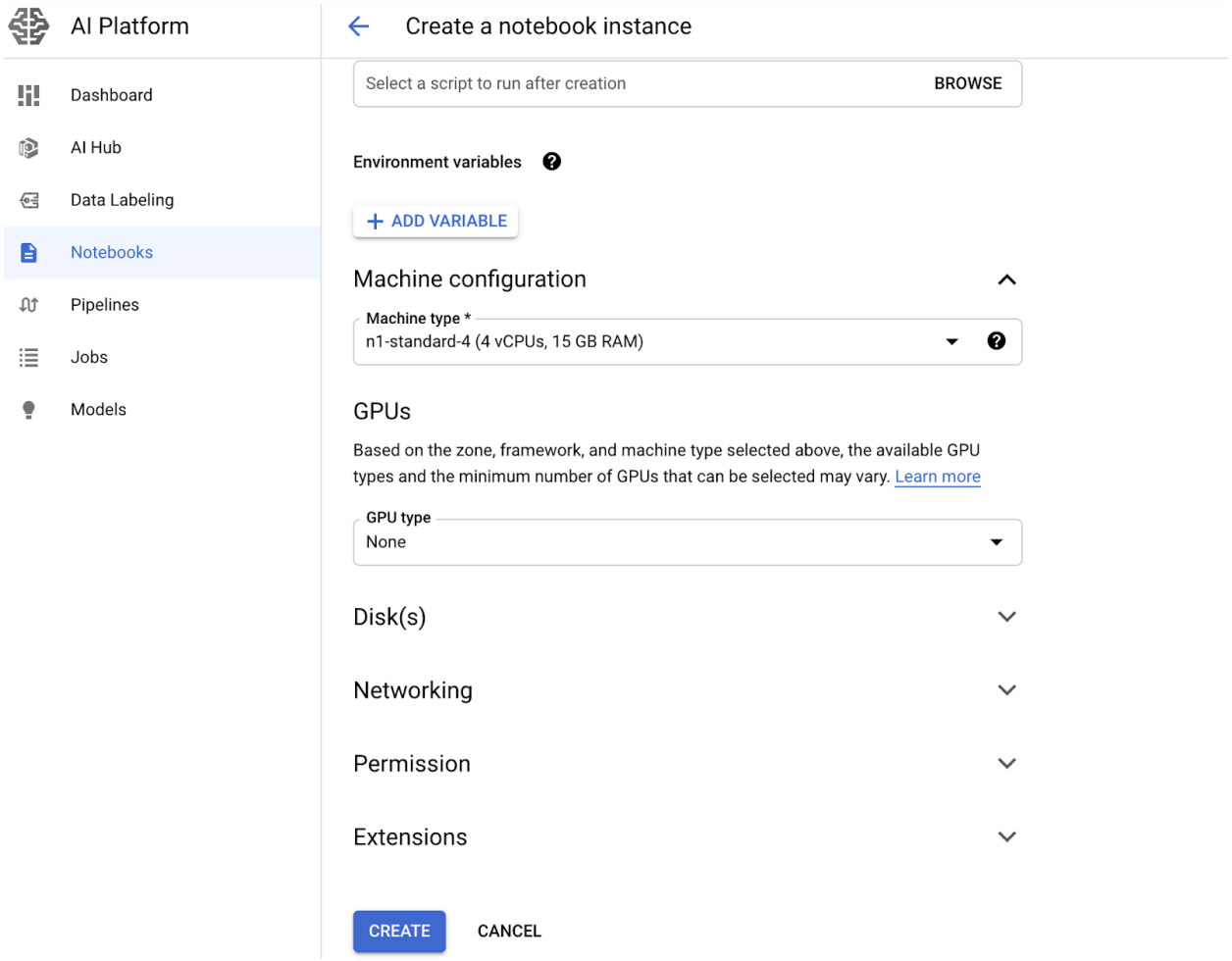
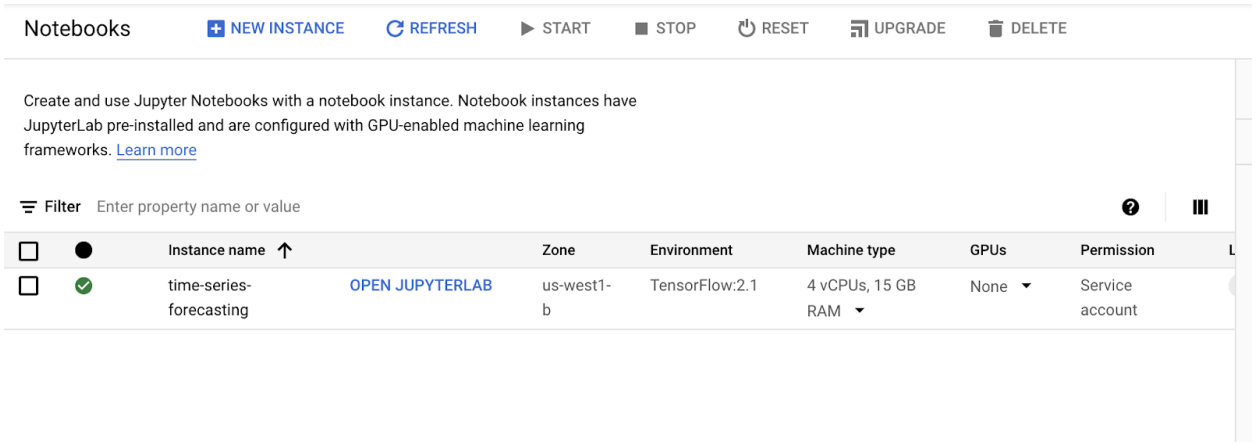
Time Series Forecasting




