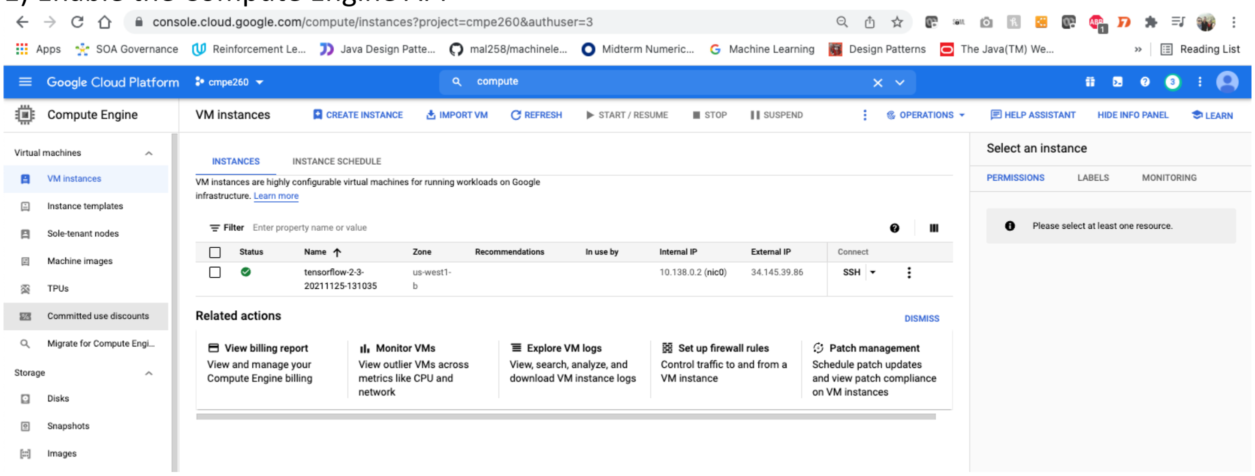
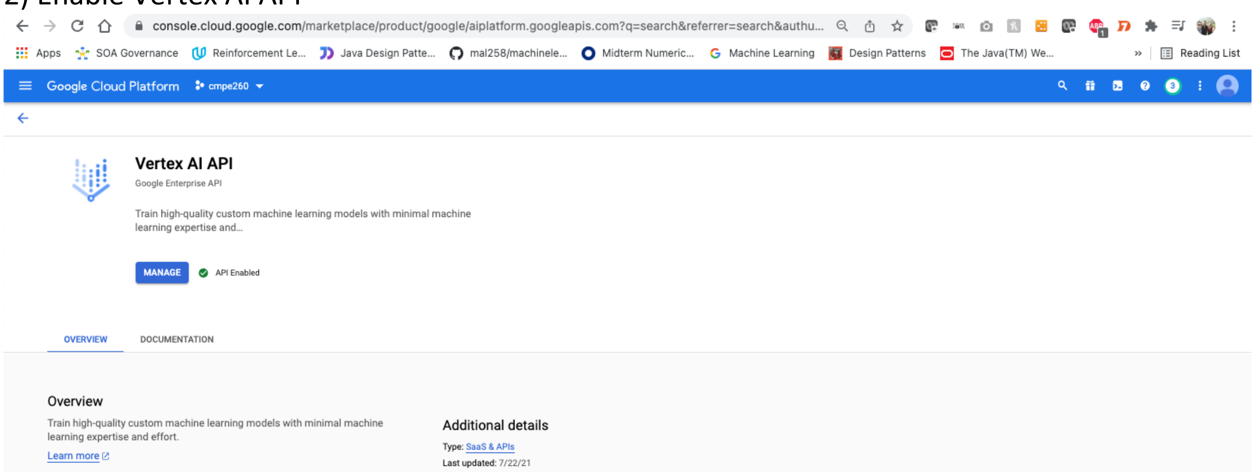
Assignment 7: optional catchup assignment 2 - VERTEX AI - for midterm and quiz - this will catch up midterm.

