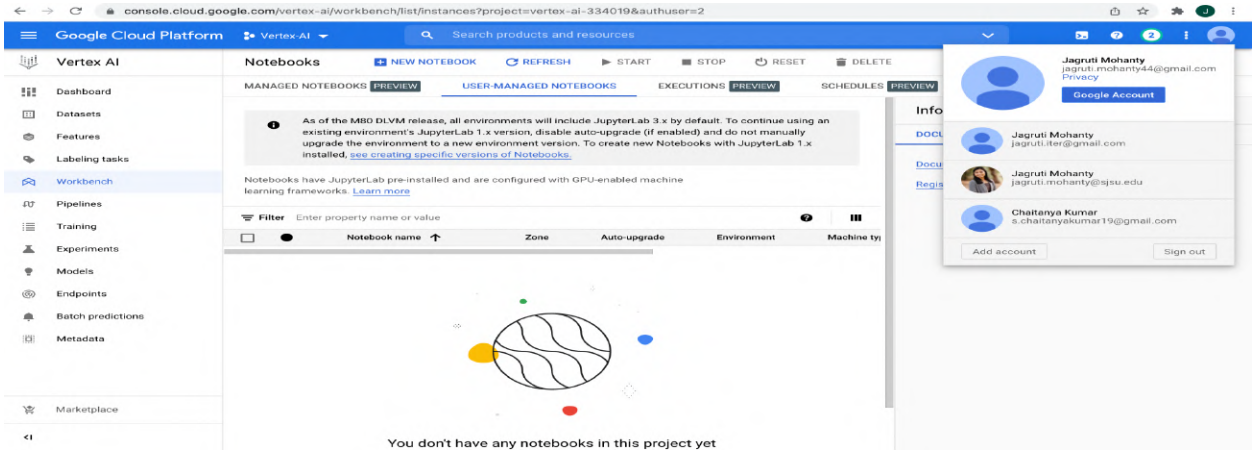
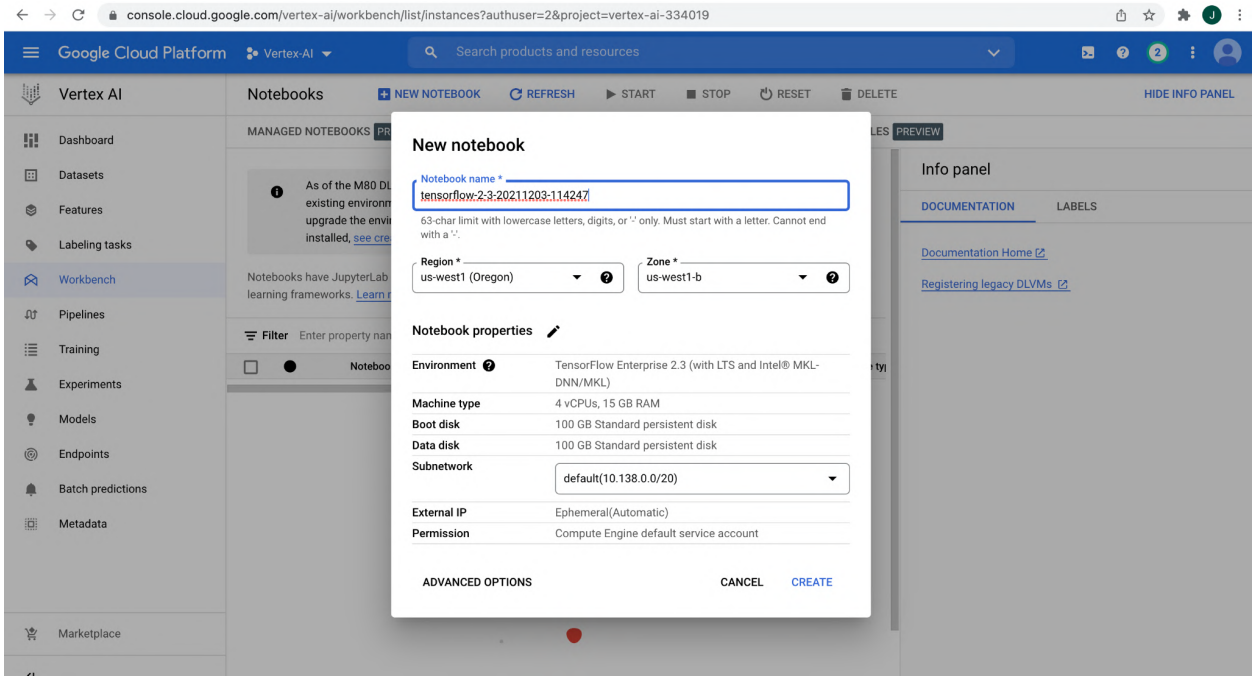
Name : Jagruti Mohanty

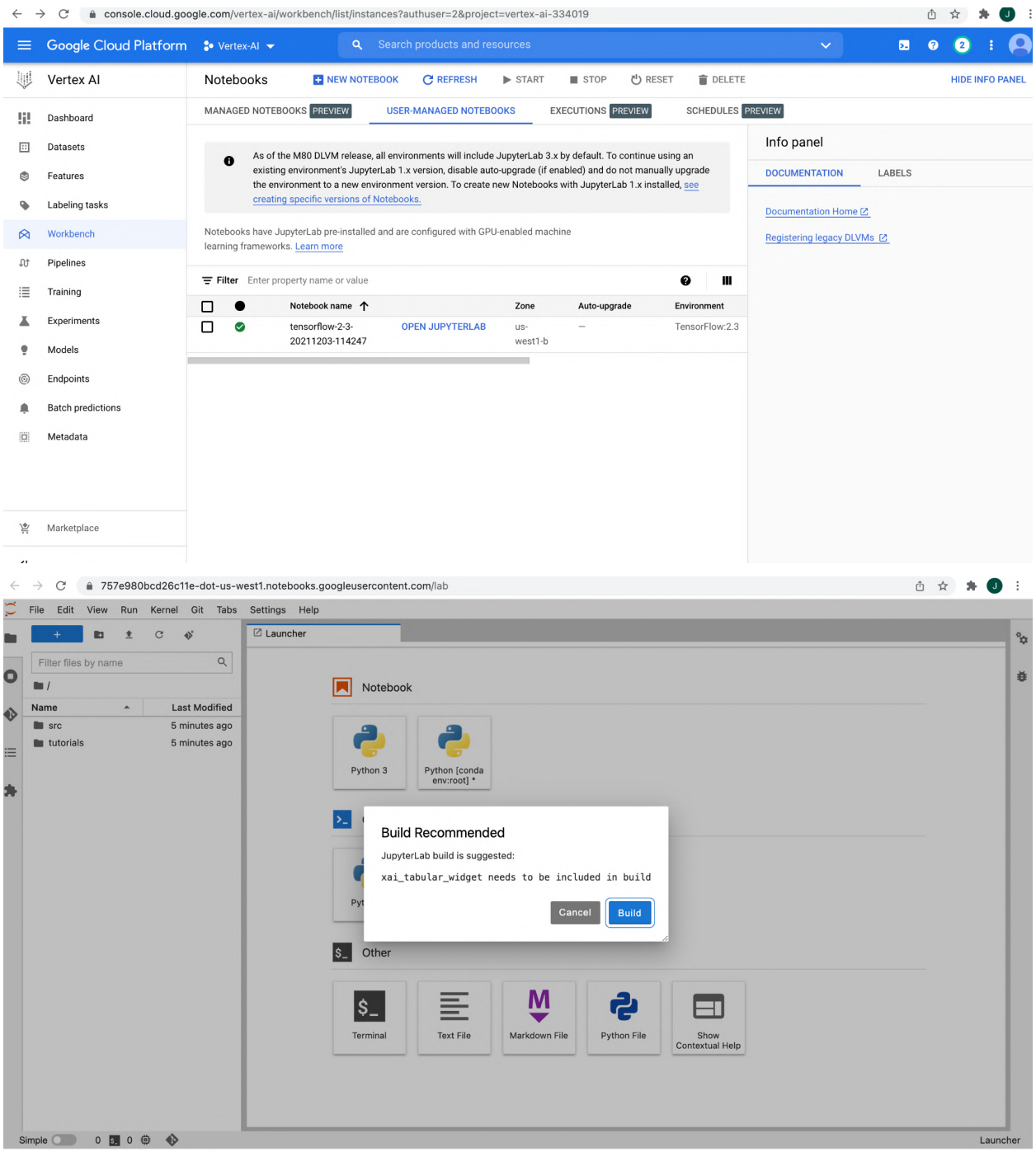
Assignment 7 - VERTEX AI

a)