AI Platform
Create a notebook instance
Dashboard
AI Hub
Data Labeling
Notebooks
Pipelines
Jobs
Models
Select a script to run after creation
BROWSE
Environment variables
ADD VARIABLE
Machine configuration
Machine type *
n1-standard-4 (4 vCPUs, 15 GB RAM)
GPUs
Based on the zone, framework, and machine type selected above, the available GPU types and the minimum number of GPUs that can be selected may vary. Learn more
GPU type
None
Disk(s)
Networking
Permission
Extensions
CREATE
CANCEL


Notebooks
NEW INSTANCE
REFRESH
START
STOP
RESET
UPGRADE
DELETE
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
time-series-forecasting
OPEN JUPYTERLAB
us-west1-b
TensorFlow:2.1
4 vCPUs, 15 GB RAM
None
Service account

```
(base) jupyter@time-series-forecasting:~$ 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 | 30.54 MiB/s, done.
Resolving deltas: 100% (28664/28664), done.
Checking out files: 100% (9198/9198), done.
(base) jupyter@time-series-forecasting:~$
```

Install packages and dependencies

Restarting the kernel may be required to use new packages.

```
%pip install -U statsmodels scikit-learn --user
```

```
Requirement already satisfied: statsmodels in /opt/conda/lib/python3.7/site-packages (0.12.2)
Requirement already satisfied: scikit-learn in /opt/conda/lib/python3.7/site-packages (0.24.1)
Requirement already satisfied: joblib>=0.11 in /opt/conda/lib/python3.7/site-packages (from scikit-learn) (1.0.1)
Requirement already satisfied: numpy>=1.13.3 in /opt/conda/lib/python3.7/site-packages (from scikit-learn) (1.19.5)
Requirement already satisfied: scipy>=0.19.1 in /opt/conda/lib/python3.7/site-packages (from scikit-learn) (1.6.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/conda/lib/python3.7/site-packages (from scikit-learn) (2.1.0)
Requirement already satisfied: patsy>=0.5 in /opt/conda/lib/python3.7/site-packages (from statsmodels) (0.5.1)
Requirement already satisfied: pandas>=0.21 in /opt/conda/lib/python3.7/site-packages (from statsmodels) (1.2.2)
Requirement already satisfied: python-dateutil>=2.7.3 in /opt/conda/lib/python3.7/site-packages (from pandas>=0.21->statsmodels) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-packages (from pandas>=0.21->statsmodels) (2021.1)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from patsy>=0.5->statsmodels) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

Note: To restart the Kernel, navigate to Kernel > Restart Kernel... on the Jupyter menu.

Import libraries and define constants

```
from pandas.plotting import register_matplotlib_converters
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import grangercausalitytests

import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

```
# 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 = 'Deep Learning - Assignment 2' # 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.
```

01-explore.ipynb | Terminal 1

Code | git | Pyth

[3]:
```
# 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 = 'Deep Learning - Assignment 2' # 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!'
```

[4]:
```
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

[5]:
```
# Import CSV file