Vertex AI Workbench: Train a TensorFlow model with data from BigQuery

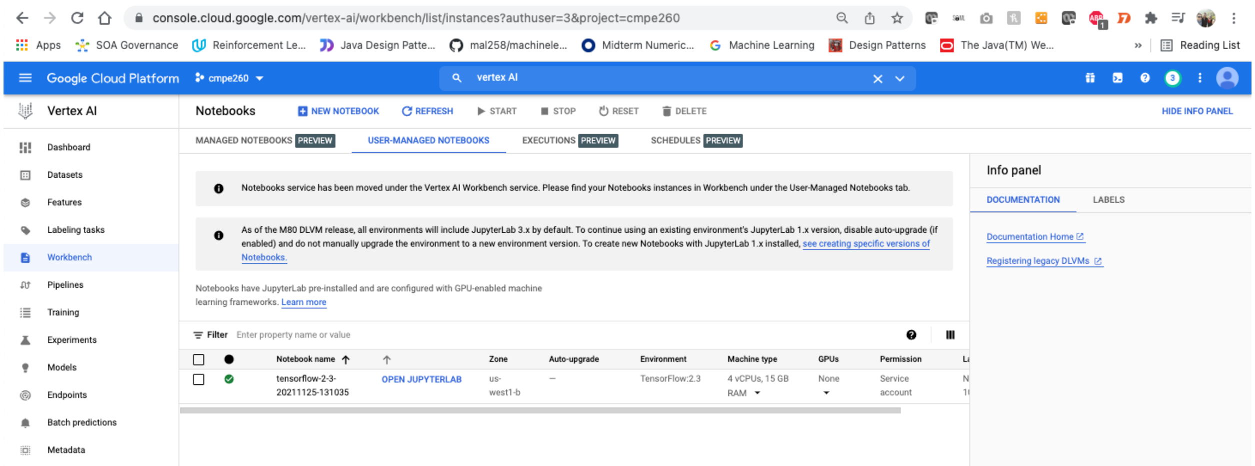
1) Enable the Compute Engine API



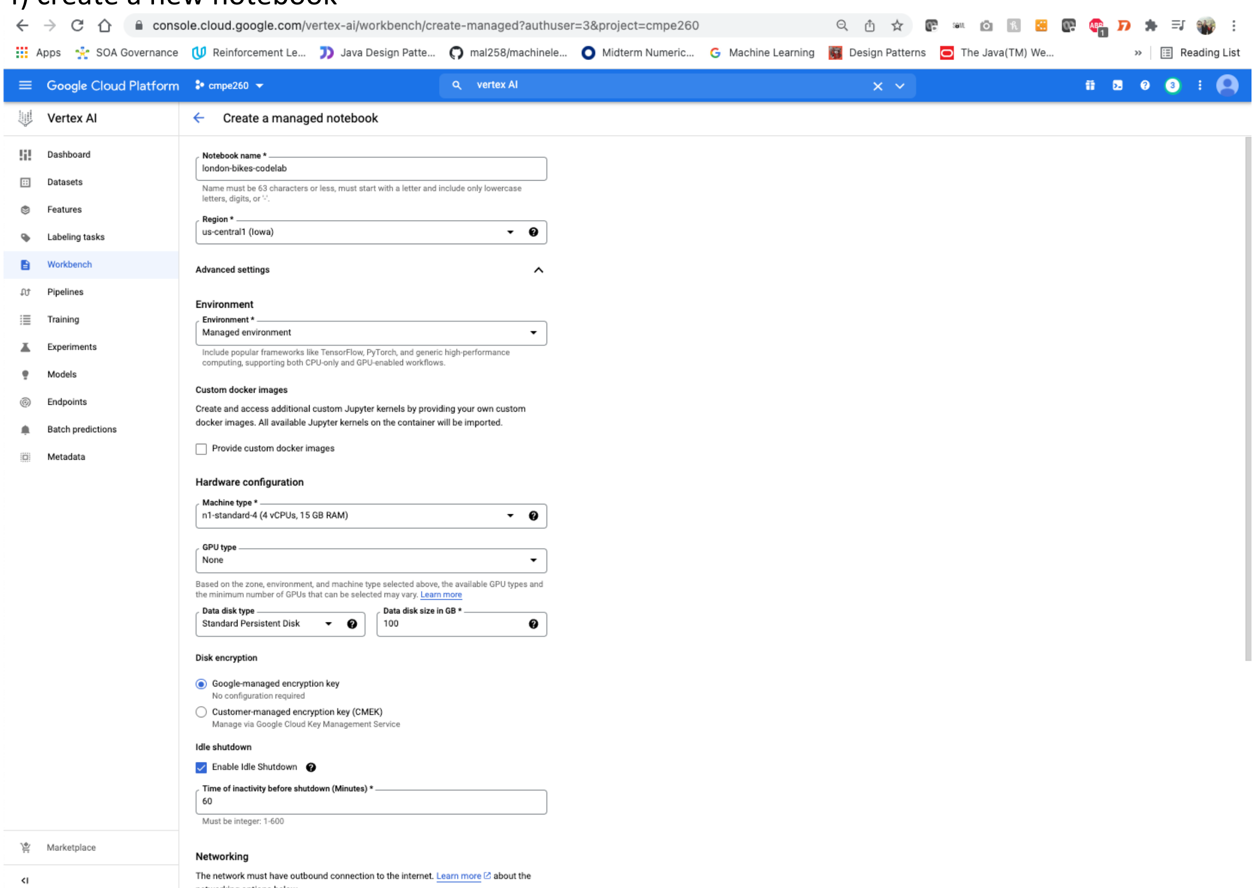
2) Enable Vertex AI API