https://codelabs.developers.google.com/vertex-pipelines-intro#0

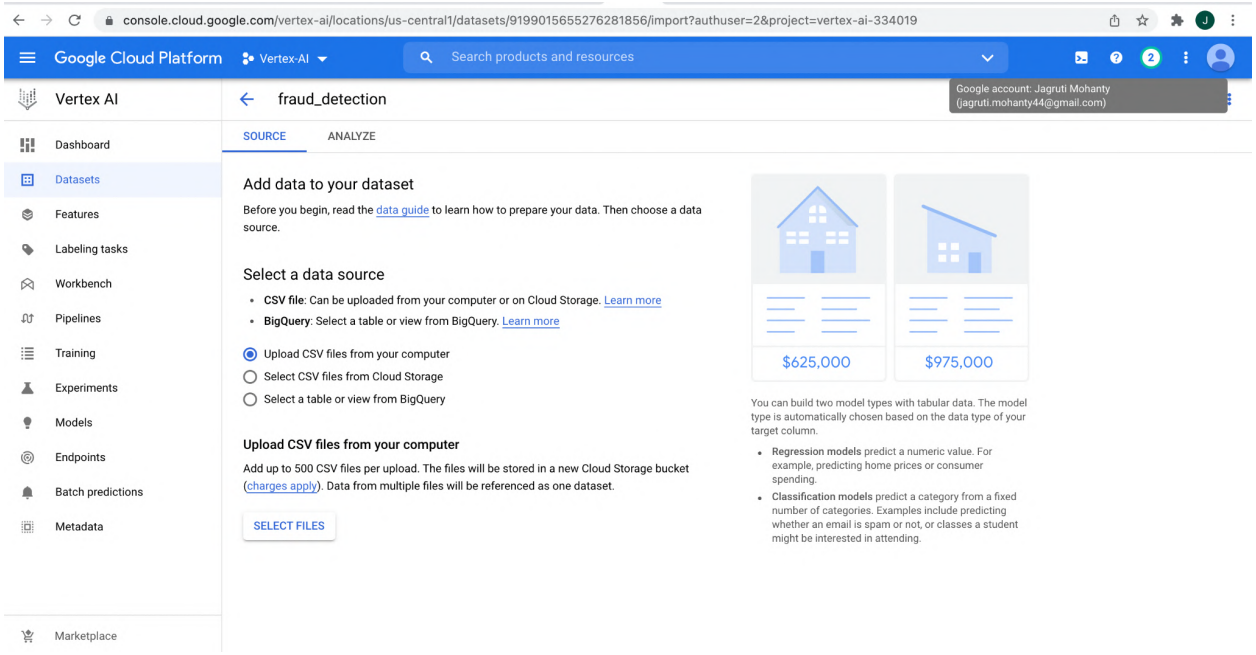


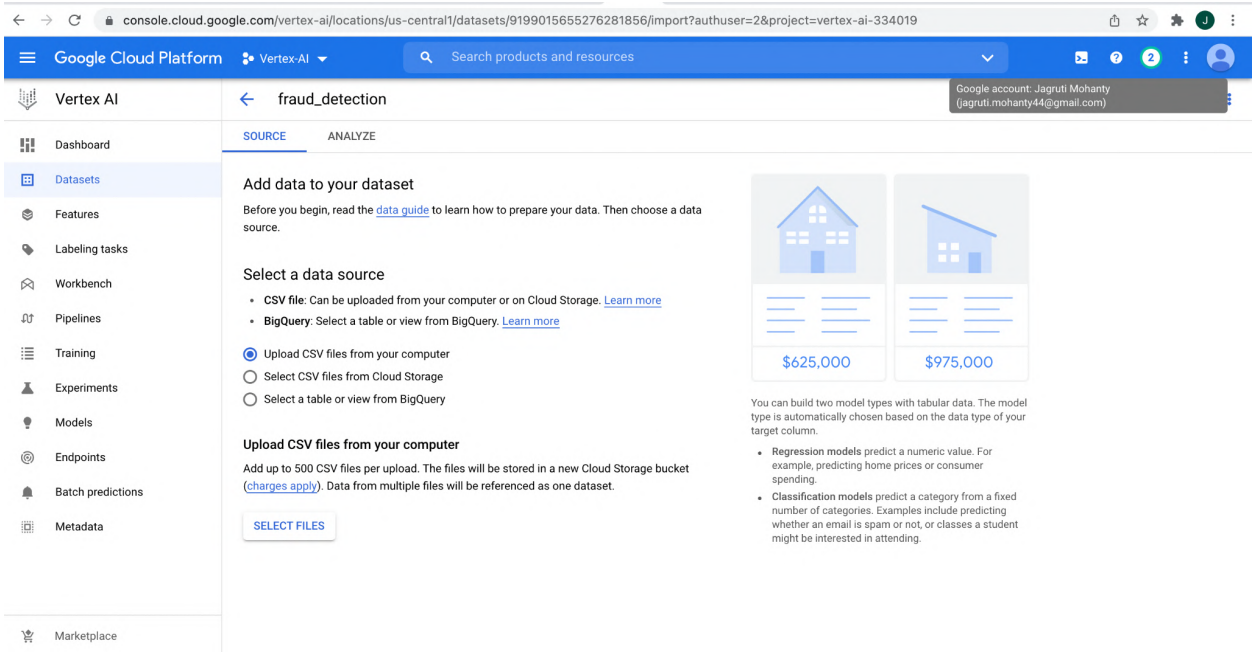


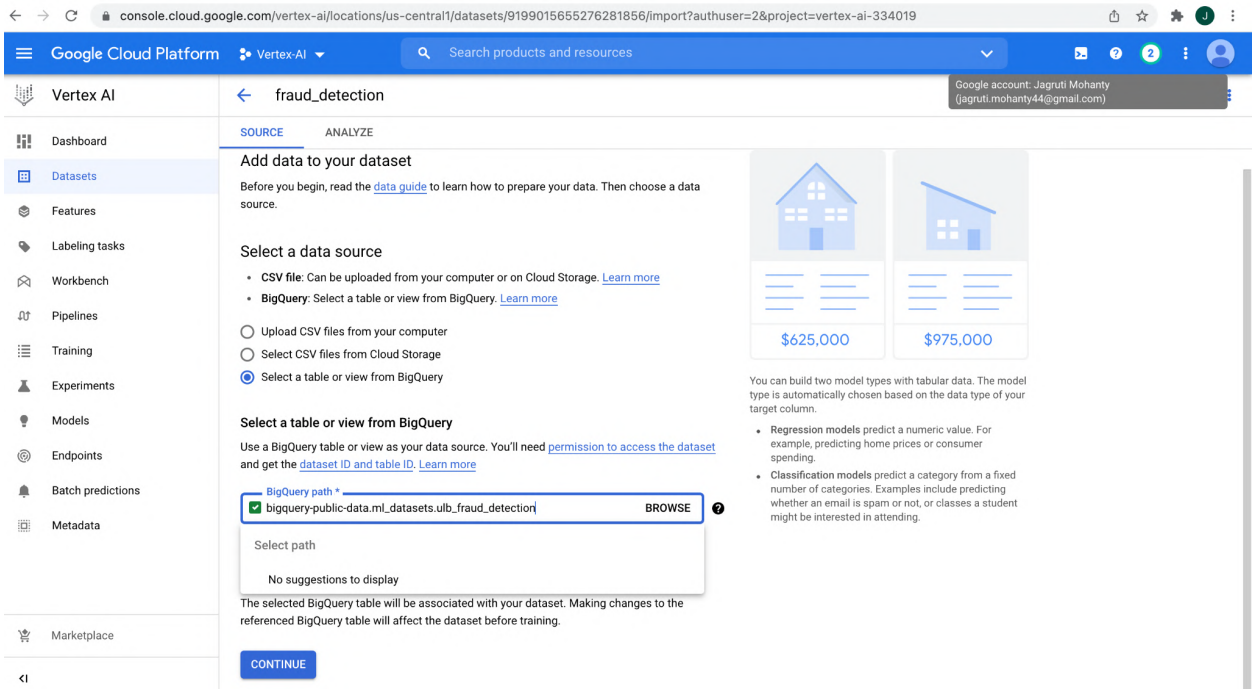


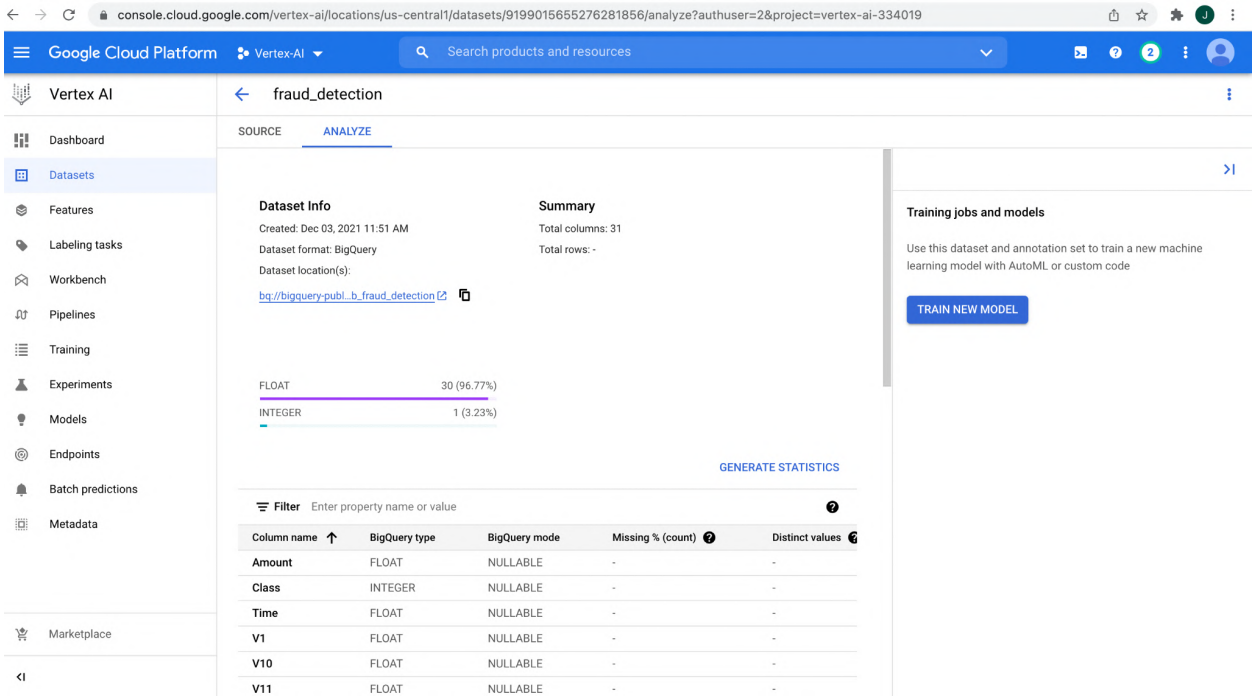
b)

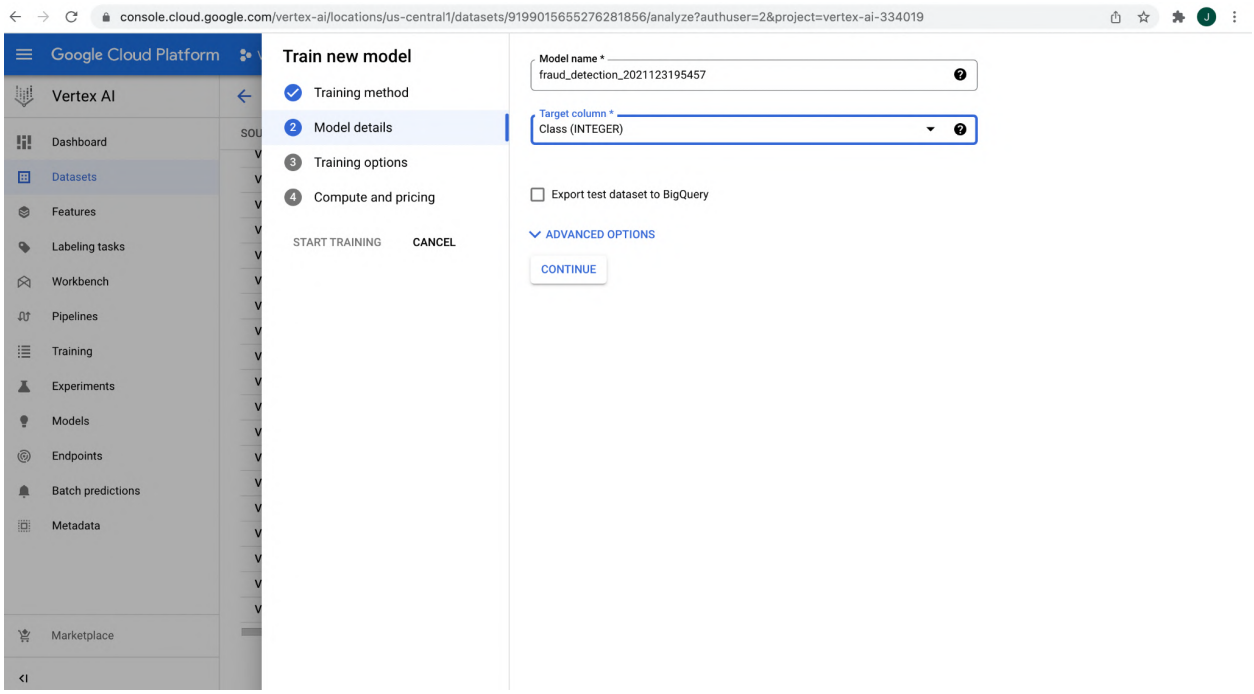
https://codelabs.developers.google.com/vertex-automl-tabular#0

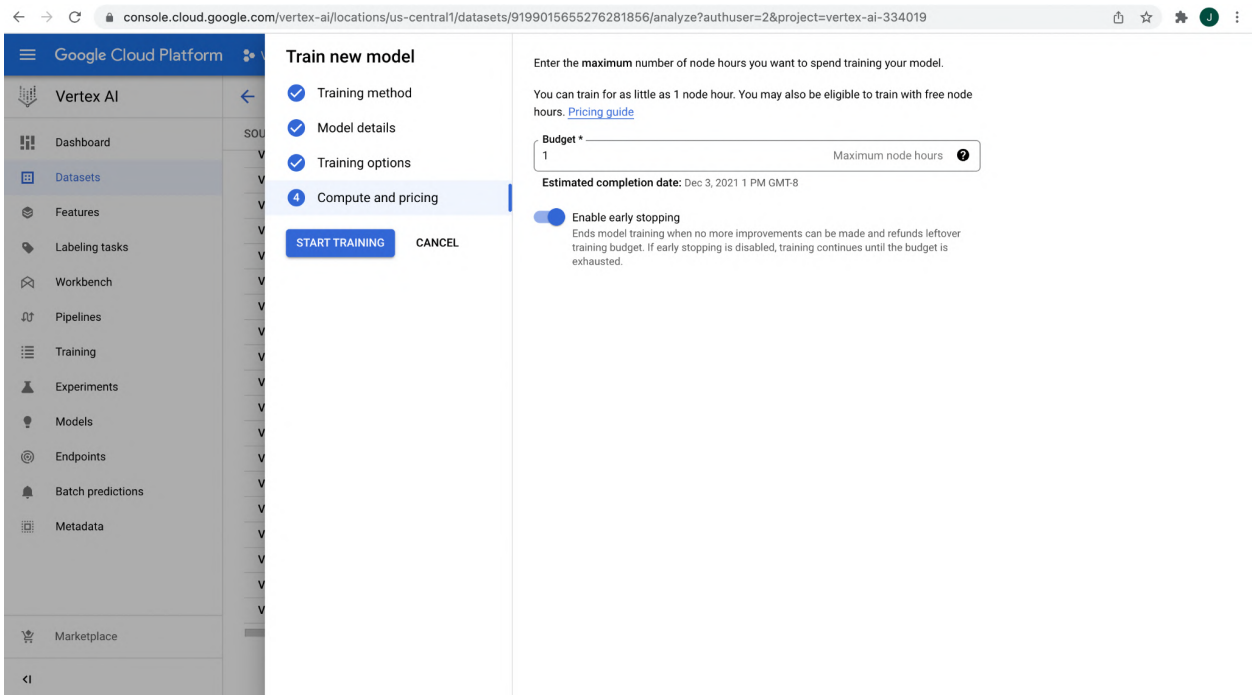

console.cloud.google.com/vertex-ai/locations/us-central1/datasets/9199015655276281856/import?authuser=2&project=vertex-ai-334019
Google Cloud Platform
Vertex-AI
Search products and resources
Google account: Jagruti Mohanty
(jagruti.mohanty44@gmail.com)
Vertex AI
fraud_detection
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
SOURCE
ANALYZE
Add data to your dataset
Before you begin, read the data guide to learn how to prepare your data. Then choose a data source.
Select a data source
CSV file: Can be uploaded from your computer or on Cloud Storage. Learn more
BigQuery: Select a table or view from BigQuery. Learn more
Upload CSV files from your computer
Select CSV files from Cloud Storage
Select a table or view from BigQuery
Upload CSV files from your computer
Add up to 500 CSV files per upload. The files will be stored in a new Cloud Storage bucket (charges apply). Data from multiple files will be referenced as one dataset.
SELECT FILES
$625,000
$975,000
You can build two model types with tabular data. The model type is automatically chosen based on the data type of your target column.
Regression models predict a numeric value. For example, predicting home prices or consumer spending.
Classification models predict a category from a fixed number of categories. Examples include predicting whether an email is spam or not, or classes a student might be interested in attending.

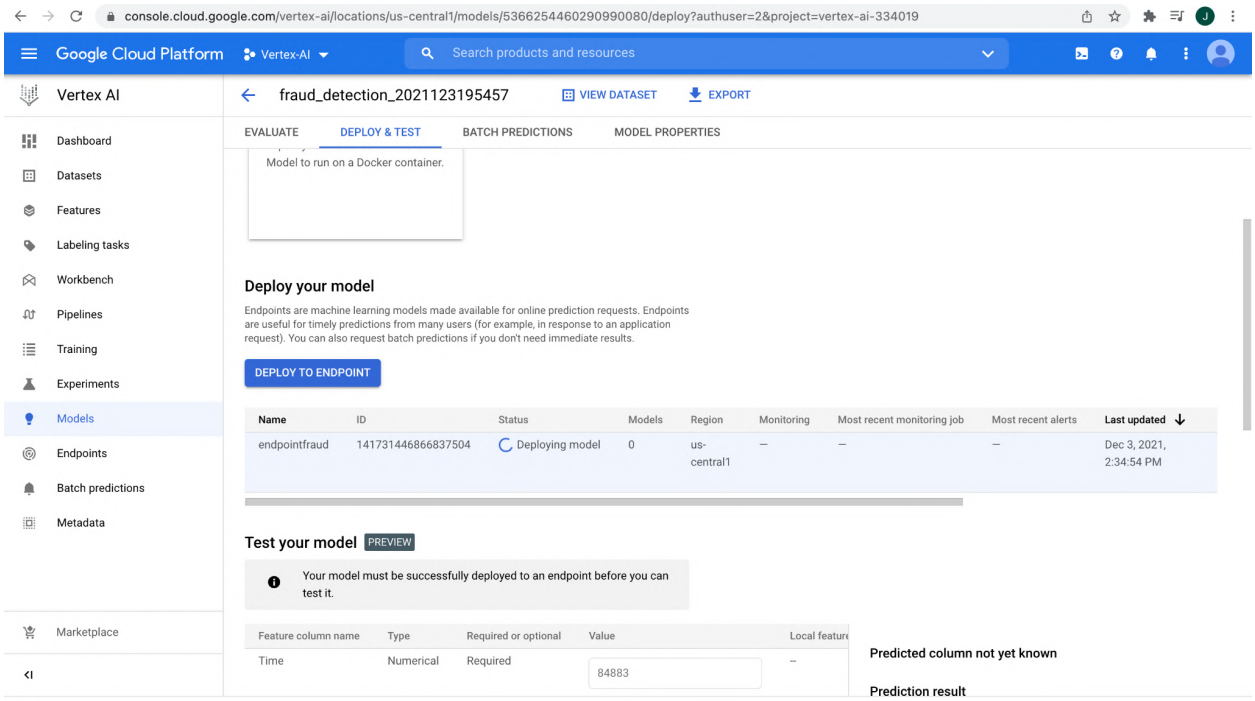

console.cloud.google.com/vertex-ai/locations/us-central1/datasets/9199015655276281856/import?authuser=2&project=vertex-ai-334019
Google Cloud Platform
Vertex-AI
Search products and resources
Google account: Jagruti Mohanty (jagruti.mohanty44@gmail.com)
Vertex AI
fraud_detection
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
SOURCE
ANALYZE
Add data to your dataset
Before you begin, read the data guide to learn how to prepare your data. Then choose a data source.
Select a data source
CSV file: Can be uploaded from your computer or on Cloud Storage. Learn more
BigQuery: Select a table or view from BigQuery. Learn more
Upload CSV files from your computer
Select CSV files from Cloud Storage
Select a table or view from BigQuery
Upload CSV files from your computer
Add up to 500 CSV files per upload. The files will be stored in a new Cloud Storage bucket (charges apply). Data from multiple files will be referenced as one dataset.
SELECT FILES
$625,000
$975,000
You can build two model types with tabular data. The model type is automatically chosen based on the data type of your target column.
Regression models predict a numeric value. For example, predicting home prices or consumer spending.
Classification models predict a category from a fixed number of categories. Examples include predicting whether an email is spam or not, or classes a student might be interested in attending.


console.cloud.google.com/vertex-ai/locations/us-central1/datasets/9199015655276281856/import?authuser=2&project=vertex-ai-334019
Google Cloud Platform
Vertex-AI
Search products and resources
Google account: Jagruti Mohanty (jagruti.mohanty44@gmail.com)
Vertex AI
fraud_detection
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
SOURCE
ANALYZE
Add data to your dataset
Before you begin, read the data guide to learn how to prepare your data. Then choose a data source.
Select a data source
CSV file: Can be uploaded from your computer or on Cloud Storage. Learn more
BigQuery: Select a table or view from BigQuery. Learn more
Upload CSV files from your computer
Select CSV files from Cloud Storage
Select a table or view from BigQuery
Select a table or view from BigQuery
Use a BigQuery table or view as your data source. You'll need permission to access the dataset and get the dataset ID and table ID. Learn more
BigQuery path *
bigquery-public-data.ml_datasets.ulb_fraud_detection
BROWSE
Select path
No suggestions to display
The selected BigQuery table will be associated with your dataset. Making changes to the referenced BigQuery table will affect the dataset before training.
CONTINUE
$625,000
$975,000
You can build two model types with tabular data. The model type is automatically chosen based on the data type of your target column.
Regression models predict a numeric value. For example, predicting home prices or consumer spending.
Classification models predict a category from a fixed number of categories. Examples include predicting whether an email is spam or not, or classes a student might be interested in attending.