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

[6]:
```
# Model data prior to 2020

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

[7]:
```
# Drop duplicates

df = df.drop_duplicates()
```

[8]:
```
# Sort by date

df = df.sort_index()
```

Explore data

[9]:
```
# Print the top 5 rows

df.head()
```

[9]:

service_date	day_type	bus	rail_boardings	total_rides
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

TODO 1: Analyze the patterns

- Is ridership changing much over time?
- Is there a difference in ridership between the weekday and weekends?
- Is the mix of bus vs rail ridership changing over time?

[10]:

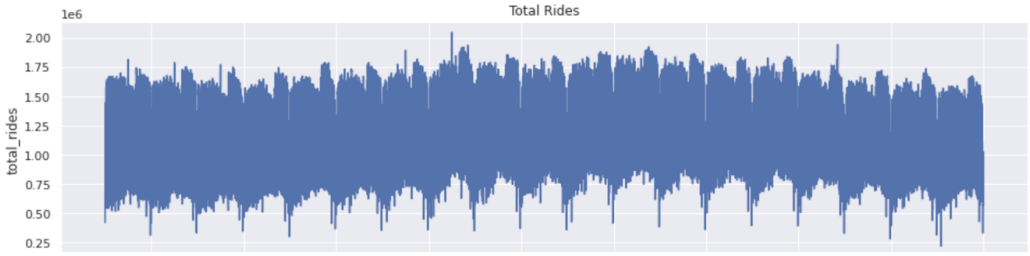
```
# Initialize plotting

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

[11]:

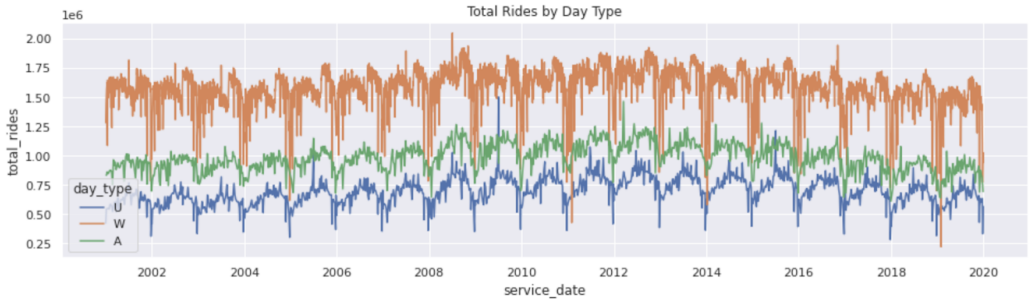
```
# Explore total rides over time

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



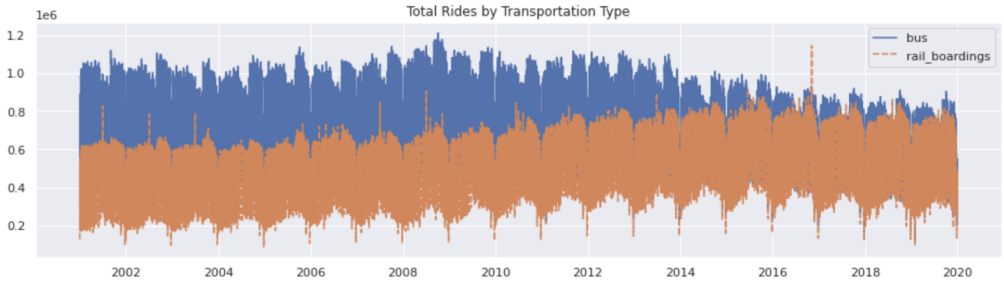
[12]:
```
# Explore rides by day type: Weekday (W), Saturday (A), Sunday/Holiday (U)