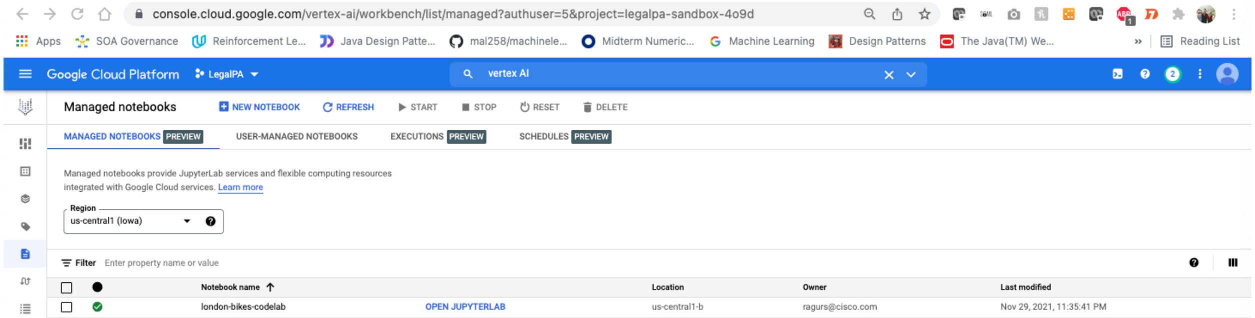
3) Create Vertex AI Workbench instance

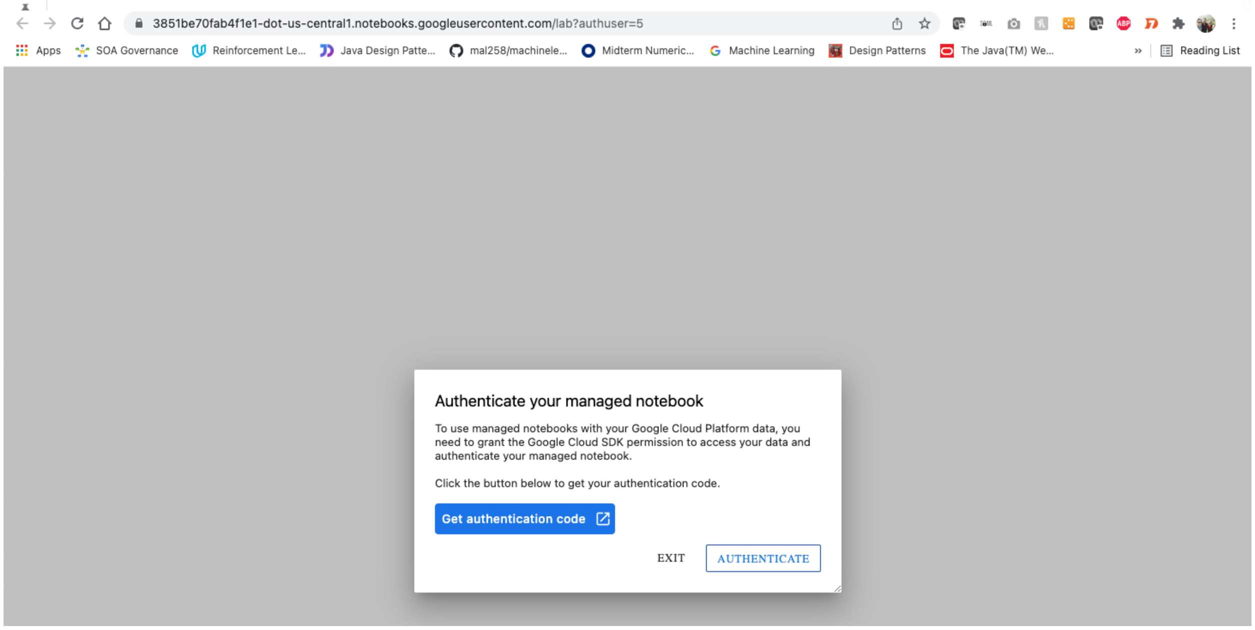


4) create a new notebook



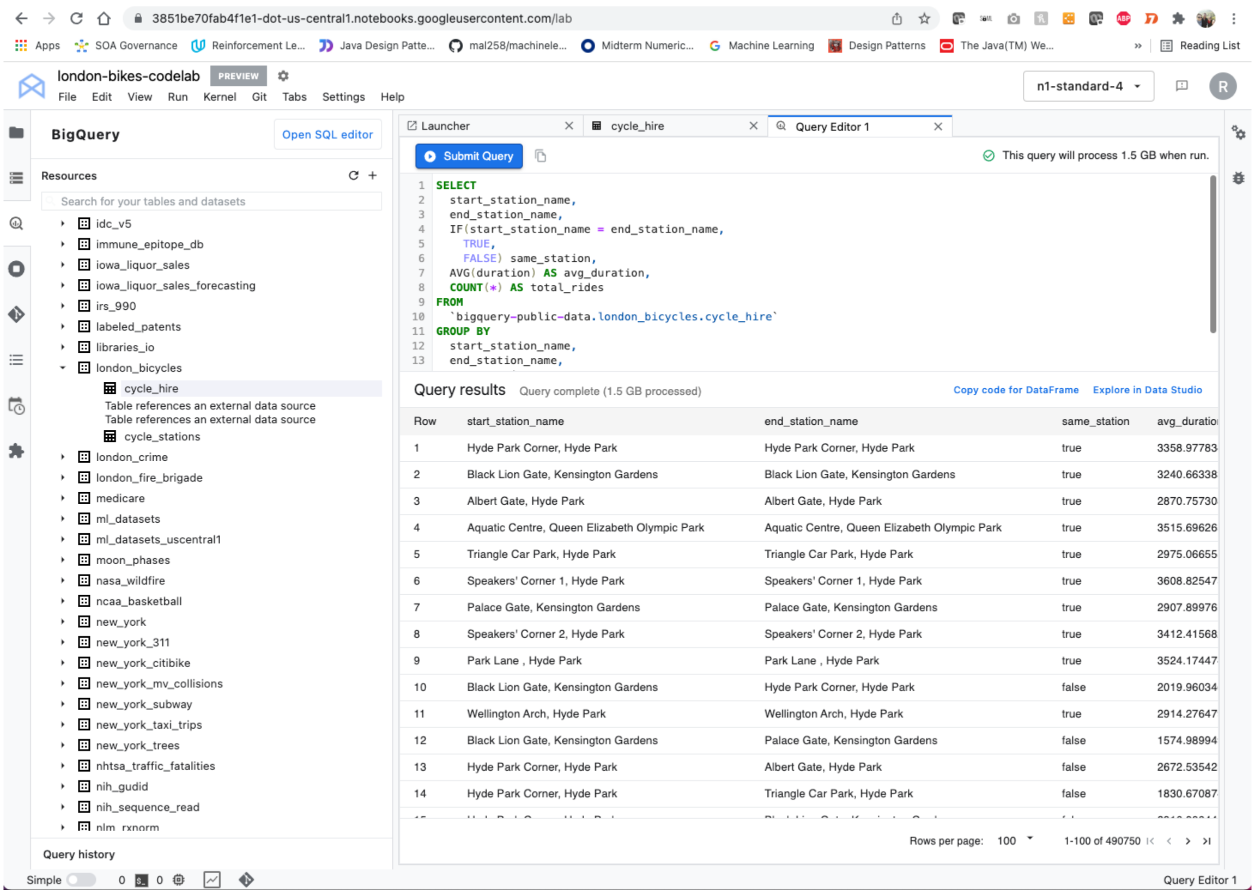
5)