console.cloud.google.com/vertex-ai/locations/us-central1/datasets/9199015655276281856/analyze?authuser=2&project=vertex-ai-334019
Google Cloud Platform
Vertex-AI
Search products and resources
Vertex AI
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
fraud_detection
SOURCE
ANALYZE
Dataset Info
Created: Dec 03, 2021 11:51 AM
Dataset format: BigQuery
Dataset location(s):
bq://bigquery-publ...b_fraud_detection
Summary
Total columns: 31
Total rows: -
FLOAT 30 (96.77%)
INTEGER 1 (3.23%)
GENERATE STATISTICS
Filter Enter property name or value
Column name
BigQuery type
BigQuery mode
Missing % (count)
Distinct values
Amount FLOAT NULLABLE - -
Class INTEGER NULLABLE - -
Time FLOAT NULLABLE - -
V1 FLOAT NULLABLE - -
V10 FLOAT NULLABLE - -
V11 FLOAT NULLABLE - -
Training jobs and models
Use this dataset and annotation set to train a new machine learning model with AutoML or custom code
TRAIN NEW MODEL


console.cloud.google.com/vertex-ai/locations/us-central1/datasets/9199015655276281856/analyze?authuser=2&project=vertex-ai-334019
Google Cloud Platform
Vertex AI
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
Train new model
Training method
2 Model details
3 Training options
4 Compute and pricing
START TRAINING
CANCEL
Model name *
fraud_detection_2021123195457
Target column *
Class (INTEGER)
Export test dataset to BigQuery
ADVANCED OPTIONS
CONTINUE

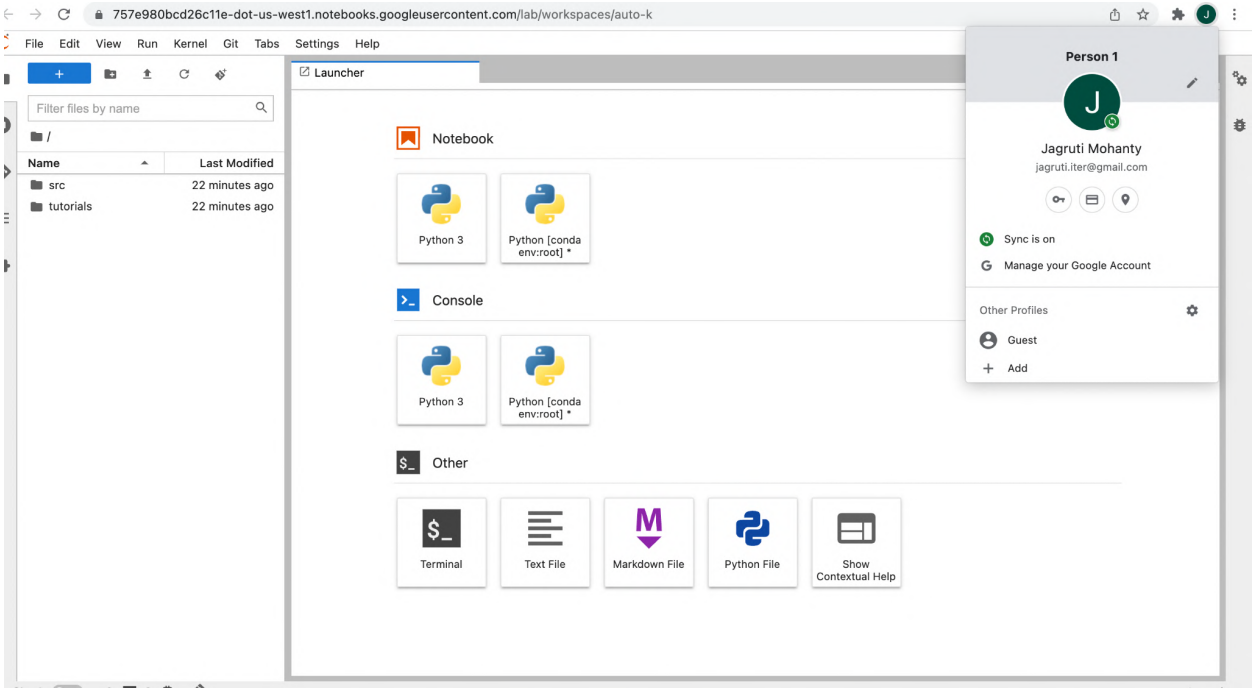

console.cloud.google.com/vertex-ai/locations/us-central1/datasets/9199015655276281856/analyze?authuser=2&project=vertex-ai-334019
Google Cloud Platform
Vertex AI
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
Train new model
Training method
Model details
Training options
4 Compute and pricing
START TRAINING
CANCEL
Enter the maximum number of node hours you want to spend training your model.
You can train for as little as 1 node hour. You may also be eligible to train with free node hours. Pricing guide
Budget *
1
Maximum node hours
Estimated completion date: Dec 3, 2021 1 PM GMT-8
Enable early stopping
Ends model training when no more improvements can be made and refunds leftover training budget. If early stopping is disabled, training continues until the budget is exhausted.

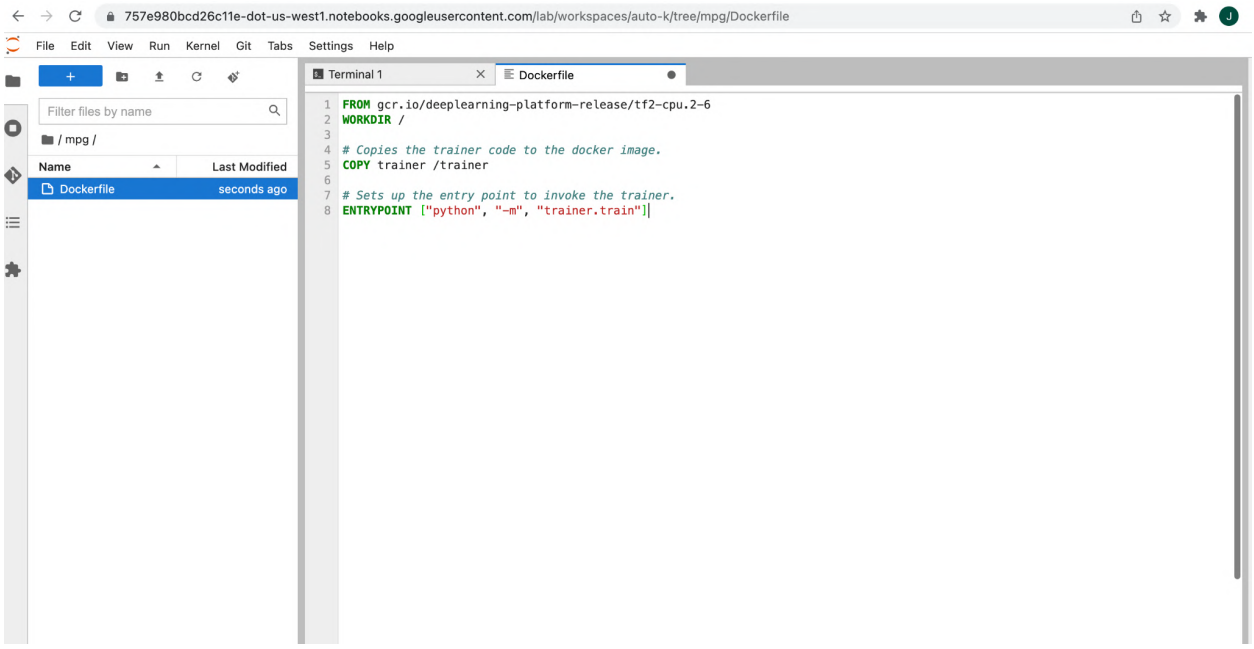


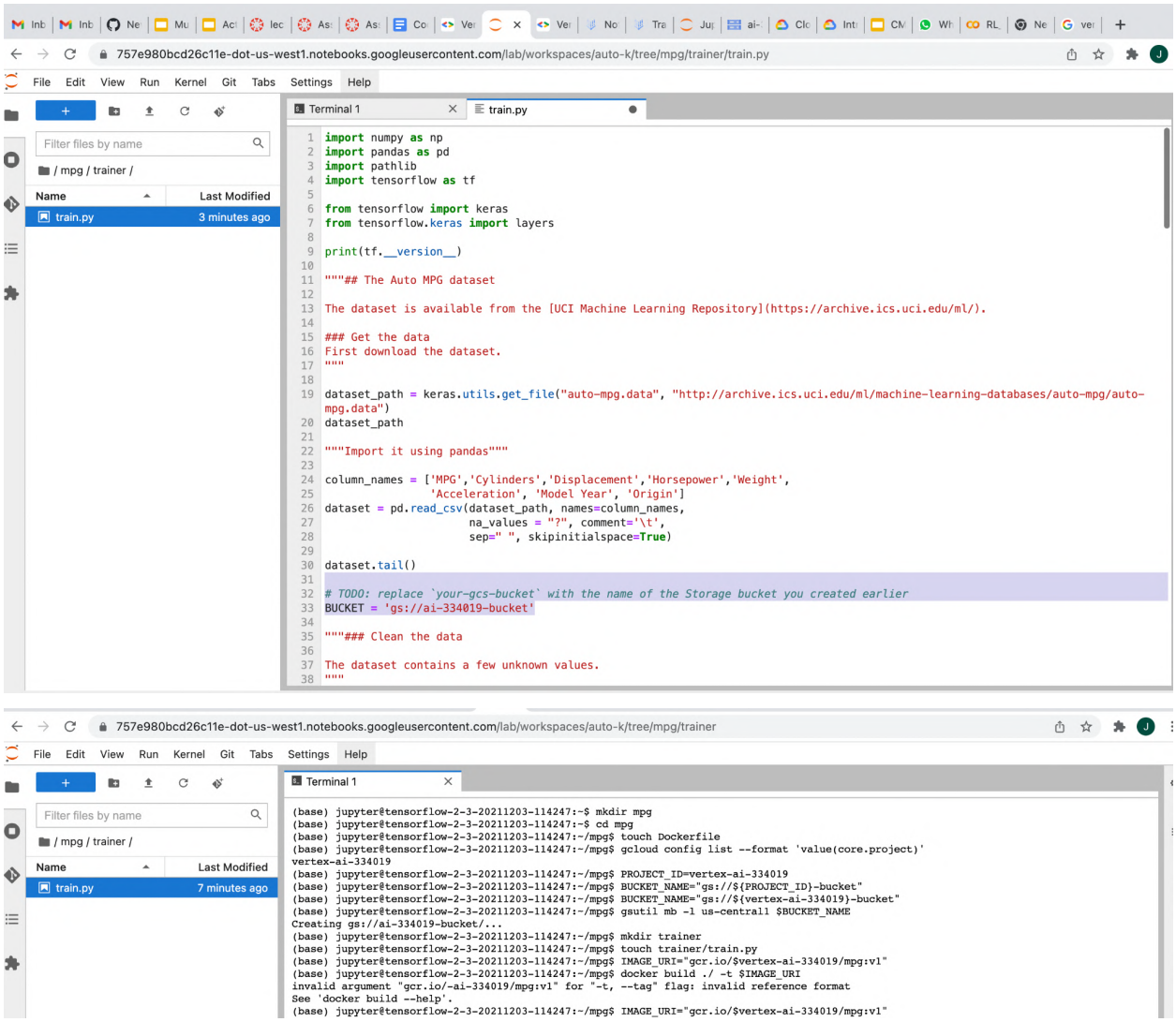
c)

https://codelabs.developers.google.com/vertex_custom_training_prediction#0

(Links to an external site.)

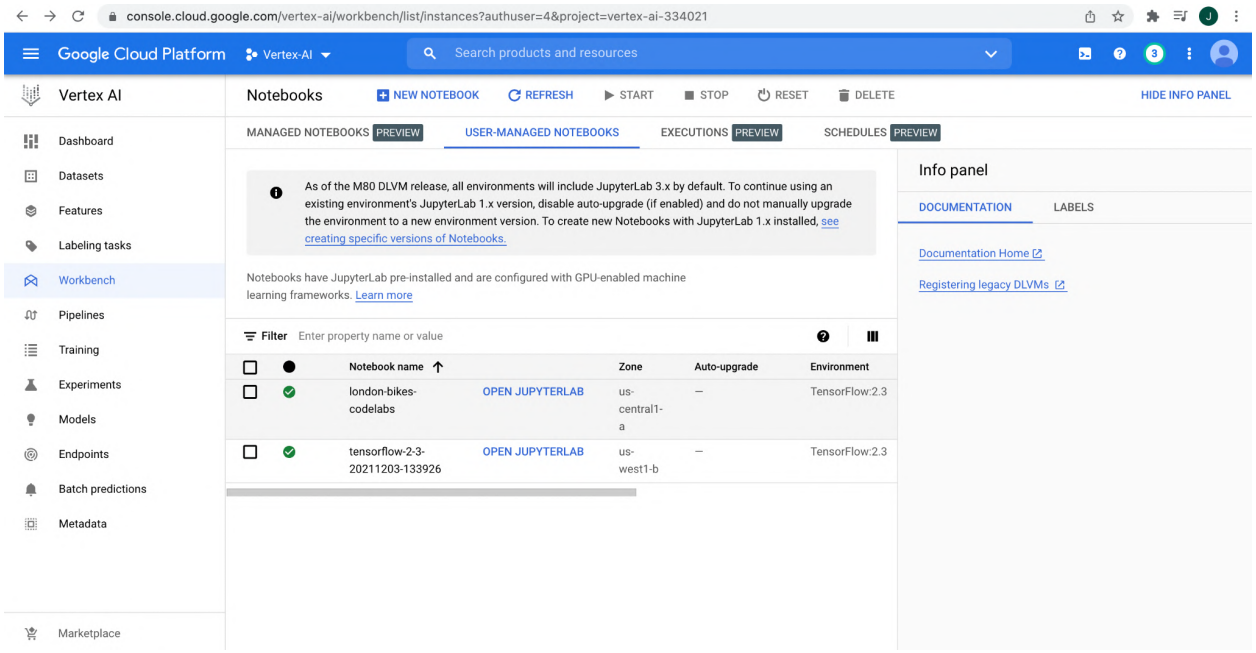

757e980bcd26c11e-dot-us-west1.notebooks.googleusercontent.com/lab/workspaces/auto-k
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/
Name
Last Modified
src
22 minutes ago
tutorials
22 minutes ago
Launcher
Notebook
Python 3
Python [conda env:root] *
Console
Python 3
Python [conda env:root] *
Other
Terminal
Text File
Markdown File
Python File
Show Contextual Help
Person 1
Jagruti Mohanty
jagruti.iter@gmail.com
Sync is on
Manage your Google Account
Other Profiles
Guest
Add