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



[13]:
```
# Explore rides by transportation type

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



TODO 2: Review summary statistics

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

[14]:
```
df[target].describe().apply(lambda x: round(x))
```

[14]:
```
count       6939
mean     1368761
std       391443
min       222071
25%      1005394
50%      1548343
75%      1660947
max      2049519
Name: total_rides, dtype: int64
```

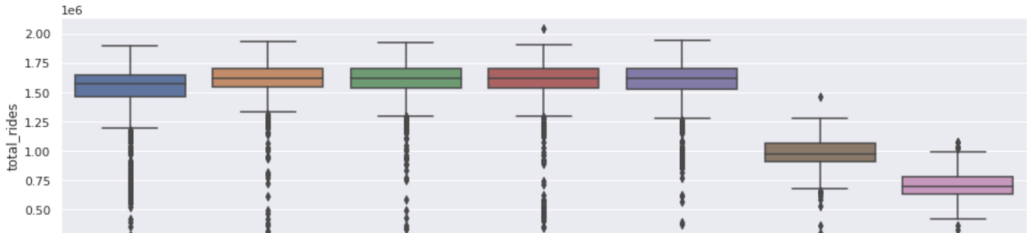
TODO 3: Explore seasonality

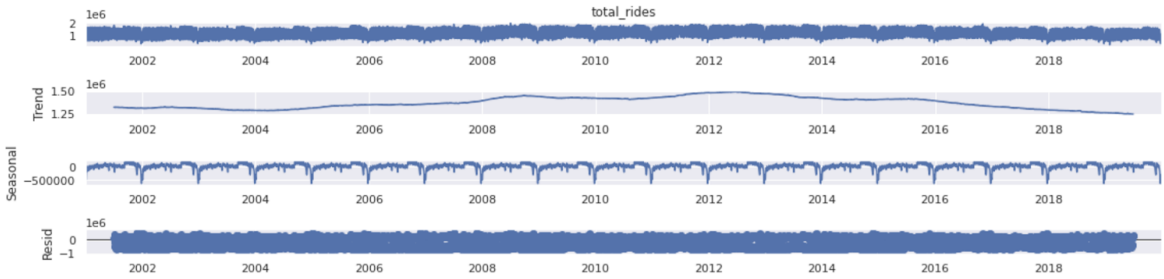
- Is there much difference between months?
- Can you extract the trend and seasonal pattern from the data?

[15]:
```
# 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])
```





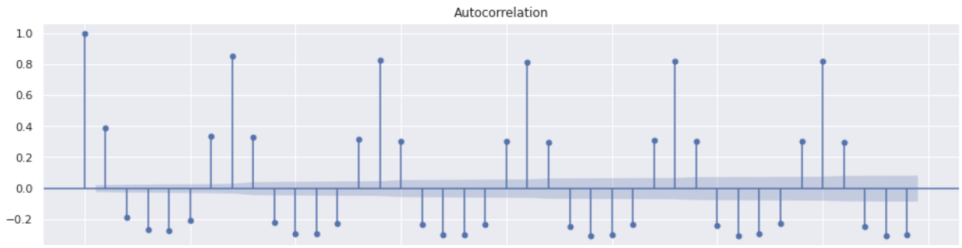
Auto-correlation

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).

[18]:
```
plot_acf(df[target])

fig = plt.show()
```

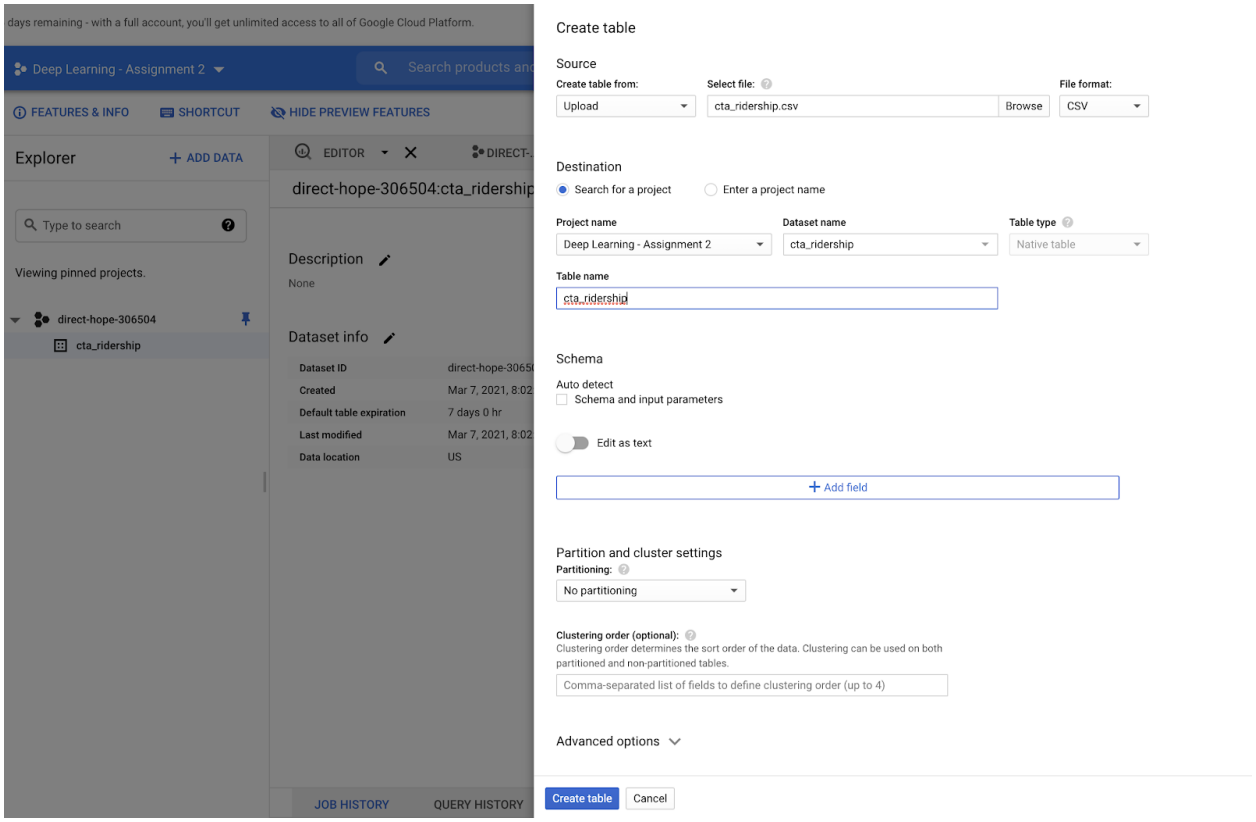


Export data

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.

[19]:
```
df[[target]].to_csv(processed_file, index=True, index_label=ts_col)
```

Conclusion