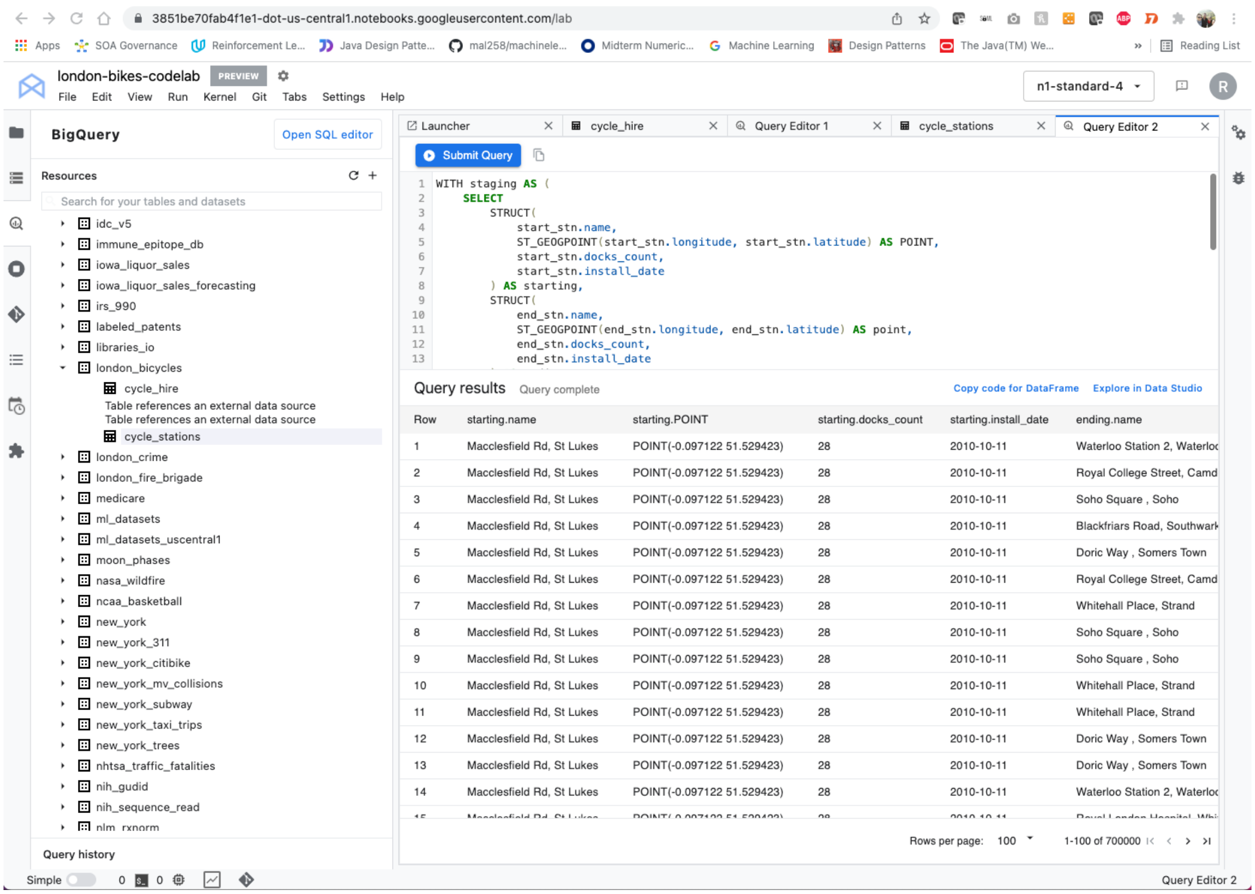


6)