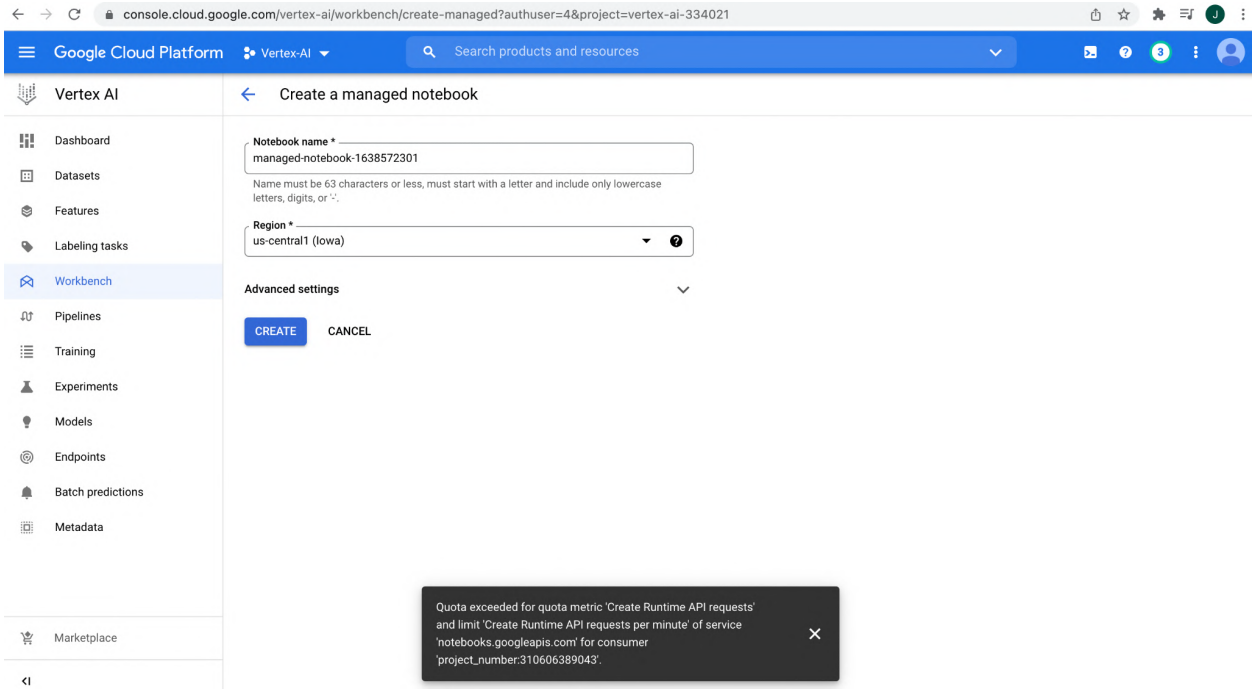

757e980bcd26c11e-dot-us-west1.notebooks.googleusercontent.com/lab/workspaces/auto-k/tree/mpg/Dockerfile
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ mpg /
Name
Last Modified
Dockerfile
seconds ago
Terminal 1
Dockerfile
1 FROM gcr.io/deeplearning-platform-release/tf2-cpu.2-6
2 WORKDIR /
3
4 # Copies the trainer code to the docker image.
5 COPY trainer /trainer
6
7 # Sets up the entry point to invoke the trainer.
8 ENTRYPOINT ["python", "-m", "trainer.train"]



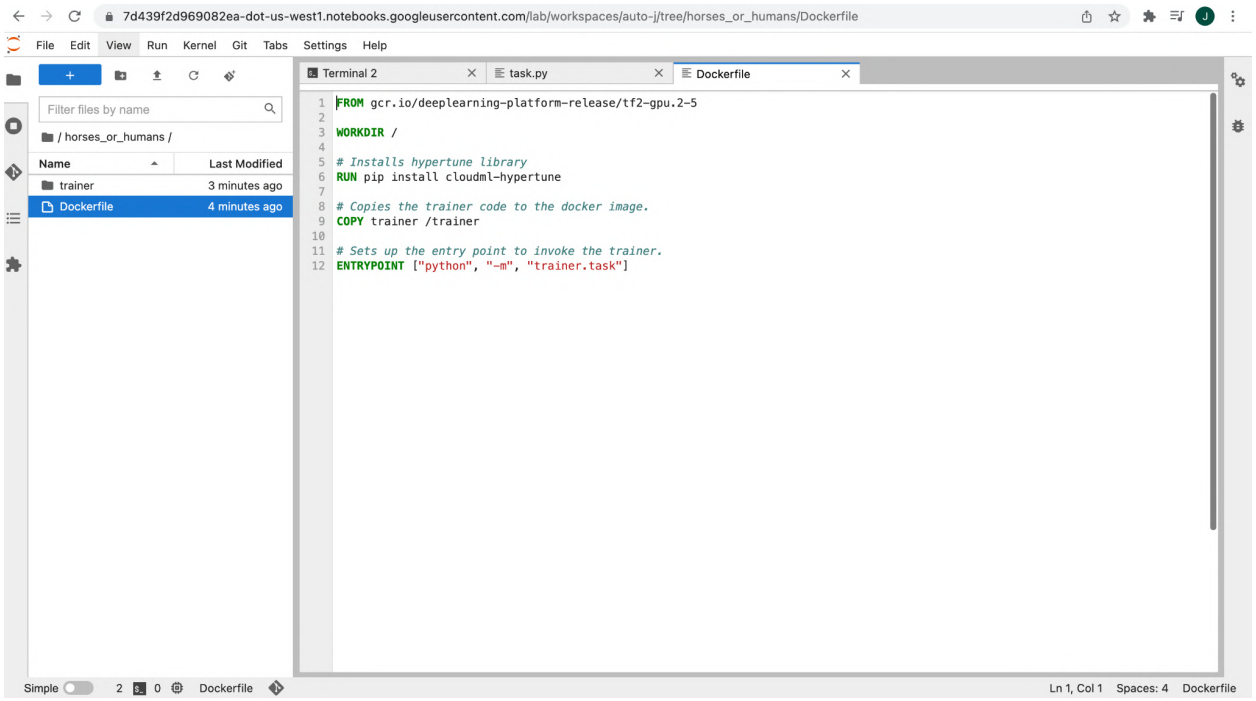
d) Big query :

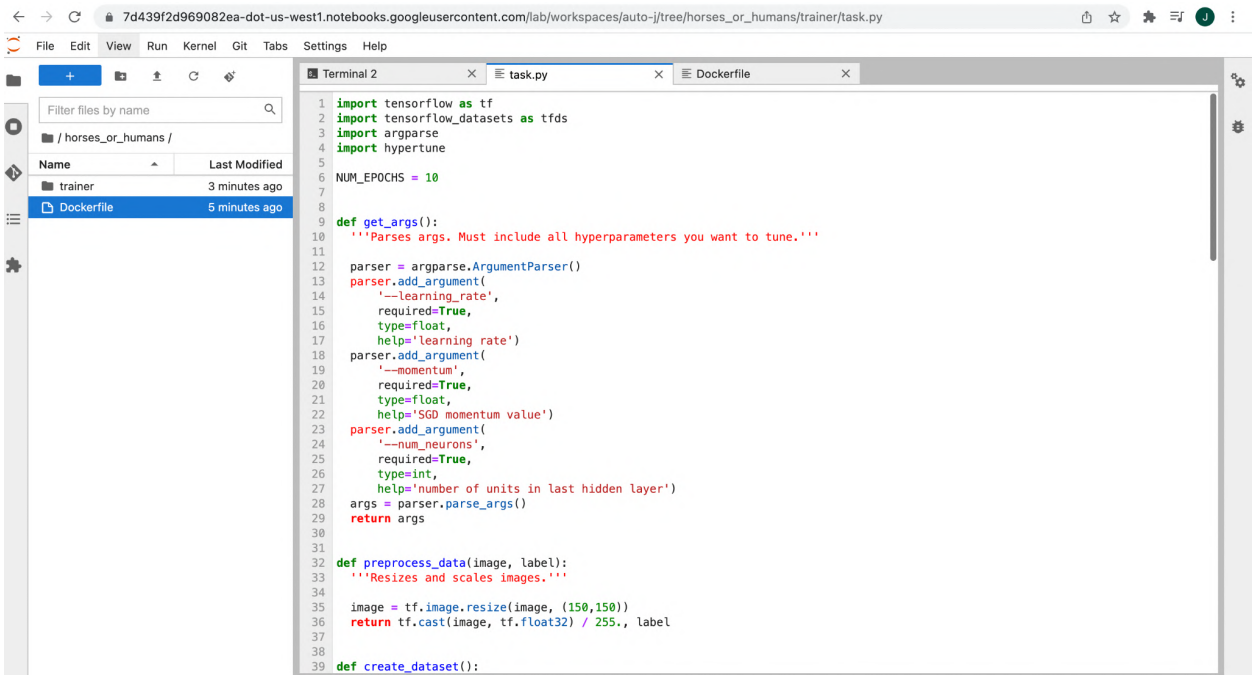
https://codelabs.developers.google.com/vertex-workbench-intro#0

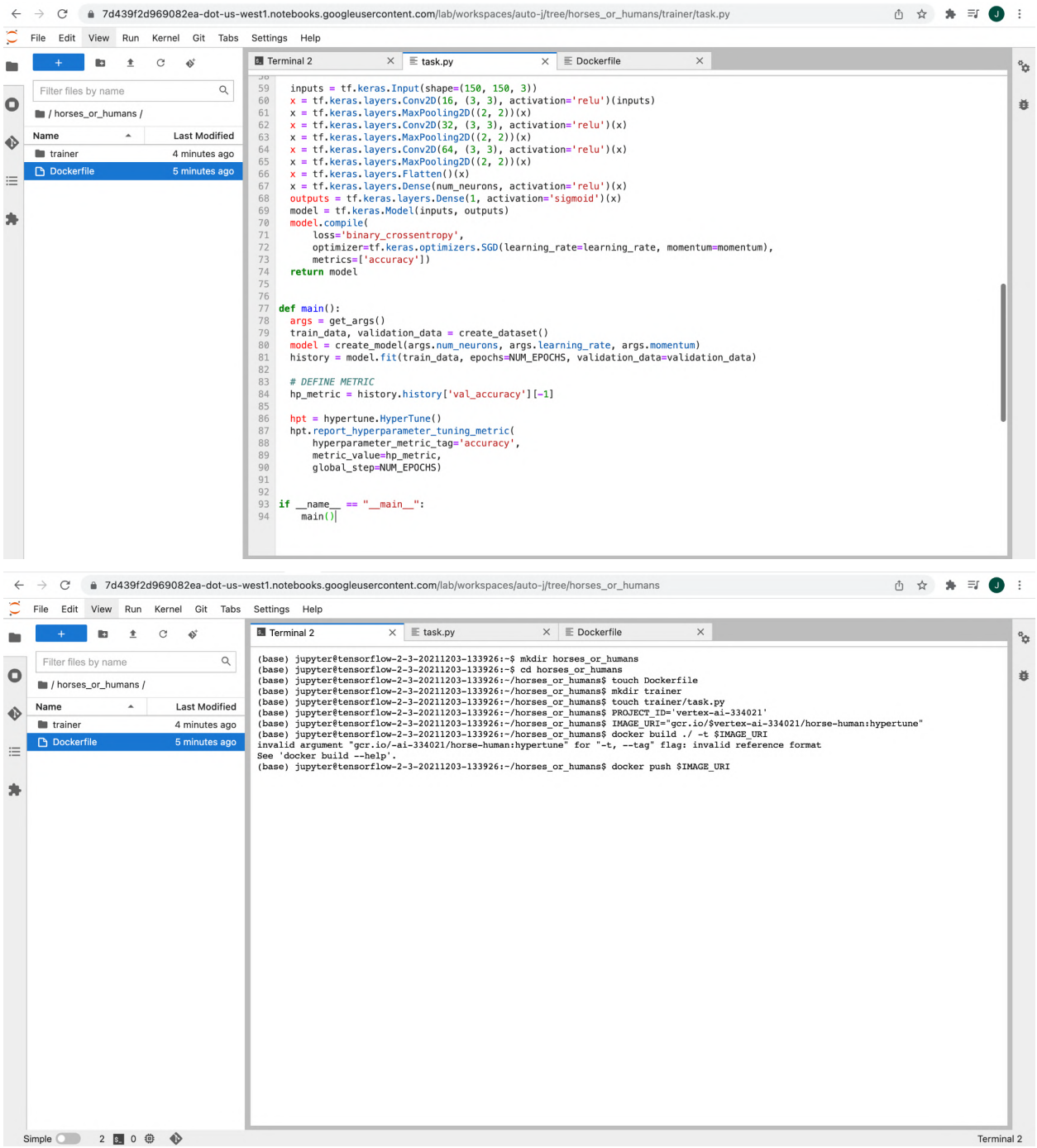




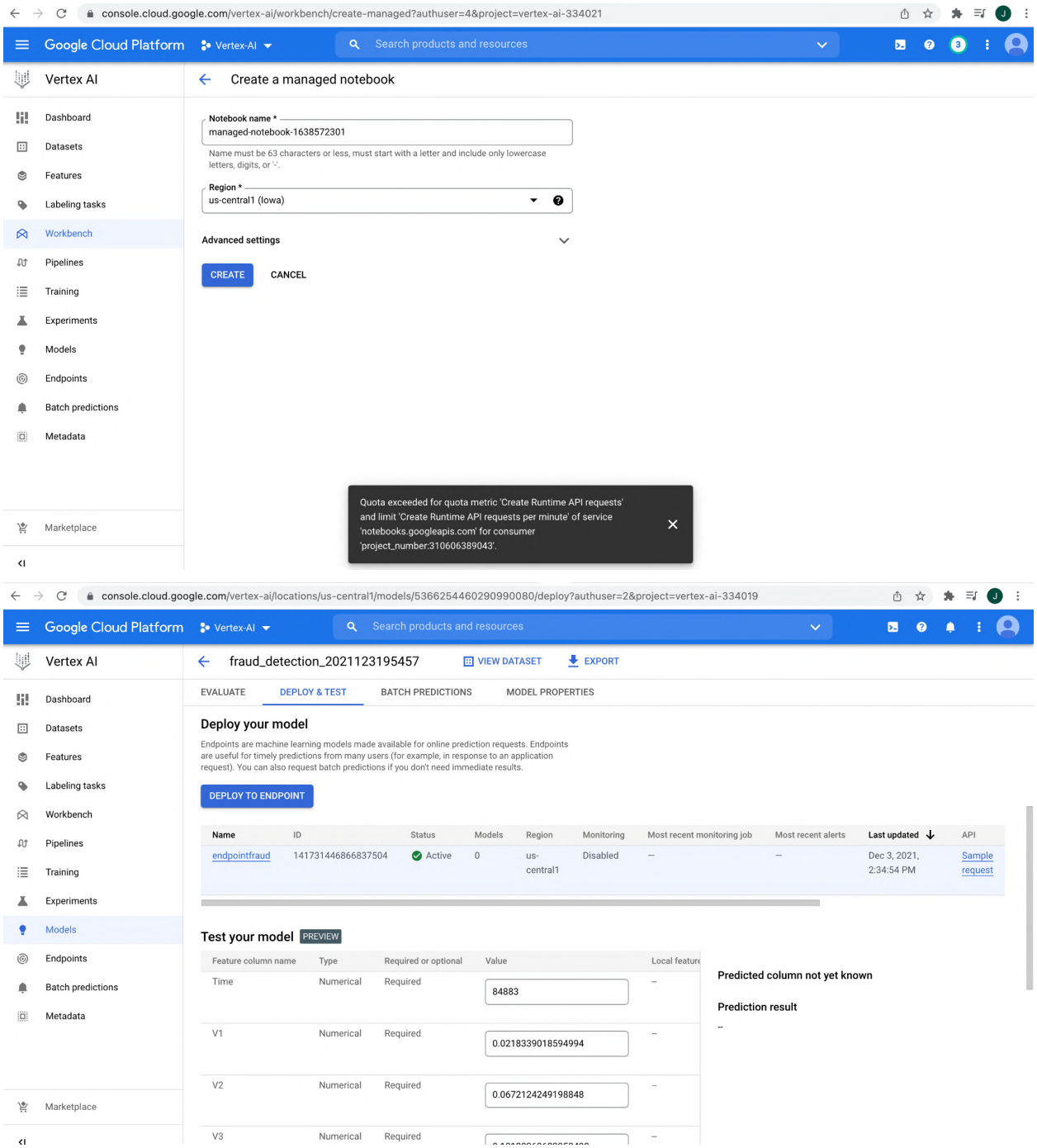
f) https://codelabs.developers.google.com/vertex_hyperparameter_tuning#0