```
CREATE OR REPLACE MODEL
  `direct-hope-306504.cta_ridership.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
`direct-hope-306504.cta_ridership.cta_ridership_table`
```

Query results

Job information | Results | Execution details

This statement will create a new model named direct-hope-306504:cta_ridership.cta_ridership_model. Depending on the type of model, this may take several hours to complete. Go to model

```sql
SELECT
  *
FROM
  ML.EVALUATE(MODEL `direct-hope-306504.cta_ridership.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

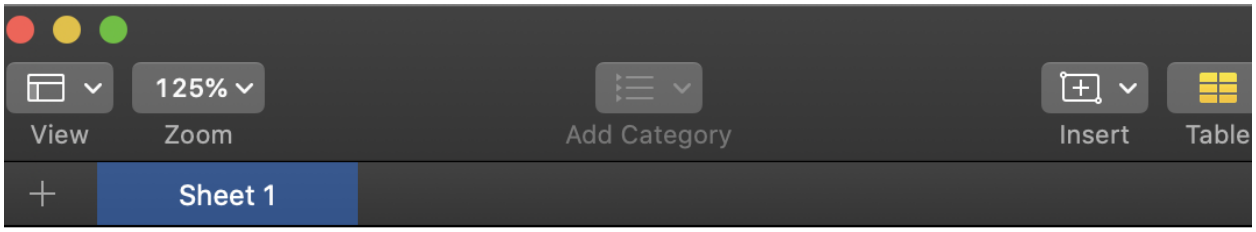
```sql
SELECT
  *
FROM
  ML.FORECAST(MODEL `direct-hope-306504.cta_ridership.cta_ridership_model`,
    STRUCT(7 AS horizon))
```

Query results SAVE RESULTS EXPLORE DATA

Query complete (0.1 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_lowe
	2020-01-01 00:00:00 UTC	662436.4424369269	46059.49014554253	0.95	572322.980240453	752549.9046334007	572322.98(
2	2020-01-02 00:00:00 UTC	1029641.4669424891	46276.328347693256	0.95	939103.76989082	1120179.1639941582	939103.76
3	2020-01-03 00:00:00 UTC	1201660.2034356925	47233.43871922012	0.95	1109249.9600529654	1294070.4468184195	1109249.960(
4	2020-01-04 00:00:00 UTC	651095.9776391207	48157.99332862347	0.95	556876.8819095747	745315.0733686666	556876.8819
5	2020-01-05 00:00:00 UTC	467394.91846646497	48621.50963880497	0.95	372268.97250121285	562520.8644317171	372268.97250
6	2020-01-06 00:00:00 UTC	1158999.319539823	48869.23710364581	0.95	1063388.705171438	1254609.9339082083	1063388.705



cta_ridership

service_date	total_rides
2001-01-01	423647
2001-01-02	1282779
2001-01-03	1361355
2001-01-04	1420032
2001-01-05	1448343
2001-01-06	832757
2001-01-07	545656
2001-01-08	1575927
2001-01-09	1578282
2001-01-10	1586936
2001-01-11	1603064
2001-01-12	1624237
2001-01-13	861847
2001-01-14	547933
2001-01-15	1087994
2001-01-16	1646530
2001-01-17	1639033
2001-01-18	1625828
2001-01-19	1493815
2001-01-20	846163
2001-01-21	550488

IMAGE dataset modified