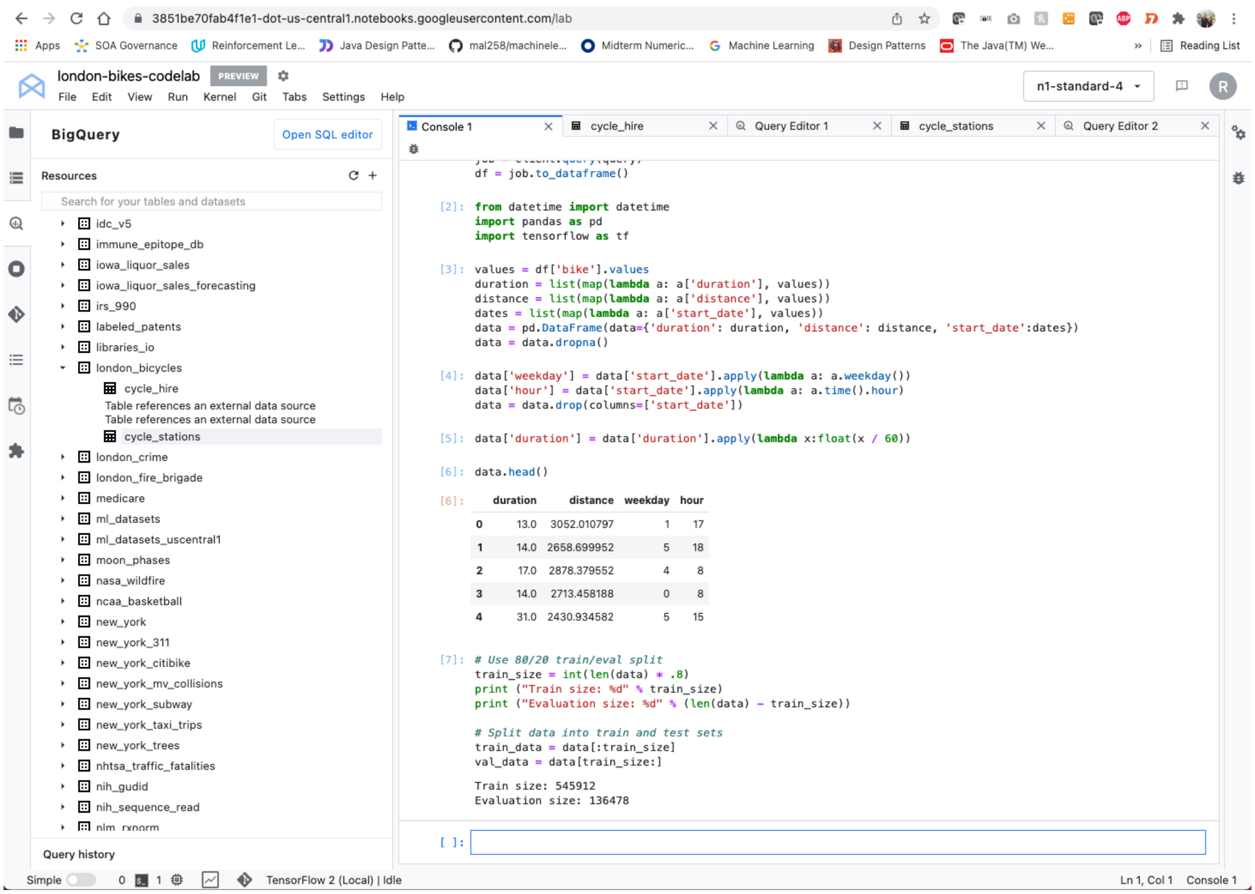
Dataset exploration in BigQuery



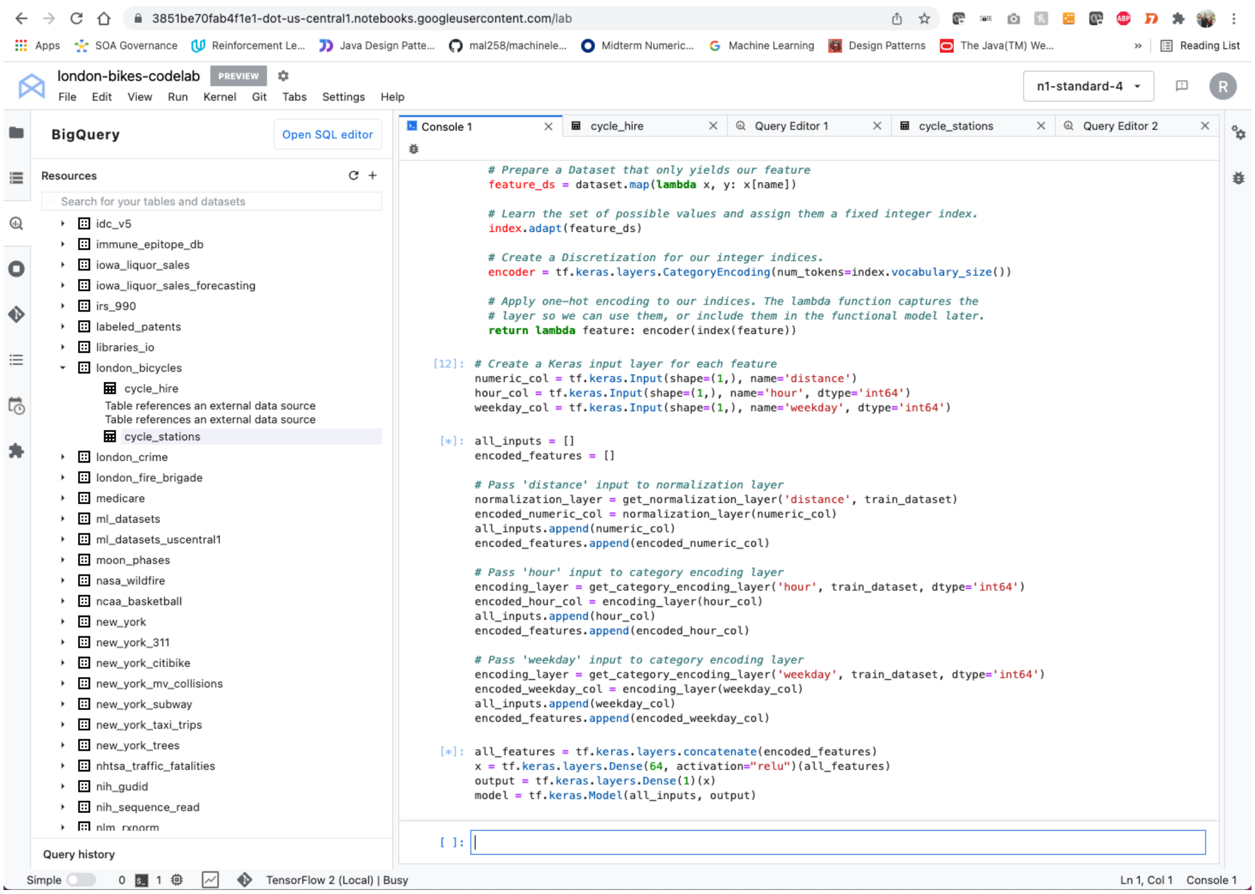
3851be70fab4f1e1-dot-us-central1.notebooks.googleusercontent.com/lab
Apps
SOA Governance
Reinforcement Le...
Java Design Patte...
mal258/machinele...
Midterm Numeric...
Machine Learning
Design Patterns
The Java(TM) We...
Reading List
london-bikes-codelab
PREVIEW
File
Edit
View
Run
Kernel
Git
Tabs
Settings
Help
n1-standard-4
BigQuery
Open SQL editor
Resources
Search for your tables and datasets
idc_v5
immune_epitope_db
iowa_liquor_sales
iowa_liquor_sales_forecasting
irs_990
labeled_patents
libraries_io
london_bicycles
cycle_hire
Table references an external data source
Table references an external data source
cycle_stations
london_crime
london_fire_brigade
medicare
ml_datasets
ml_datasets_uscentral1
moon_phases
nasa_wildfire
ncaa_basketball
new_york
new_york_311
new_york_citibike
new_york_mv_collisions
new_york_subway
new_york_taxi_trips
new_york_trees
nhtsa_traffic_fatalities
nih_gudid
nih_sequence_read
Query history
Simple
Launcher
cycle_hire
Query Editor 1
Submit Query
This query will process 1.5 GB when run.
SELECT
start_station_name,
end_station_name,
IF(start_station_name = end_station_name,
TRUE,
FALSE) same_station,
AVG(duration) AS avg_duration,
COUNT(*) AS total_rides
FROM
`bigquery-public-data.london_bicycles.cycle_hire`
GROUP BY
start_station_name,
end_station_name,
Query results