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/workspaces/auto-j/tree/horses_or_humans/Dockerfile
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ horses_or_humans /
Name
Last Modified
trainer
3 minutes ago
Dockerfile
4 minutes ago
Terminal 2
task.py
Dockerfile
1 FROM gcr.io/deeplearning-platform-release/tf2-gpu.2-5
2
3 WORKDIR /
4
5 # Installs hypertune library
6 RUN pip install cloudml-hypertune
7
8 # Copies the trainer code to the docker image.
9 COPY trainer /trainer
10
11 # Sets up the entry point to invoke the trainer.
12 ENTRYPOINT ["python", "-m", "trainer.task"]
Simple
2
0
Dockerfile
Ln 1, Col 1
Spaces: 4
Dockerfile


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/workspaces/auto-j/tree/horses_or_humans/trainer/task.py
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ horses_or_humans /
Name
Last Modified
trainer
3 minutes ago
Dockerfile
5 minutes ago
Terminal 2
task.py
Dockerfile
1 import tensorflow as tf
2 import tensorflow_datasets as tfds
3 import argparse
4 import hypertune
5
6 NUM_EPOCHS = 10
7
8
9 def get_args():
10 '''Parses args. Must include all hyperparameters you want to tune.'''
11
12 parser = argparse.ArgumentParser()
13 parser.add_argument(
14 '--learning_rate',
15 required=True,
16 type=float,
17 help='learning rate')
18 parser.add_argument(
19 '--momentum',
20 required=True,
21 type=float,
22 help='SGD momentum value')
23 parser.add_argument(
24 '--num_neurons',
25 required=True,
26 type=int,
27 help='number of units in last hidden layer')
28 args = parser.parse_args()
29 return args
30
31
32 def preprocess_data(image, label):
33 '''Resizes and scales images.'''
34
35 image = tf.image.resize(image, (150,150))
36 return tf.cast(image, tf.float32) / 255., label
37
38
39 def create_dataset():

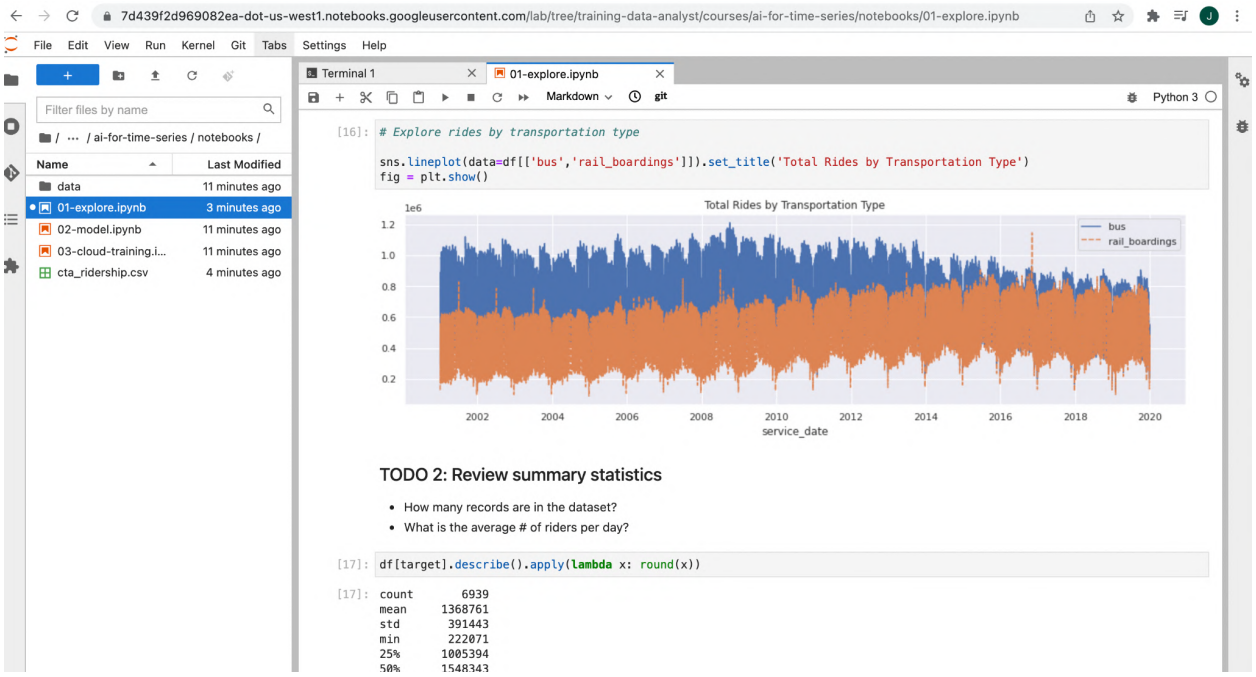


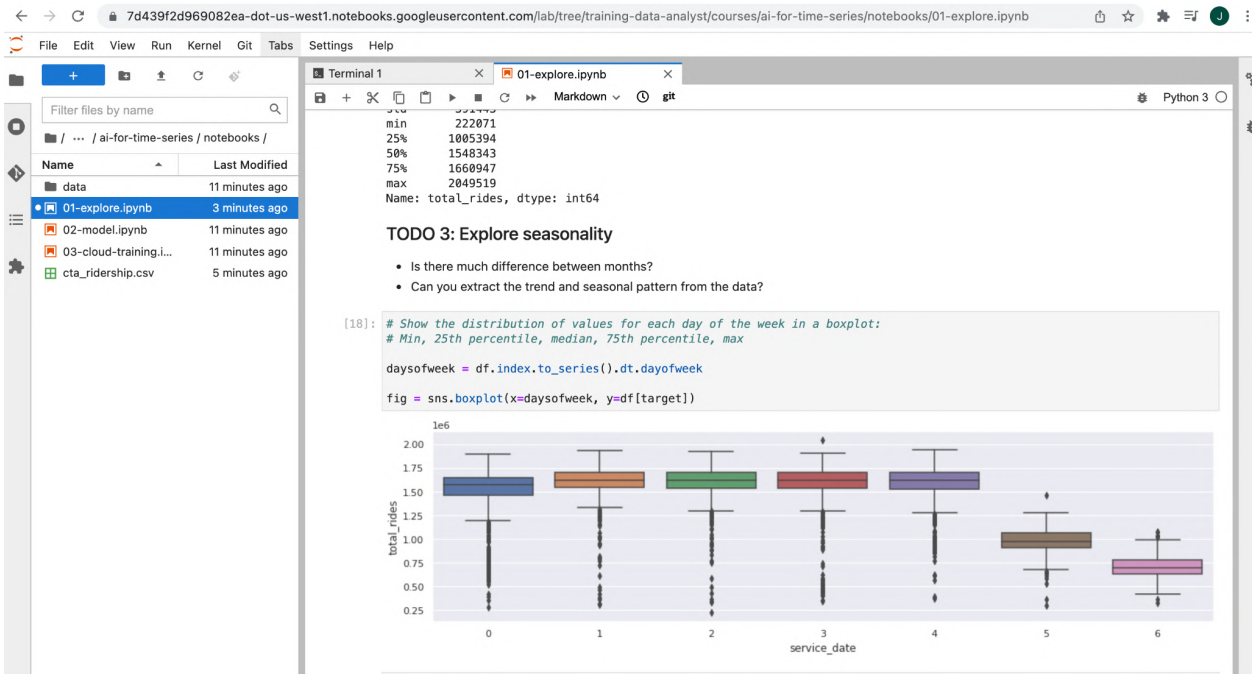
g) https://codelabs.developers.google.com/vertex-mlmd-pipelines#0

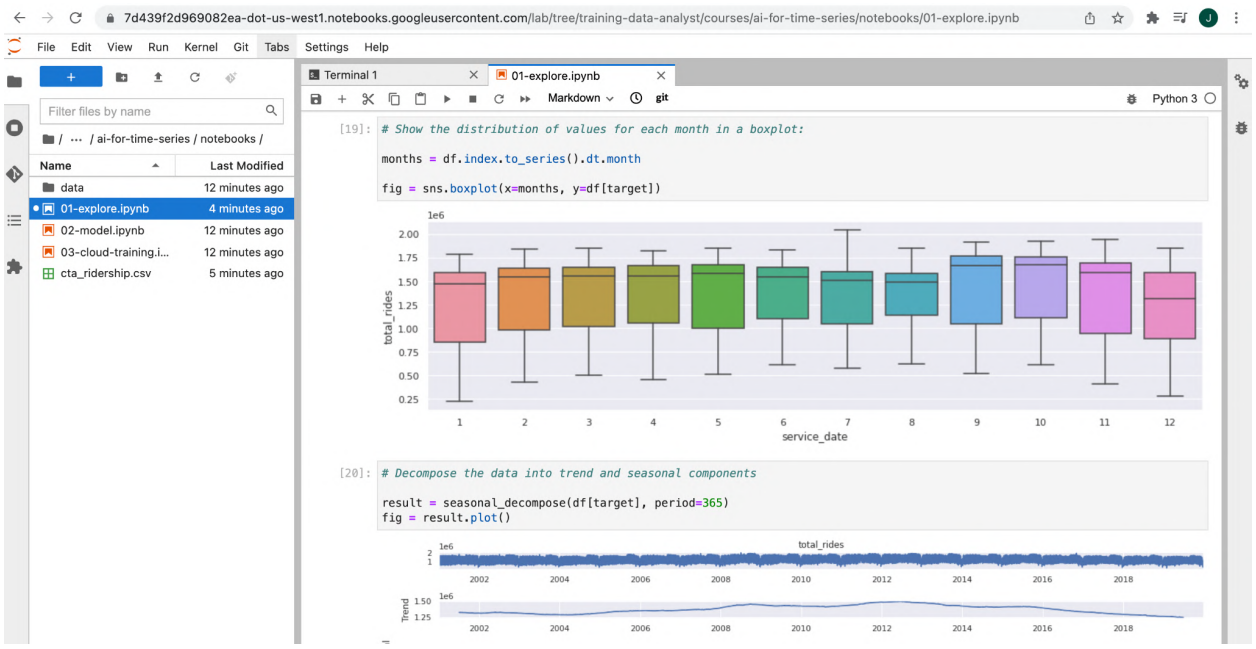


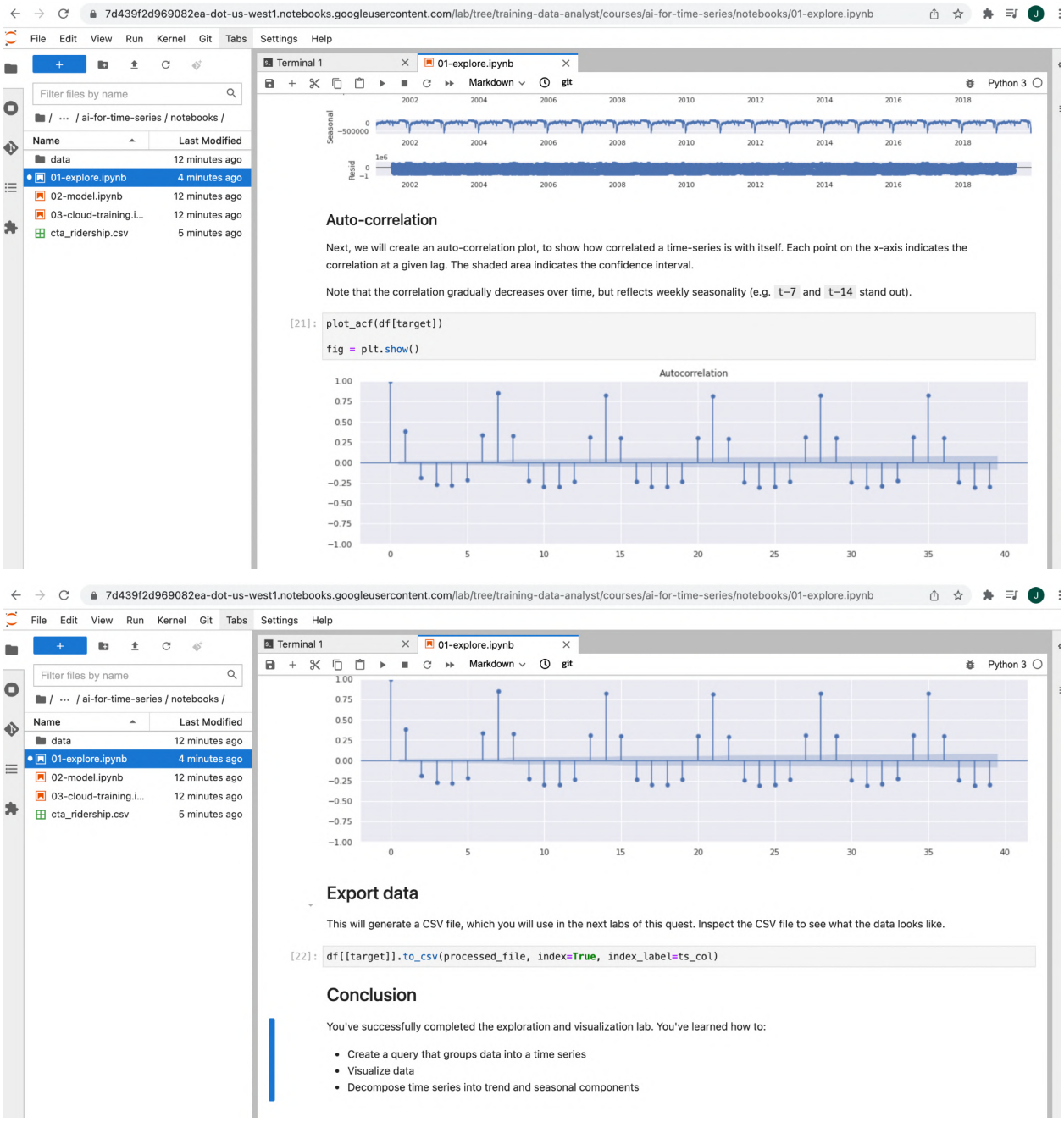
h) https://codelabs.developers.google.com/codelabs/automl-forecasting-with-vertex-ai#0


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 11 minutes ago
01-explore.ipynb 3 minutes ago
02-model.ipynb 11 minutes ago
03-cloud-training.i... 11 minutes ago
cta_ridership.csv 4 minutes ago
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
total_rides
service_date
[15]:
# 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()
Total Rides by Day Type
day_type
U
W
A
Simple
Python 3 | Idle
Mode: Command
Ln 1, Col 1
01-explore.ipynb

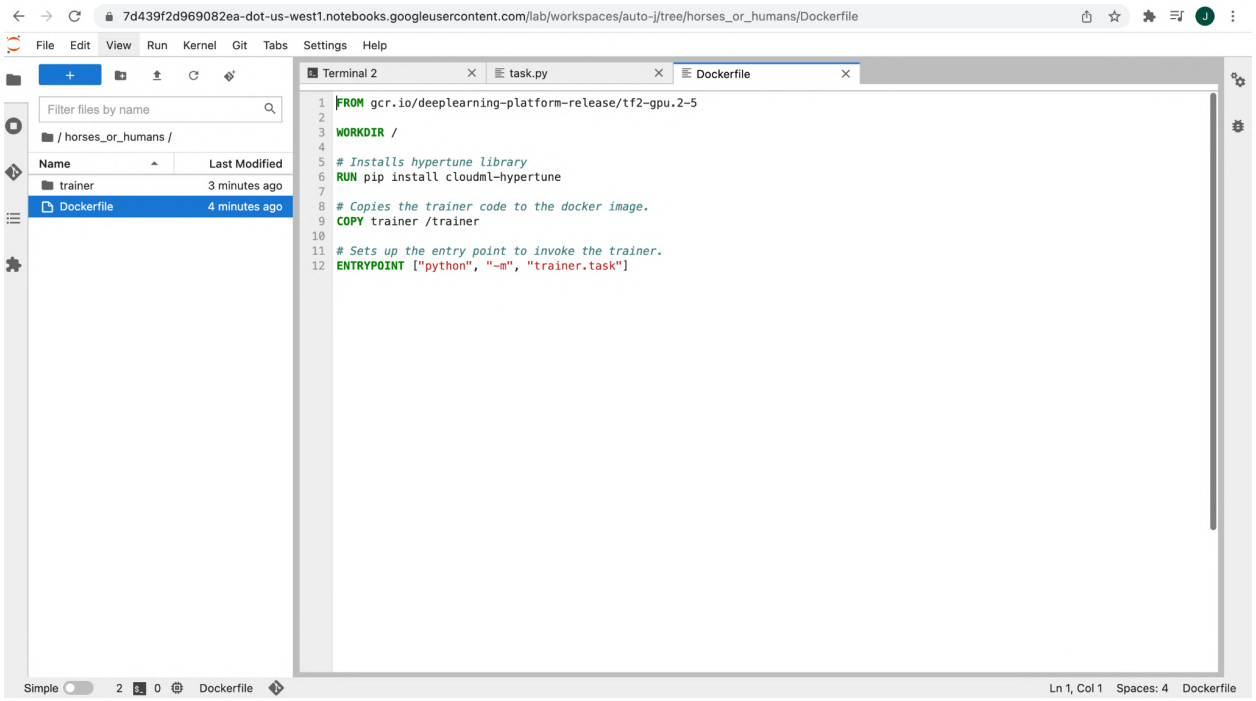

7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 11 minutes ago
01-explore.ipynb 3 minutes ago
02-model.ipynb 11 minutes ago
03-cloud-training.i... 11 minutes ago
cta_ridership.csv 4 minutes ago
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
[16]:
# Explore rides by transportation type
sns.lineplot(data=df[['bus','rail_boardings']]).set_title('Total Rides by Transportation Type')
fig = plt.show()
Total Rides by Transportation Type
bus
rail_boardings
service_date
TODO 2: Review summary statistics
How many records are in the dataset?
What is the average # of riders per day?
[17]:
df[target].describe().apply(lambda x: round(x))
[17]:
count 6939
mean 1368761
std 391443
min 222071
25% 1005394
50% 1548343

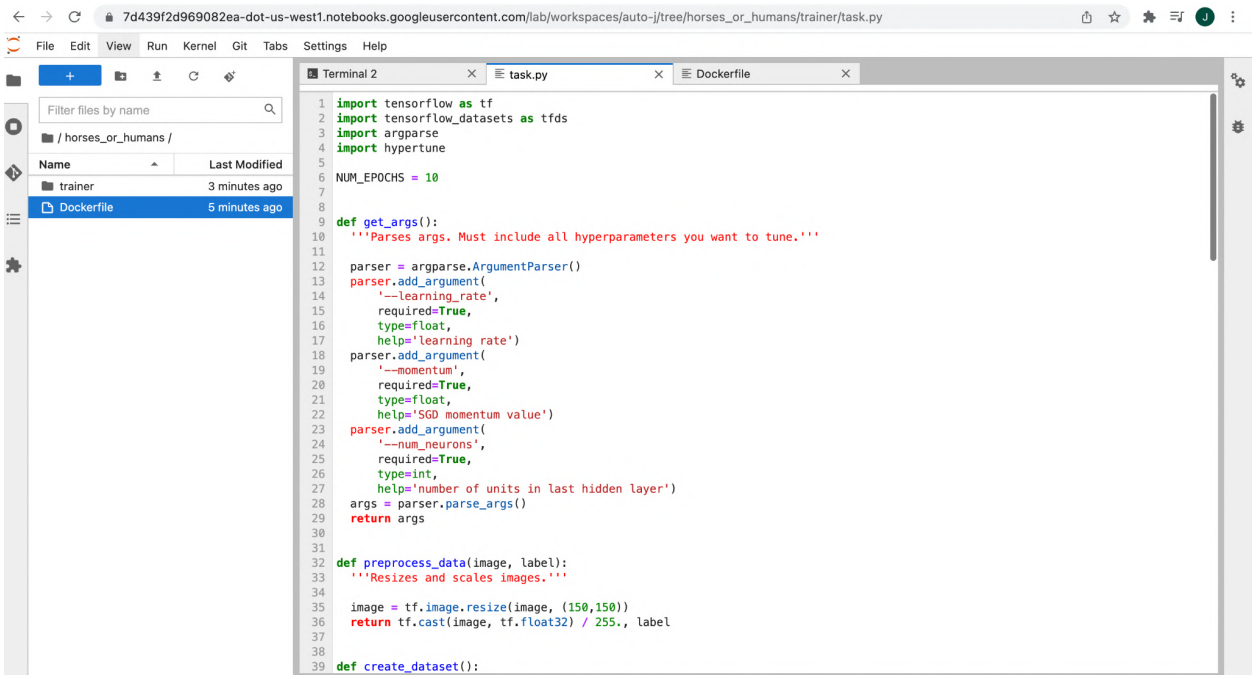

7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 11 minutes ago
01-explore.ipynb 3 minutes ago
02-model.ipynb 11 minutes ago
03-cloud-training.i... 11 minutes ago
cta_ridership.csv 5 minutes ago
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
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?
[18]:
# 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])
1e6
total_rides
service_date

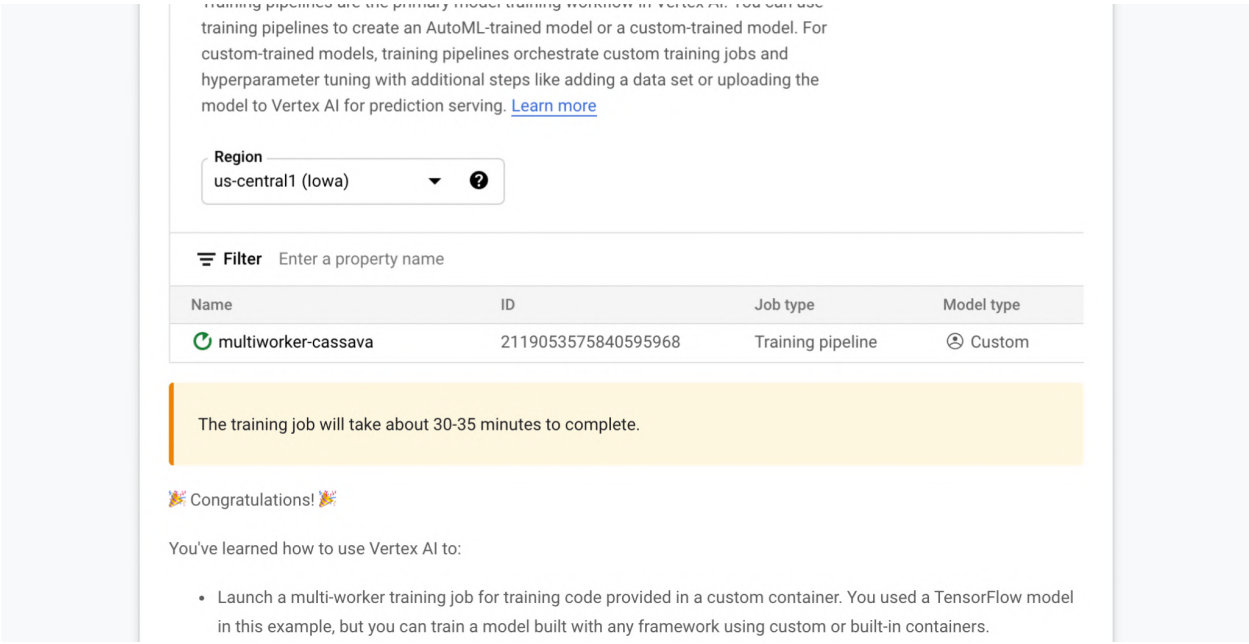

7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 12 minutes ago
01-explore.ipynb 4 minutes ago
02-model.ipynb 12 minutes ago
03-cloud-training.i... 12 minutes ago
cta_ridership.csv 5 minutes ago
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
[19]:
# 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])
1e6
total_rides
service_date
[20]:
# Decompose the data into trend and seasonal components
result = seasonal_decompose(df[target], period=365)
fig = result.plot()
total_rides
Trend



i) https://codelabs.developers.google.com/vertex_multiworker_training#0


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/workspaces/auto-j/tree/horses_or_humans/Dockerfile
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ horses_or_humans /
Name
Last Modified
trainer
3 minutes ago
Dockerfile
4 minutes ago
Terminal 2
task.py
Dockerfile
1 FROM gcr.io/deeplearning-platform-release/tf2-gpu.2-5
2
3 WORKDIR /
4
5 # Installs hypertune library
6 RUN pip install cloudml-hypertune
7
8 # Copies the trainer code to the docker image.
9 COPY trainer /trainer
10
11 # Sets up the entry point to invoke the trainer.
12 ENTRYPOINT ["python", "-m", "trainer.task"]
Simple
2
0
Dockerfile
Ln 1, Col 1
Spaces: 4
Dockerfile


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/workspaces/auto-j/tree/horses_or_humans/trainer/task.py
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ horses_or_humans /
Name
Last Modified
trainer
3 minutes ago
Dockerfile
5 minutes ago
Terminal 2
task.py
Dockerfile
1 import tensorflow as tf
2 import tensorflow_datasets as tfds
3 import argparse
4 import hypertune
5
6 NUM_EPOCHS = 10
7
8
9 def get_args():
10 '''Parses args. Must include all hyperparameters you want to tune.'''
11
12 parser = argparse.ArgumentParser()
13 parser.add_argument(
14 '--learning_rate',
15 required=True,
16 type=float,
17 help='learning rate')
18 parser.add_argument(
19 '--momentum',
20 required=True,
21 type=float,
22 help='SGD momentum value')
23 parser.add_argument(
24 '--num_neurons',
25 required=True,
26 type=int,
27 help='number of units in last hidden layer')
28 args = parser.parse_args()
29 return args
30
31
32 def preprocess_data(image, label):
33 '''Resizes and scales images.'''
34
35 image = tf.image.resize(image, (150,150))
36 return tf.cast(image, tf.float32) / 255., label
37
38
39 def create_dataset():



j) https://codelabs.developers.google.com/codelabs/bqml-vertex-prediction#0

training pipelines to create an AutoML-trained model or a custom-trained model. For custom-trained models, training pipelines orchestrate custom training jobs and hyperparameter tuning with additional steps like adding a data set or uploading the model to Vertex AI for prediction serving. Learn more

Region
us-central1 (Iowa)

Filter Enter a property name

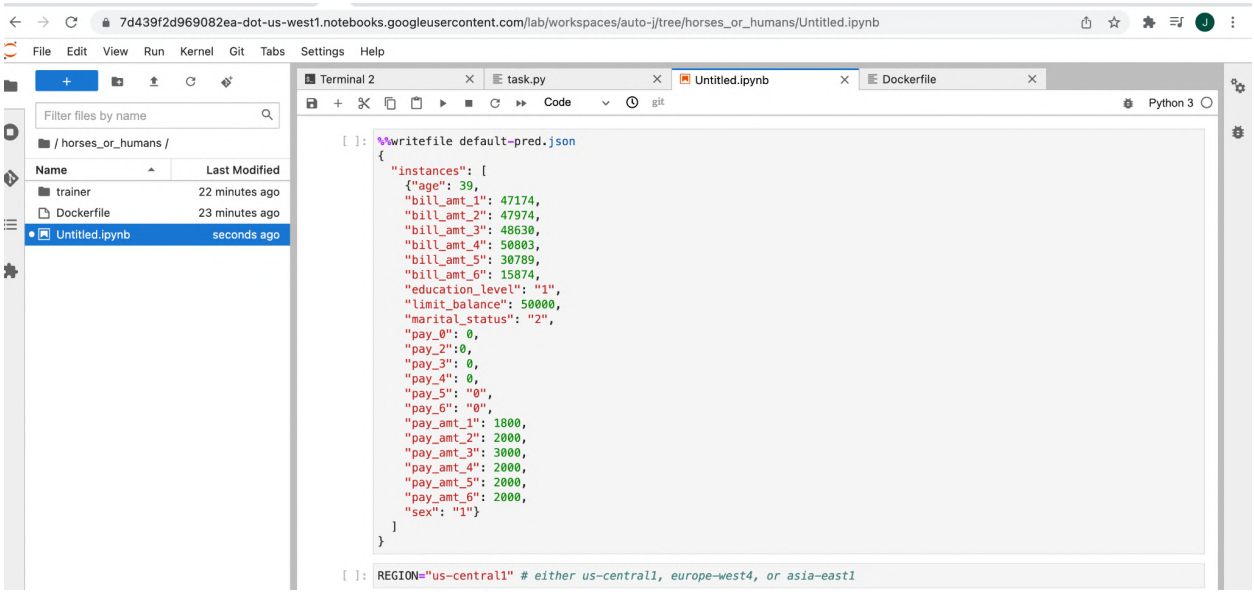
Name	ID	Job type	Model type
multiworker-cassava	2119053575840595968	Training pipeline	Custom

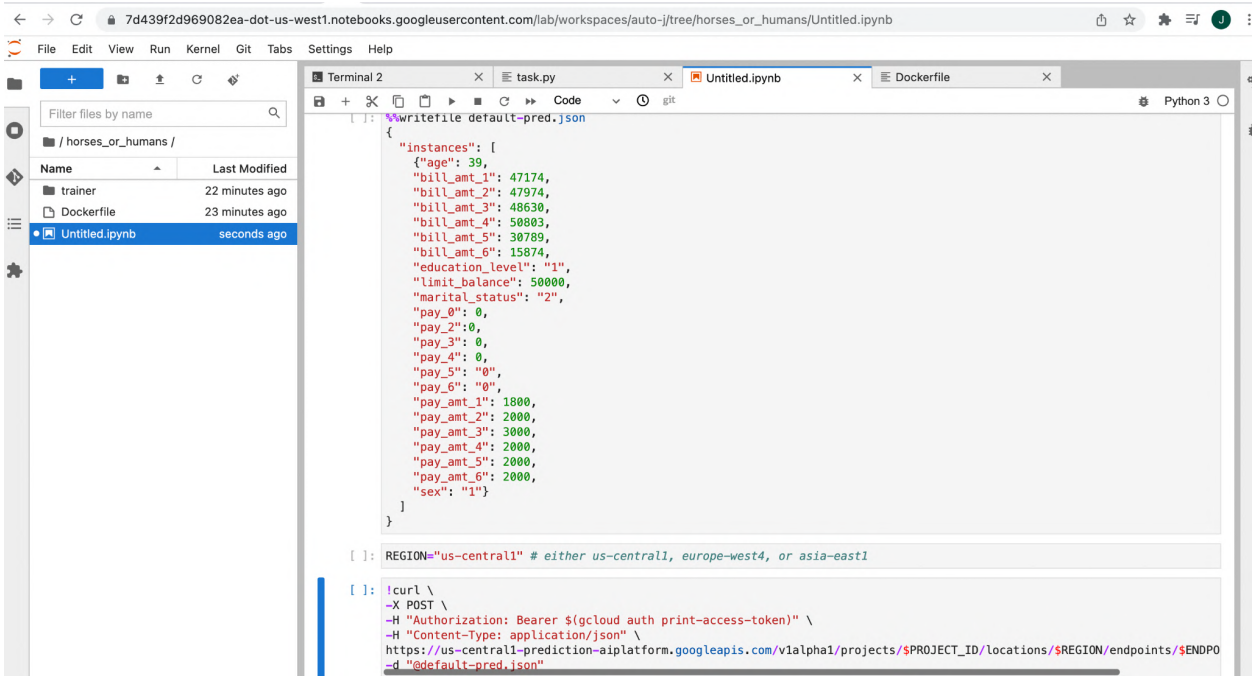
The training job will take about 30-35 minutes to complete.