Query complete (1.5 GB processed)
Copy code for DataFrame
Explore in Data Studio
Row | start_station_name | end_station_name | same_station | avg_duratio
1 | Hyde Park Corner, Hyde Park | Hyde Park Corner, Hyde Park | true | 3358.97783
2 | Black Lion Gate, Kensington Gardens | Black Lion Gate, Kensington Gardens | true | 3240.66338
3 | Albert Gate, Hyde Park | Albert Gate, Hyde Park | true | 2870.75730
4 | Aquatic Centre, Queen Elizabeth Olympic Park | Aquatic Centre, Queen Elizabeth Olympic Park | true | 3515.69626
5 | Triangle Car Park, Hyde Park | Triangle Car Park, Hyde Park | true | 2975.06655
6 | Speakers' Corner 1, Hyde Park | Speakers' Corner 1, Hyde Park | true | 3608.82547
7 | Palace Gate, Kensington Gardens | Palace Gate, Kensington Gardens | true | 2907.89976
8 | Speakers' Corner 2, Hyde Park | Speakers' Corner 2, Hyde Park | true | 3412.41568
9 | Park Lane , Hyde Park | Park Lane , Hyde Park | true | 3524.17447
10 | Black Lion Gate, Kensington Gardens | Hyde Park Corner, Hyde Park | false | 2019.96034
11 | Wellington Arch, Hyde Park | Wellington Arch, Hyde Park | true | 2914.27647
12 | Black Lion Gate, Kensington Gardens | Palace Gate, Kensington Gardens | false | 1574.98994
13 | Hyde Park Corner, Hyde Park | Albert Gate, Hyde Park | false | 2672.53542
14 | Hyde Park Corner, Hyde Park | Triangle Car Park, Hyde Park | false | 1830.67087
Rows per page: 100
1-100 of 490750
Query Editor 1