🎉 Congratulations! 🎉

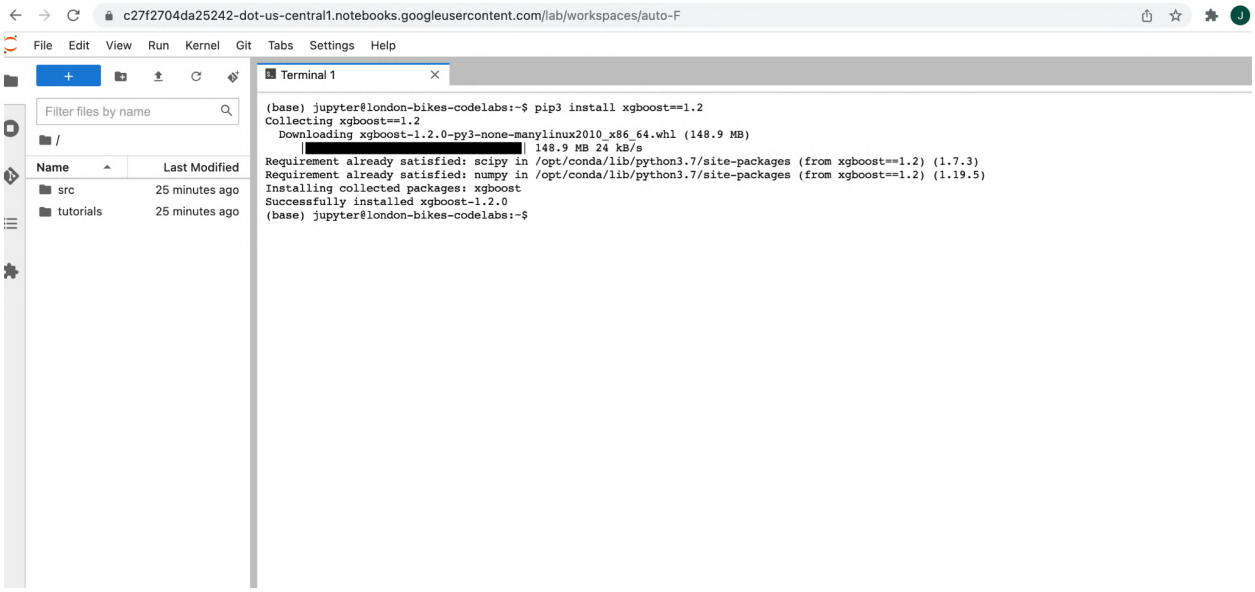
You've learned how to use Vertex AI to:

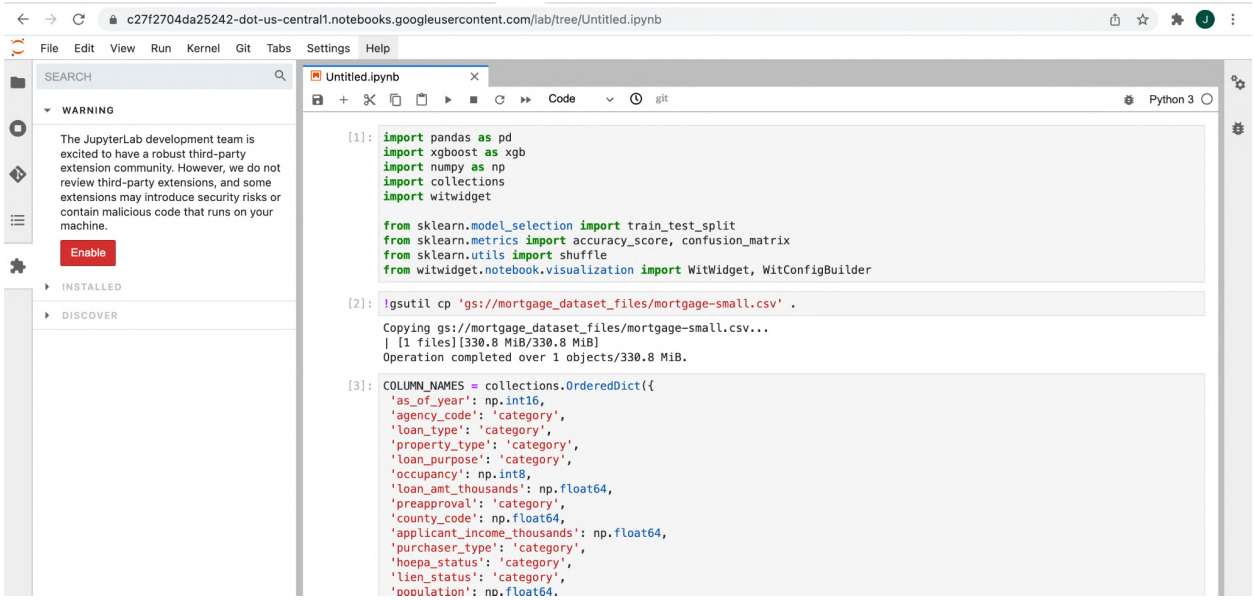
- Launch a multi-worker training job for training code provided in a custom container. You used a TensorFlow model in this example, but you can train a model built with any framework using custom or built-in containers.

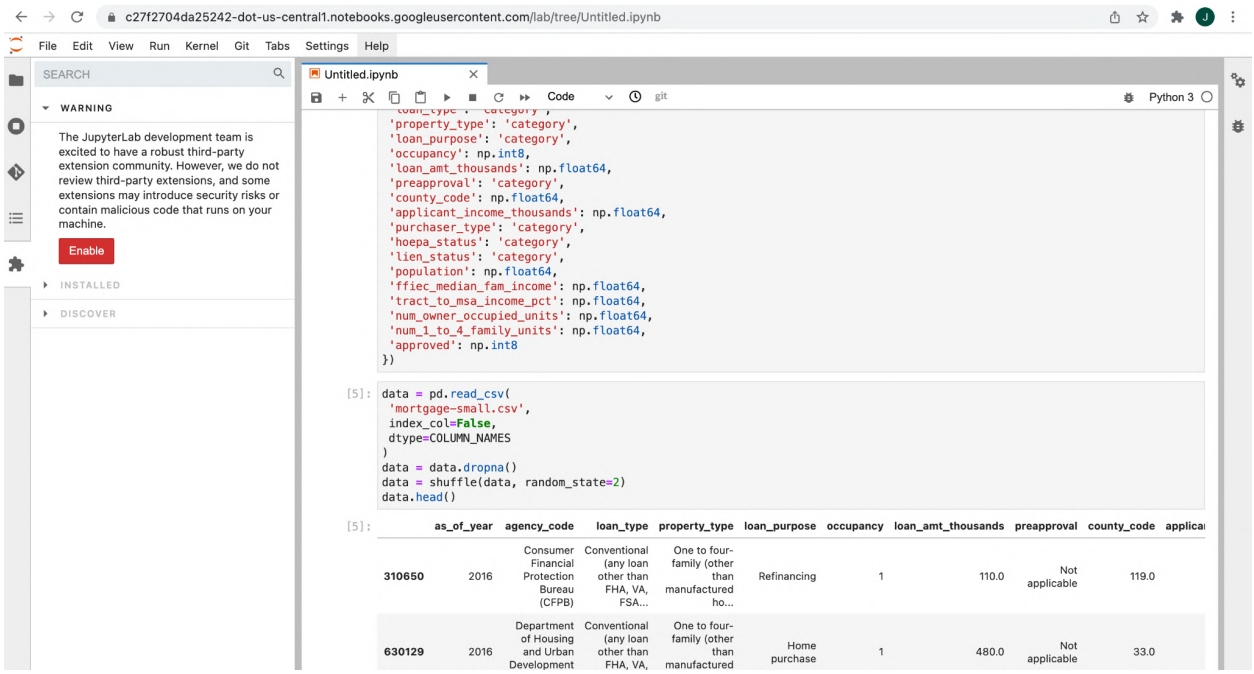


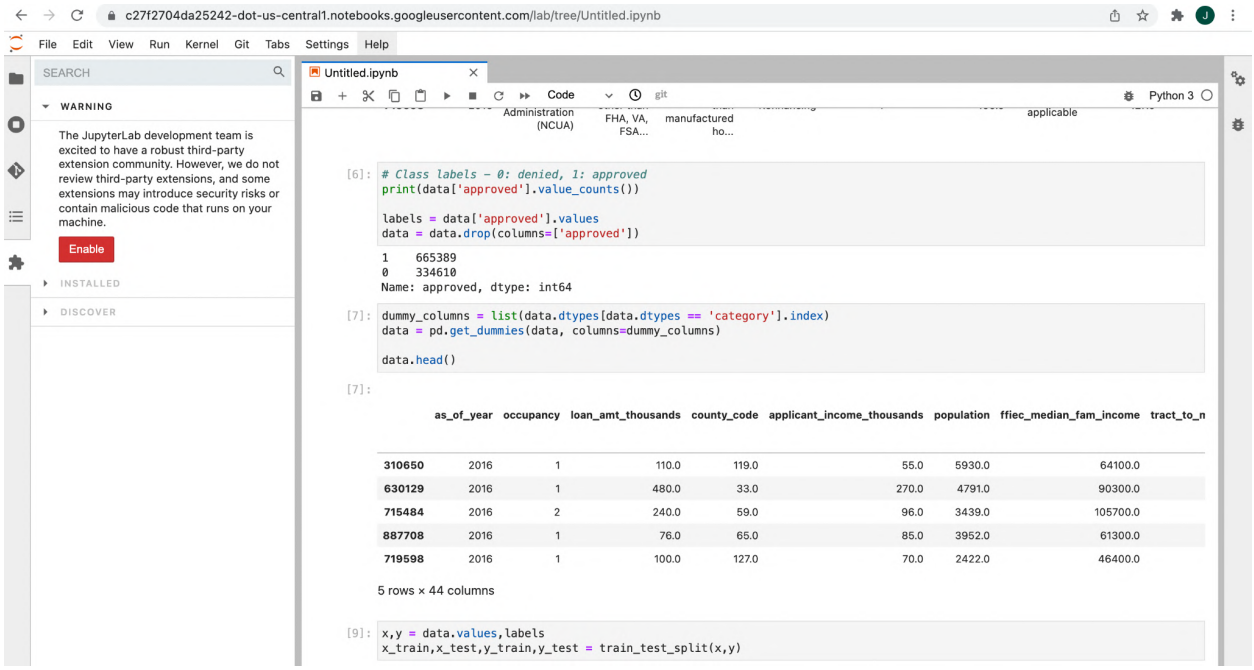


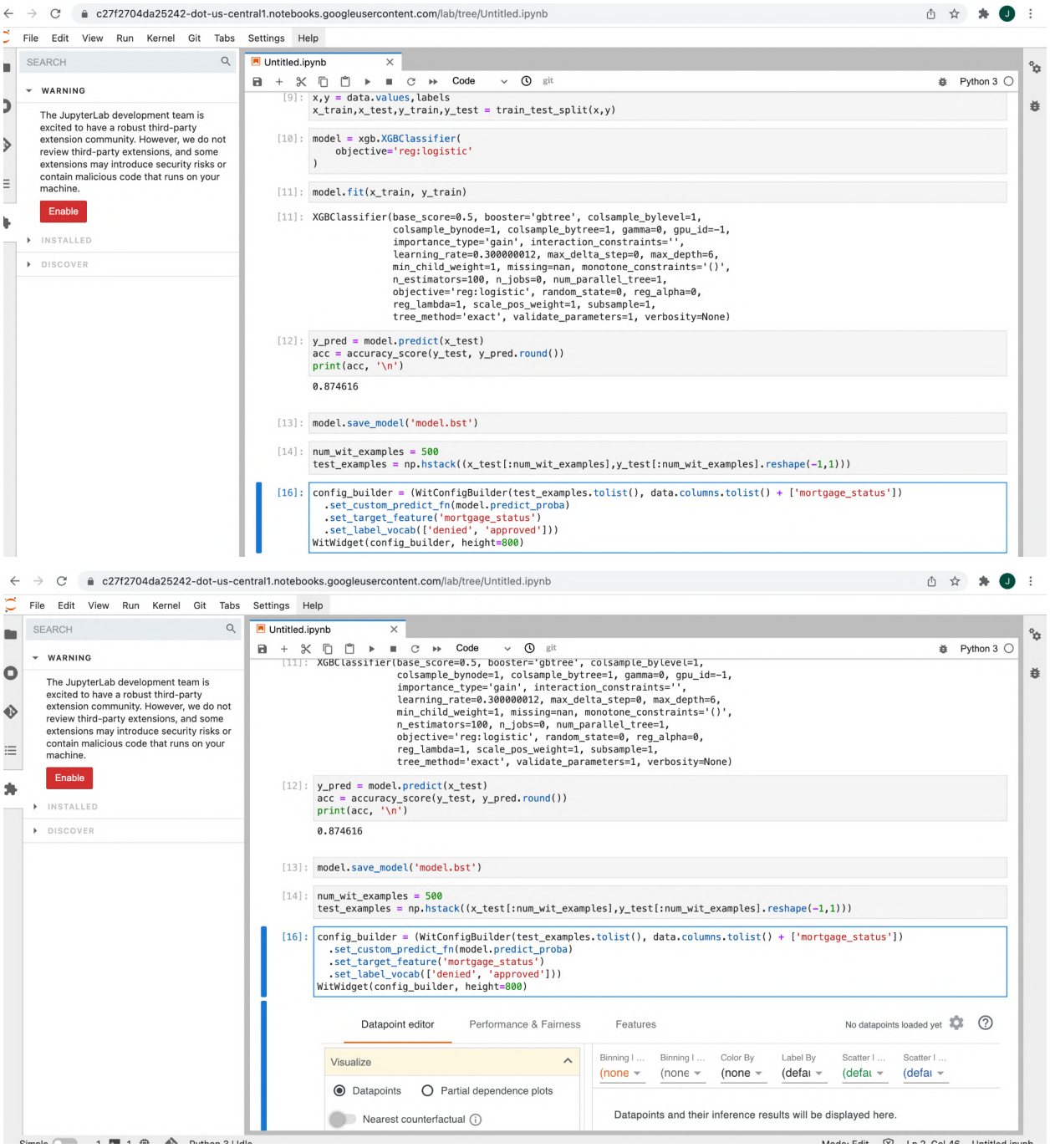
k) https://codelabs.developers.google.com/vertexx-xgb-wit#0