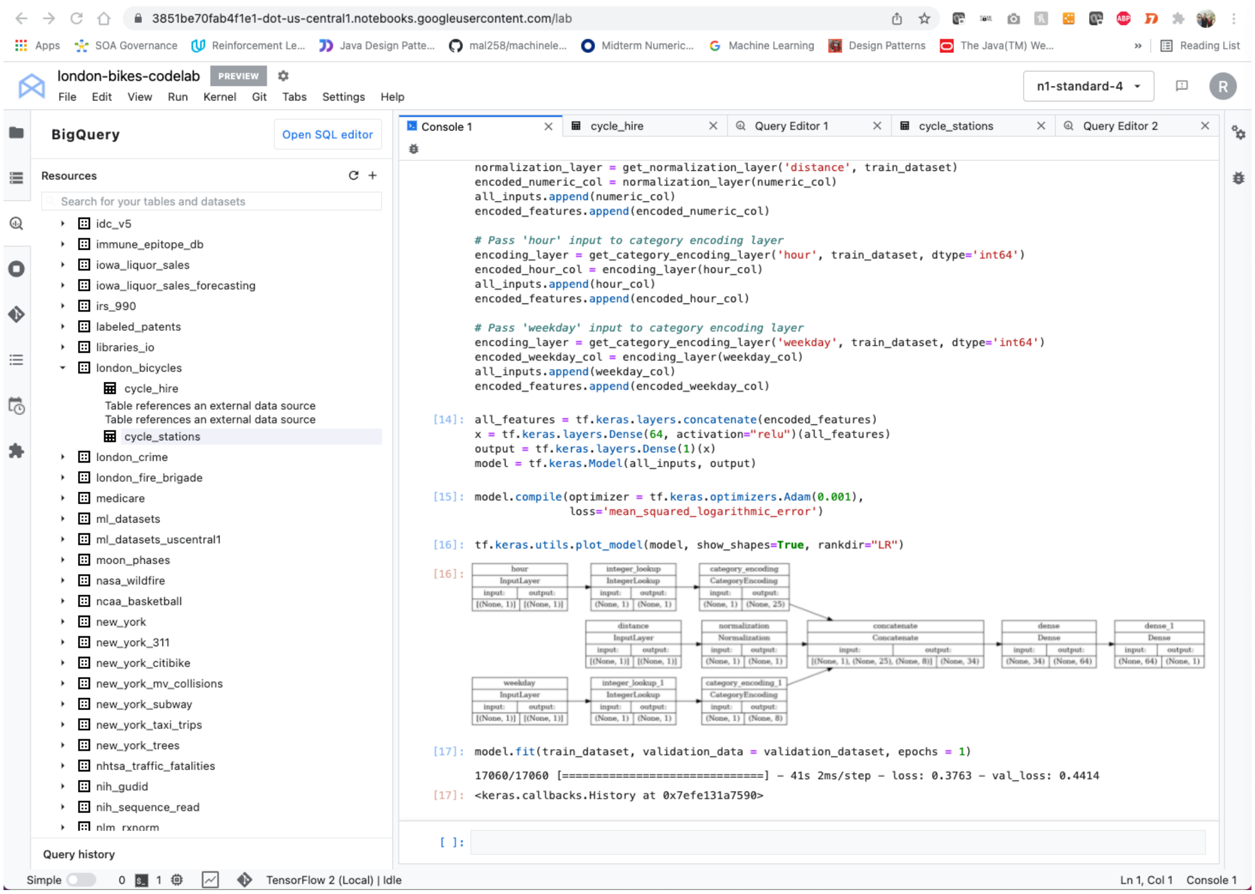


7)Creating the dataframe



Training model on tensorflow kernel


3851be70fab4f1e1-dot-us-central1.notebooks.googleusercontent.com/lab
Apps
SOA Governance
Reinforcement Le...
Java Design Patte...
mal258/machinele...
Midterm Numeric...
Machine Learning
Design Patterns
The Java(TM) We...
Reading List
london-bikes-codelab
PREVIEW
File Edit View Run Kernel Git Tabs Settings Help
n1-standard-4
BigQuery
Open SQL editor
Resources
Search for your tables and datasets
idc_v5
immune_epitope_db
iowa_liquor_sales
iowa_liquor_sales_forecasting
irs_990
labeled_patents
libraries_io
london_bicycles
cycle_hire
Table references an external data source
Table references an external data source
cycle_stations
london_crime
london_fire_brigade
medicare
ml_datasets
ml_datasets_uscentral1
moon_phases
nasa_wildfire
ncaa_basketball
new_york
new_york_311
new_york_citibike
new_york_mv_collisions
new_york_subway
new_york_taxi_trips
new_york_trees
nhtsa_traffic_fatalities
nih_gudid
nih_sequence_read
nlm_rxnorm
Query history
Console 1
cycle_hire
Query Editor 1
cycle_stations
Query Editor 2
# Prepare a Dataset that only yields our feature
feature_ds = dataset.map(lambda x, y: x[name])
# Learn the set of possible values and assign them a fixed integer index.
index.adapt(feature_ds)
# Create a Discretization for our integer indices.
encoder = tf.keras.layers.CategoryEncoding(num_tokens=index.vocabulary_size())
# Apply one-hot encoding to our indices. The lambda function captures the
# layer so we can use them, or include them in the functional model later.
return lambda feature: encoder(index(feature))
[12]: # Create a Keras input layer for each feature
numeric_col = tf.keras.Input(shape=(1,), name='distance')
hour_col = tf.keras.Input(shape=(1,), name='hour', dtype='int64')
weekday_col = tf.keras.Input(shape=(1,), name='weekday', dtype='int64')
[*]: all_inputs = []
encoded_features = []
# Pass 'distance' input to normalization layer
normalization_layer = get_normalization_layer('distance', train_dataset)
encoded_numeric_col = normalization_layer(numeric_col)
all_inputs.append(numeric_col)
encoded_features.append(encoded_numeric_col)
# Pass 'hour' input to category encoding layer
encoding_layer = get_category_encoding_layer('hour', train_dataset, dtype='int64')
encoded_hour_col = encoding_layer(hour_col)
all_inputs.append(hour_col)
encoded_features.append(encoded_hour_col)
# Pass 'weekday' input to category encoding layer
encoding_layer = get_category_encoding_layer('weekday', train_dataset, dtype='int64')
encoded_weekday_col = encoding_layer(weekday_col)
all_inputs.append(weekday_col)
encoded_features.append(encoded_weekday_col)
[*]: all_features = tf.keras.layers.concatenate(encoded_features)
x = tf.keras.layers.Dense(64, activation="relu")(all_features)
output = tf.keras.layers.Dense(1)(x)
model = tf.keras.Model(all_inputs, output)
[ ]:
Simple
TensorFlow 2 (Local) | Busy
Ln 1, Col 1
Console 1


3851be70fab4f1e1-dot-us-central1.notebooks.googleusercontent.com/lab
Apps
SOA Governance
Reinforcement Le...
Java Design Patte...
mal258/machinele...
Midterm Numeric...
Machine Learning
Design Patterns
The Java(TM) We...
Reading List
london-bikes-codelab
PREVIEW
File Edit View Run Kernel Git Tabs Settings Help
n1-standard-4
BigQuery
Open SQL editor
Resources
Search for your tables and datasets
idc_v5
immune_epitope_db
iowa_liquor_sales
iowa_liquor_sales_forecasting
irs_990
labeled_patents
libraries_io
london_bicycles
cycle_hire
Table references an external data source
Table references an external data source
cycle_stations
london_crime
london_fire_brigade
medicare
ml_datasets
ml_datasets_uscentral1
moon_phases
nasa_wildfire
ncaa_basketball
new_york
new_york_311
new_york_citibike
new_york_mv_collisions
new_york_subway
new_york_taxi_trips
new_york_trees
nhtsa_traffic_fatalities
nih_gudid
nih_sequence_read
nlm_rxnorm
Query history
Console 1
cycle_hire
Query Editor 1
cycle_stations
Query Editor 2
normalization_layer = get_normalization_layer('distance', train_dataset)
encoded_numeric_col = normalization_layer(numeric_col)
all_inputs.append(numeric_col)
encoded_features.append(encoded_numeric_col)
# Pass 'hour' input to category encoding layer
encoding_layer = get_category_encoding_layer('hour', train_dataset, dtype='int64')
encoded_hour_col = encoding_layer(hour_col)
all_inputs.append(hour_col)
encoded_features.append(encoded_hour_col)
# Pass 'weekday' input to category encoding layer
encoding_layer = get_category_encoding_layer('weekday', train_dataset, dtype='int64')
encoded_weekday_col = encoding_layer(weekday_col)
all_inputs.append(weekday_col)
encoded_features.append(encoded_weekday_col)
[14]: all_features = tf.keras.layers.concatenate(encoded_features)
x = tf.keras.layers.Dense(64, activation="relu")(all_features)
output = tf.keras.layers.Dense(1)(x)
model = tf.keras.Model(all_inputs, output)
[15]: model.compile(optimizer = tf.keras.optimizers.Adam(0.001),
loss='mean_squared_logarithmic_error')
[16]: tf.keras.utils.plot_model(model, show_shapes=True, rankdir="LR")
[16]:
[17]: model.fit(train_dataset, validation_data = validation_dataset, epochs = 1)
17060/17060 [==============================] - 41s 2ms/step - loss: 0.3763 - val_loss: 0.4414
[17]: <keras.callbacks.History at 0x7efe131a7590>
[ ]:
Simple
TensorFlow 2 (Local) | Idle
Ln 1, Col 1
Console 1