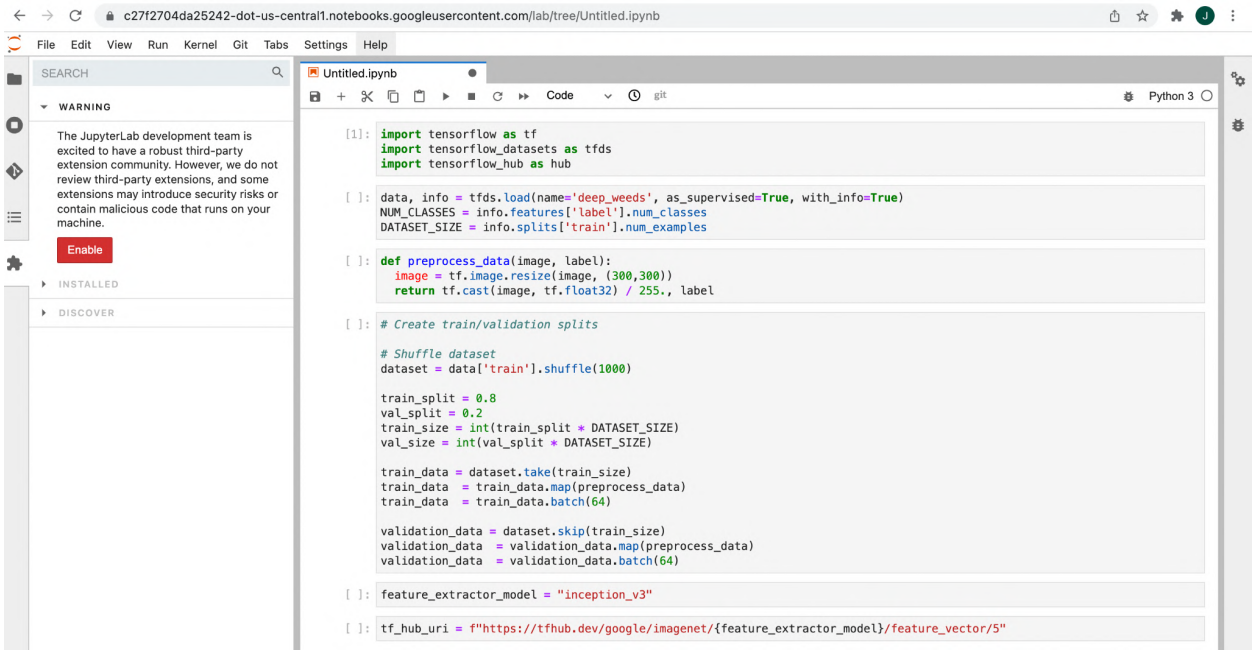

c27f2704da25242-dot-us-central1.notebooks.googleusercontent.com/lab/tree/Untitled.ipynb
File Edit View Run Kernel Git Tabs Settings Help
SEARCH
WARNING
The JupyterLab development team is excited to have a robust third-party extension community. However, we do not review third-party extensions, and some extensions may introduce security risks or contain malicious code that runs on your machine.
Enable
INSTALLED
DISCOVER
Untitled.ipynb
Code
git
Python 3
[1]:
import pandas as pd
import xgboost as xgb
import numpy as np
import collections
import witwidget
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.utils import shuffle
from witwidget.notebook.visualization import WitWidget, WitConfigBuilder
[2]:
!gsutil cp 'gs://mortgage_dataset_files/mortgage-small.csv' .
Copying gs://mortgage_dataset_files/mortgage-small.csv...
| [1 files][330.8 MiB/330.8 MiB]
Operation completed over 1 objects/330.8 MiB.
[3]:
COLUMN_NAMES = collections.OrderedDict({
'as_of_year': np.int16,
'agency_code': 'category',
'loan_type': 'category',
'property_type': 'category',
'loan_purpose': 'category',
'occupancy': np.int8,
'loan_amt_thousands': np.float64,
'preapproval': 'category',
'county_code': np.float64,
'applicant_income_thousands': np.float64,
'purchaser_type': 'category',
'hoepa_status': 'category',
'lien_status': 'category',
'population': np.float64,

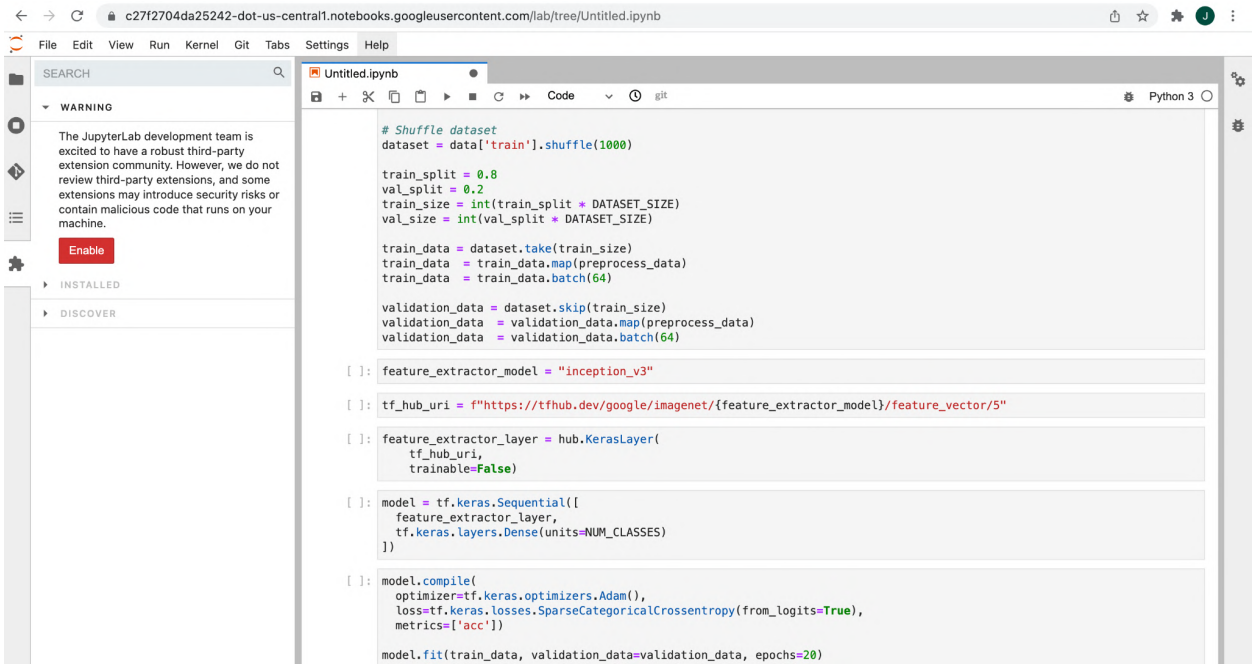

c27f2704da25242-dot-us-central1.notebooks.googleusercontent.com/lab/tree/Untitled.ipynb
File Edit View Run Kernel Git Tabs Settings Help
SEARCH
WARNING
The JupyterLab development team is excited to have a robust third-party extension community. However, we do not review third-party extensions, and some extensions may introduce security risks or contain malicious code that runs on your machine.
Enable
INSTALLED
DISCOVER
Untitled.ipynb
Code
git
Python 3
'property_type': 'category',
'loan_purpose': 'category',
'occupancy': np.int8,
'loan_amt_thousands': np.float64,
'preapproval': 'category',
'county_code': np.float64,
'applicant_income_thousands': np.float64,
'purchaser_type': 'category',
'hoepa_status': 'category',
'lien_status': 'category',
'population': np.float64,
'ffiec_median_fam_income': np.float64,
'tract_to_msa_income_pct': np.float64,
'num_owner_occupied_units': np.float64,
'num_1_to_4_family_units': np.float64,
'approved': np.int8
})
[5]:
data = pd.read_csv(
'mortgage-small.csv',
index_col=False,
dtype=COLUMN_NAMES
)
data = data.dropna()
data = shuffle(data, random_state=2)
data.head()
[5]:
as_of_year agency_code loan_type property_type loan_purpose occupancy loan_amt_thousands preapproval county_code applica
310650 2016 Consumer Financial Protection Bureau (CFPB) Conventional (any loan other than FHA, VA, FSA... One to four-family (other than manufactured ho... Refinancing 1 110.0 Not applicable 119.0
630129 2016 Department of Housing and Urban Development Conventional (any loan other than FHA, VA, One to four-family (other than manufactured Home purchase 1 480.0 Not applicable 33.0


c27f2704da25242-dot-us-central1.notebooks.googleusercontent.com/lab/tree/Untitled.ipynb
File Edit View Run Kernel Git Tabs Settings Help
SEARCH
WARNING
The JupyterLab development team is excited to have a robust third-party extension community. However, we do not review third-party extensions, and some extensions may introduce security risks or contain malicious code that runs on your machine.
Enable
INSTALLED
DISCOVER
Untitled.ipynb
Code
git
Python 3
Administration (NCUA)
FHA, VA, FSA...
manufactured ho...
applicable
[6]:
# Class labels – 0: denied, 1: approved
print(data['approved'].value_counts())
labels = data['approved'].values
data = data.drop(columns=['approved'])
1 665389
0 334610
Name: approved, dtype: int64
[7]:
dummy_columns = list(data.dtypes[data.dtypes == 'category'].index)
data = pd.get_dummies(data, columns=dummy_columns)
data.head()
[7]:
as_of_year occupancy loan_amt_thousands county_code applicant_income_thousands population ffiec_median_fam_income tract_to_n
310650 2016 1 110.0 119.0 55.0 5930.0 64100.0
630129 2016 1 480.0 33.0 270.0 4791.0 90300.0
715484 2016 2 240.0 59.0 96.0 3439.0 105700.0
887708 2016 1 76.0 65.0 85.0 3952.0 61300.0
719598 2016 1 100.0 127.0 70.0 2422.0 46400.0
5 rows × 44 columns
[9]:
x,y = data.values,labels
x_train,x_test,y_train,y_test = train_test_split(x,y)

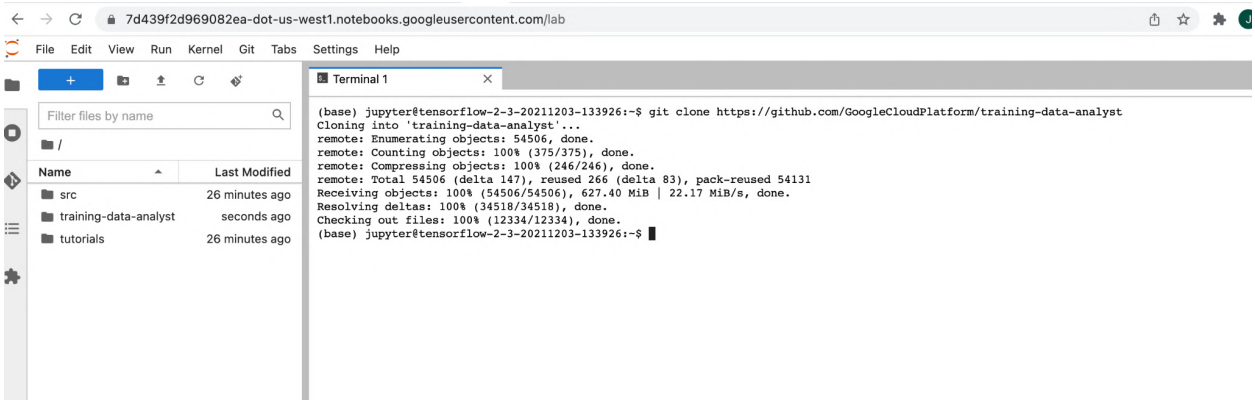


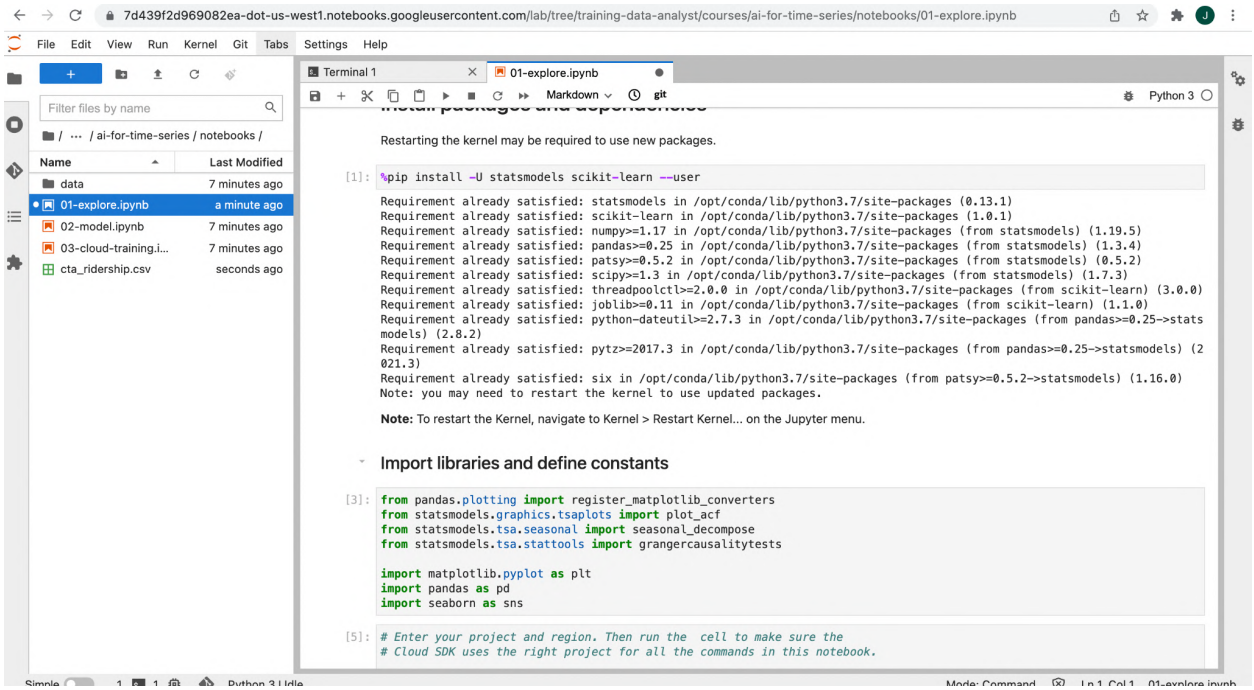
l) https://codelabs.developers.google.com/vertex_notebook_executor#0

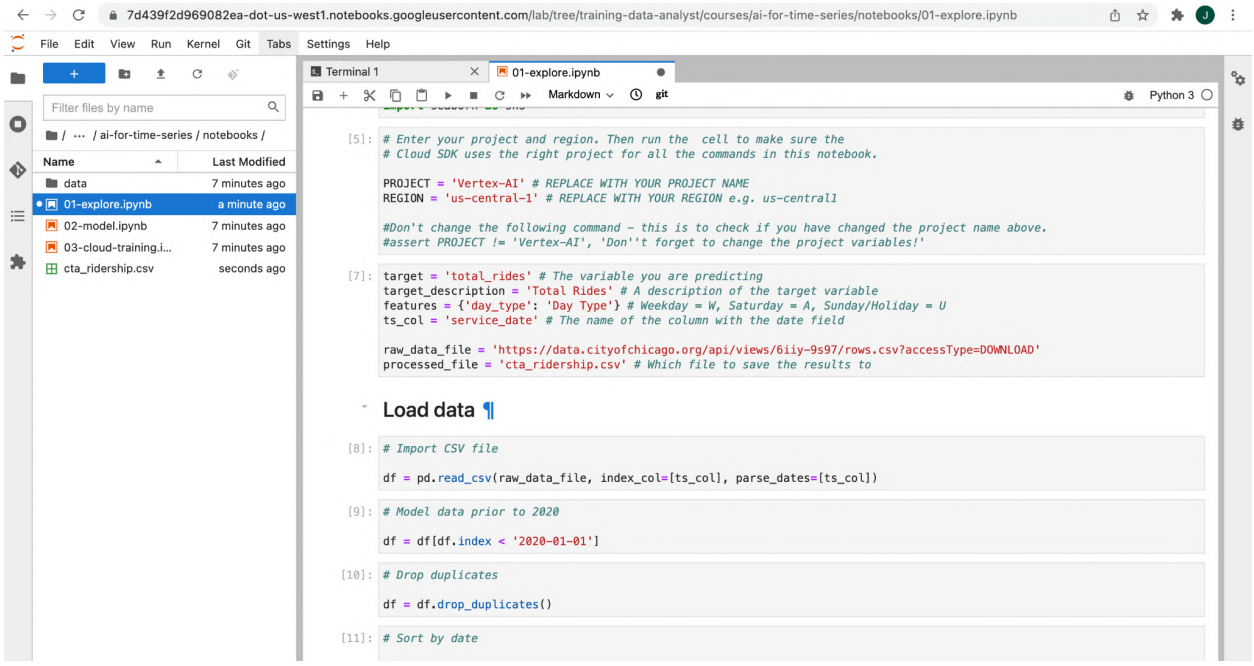


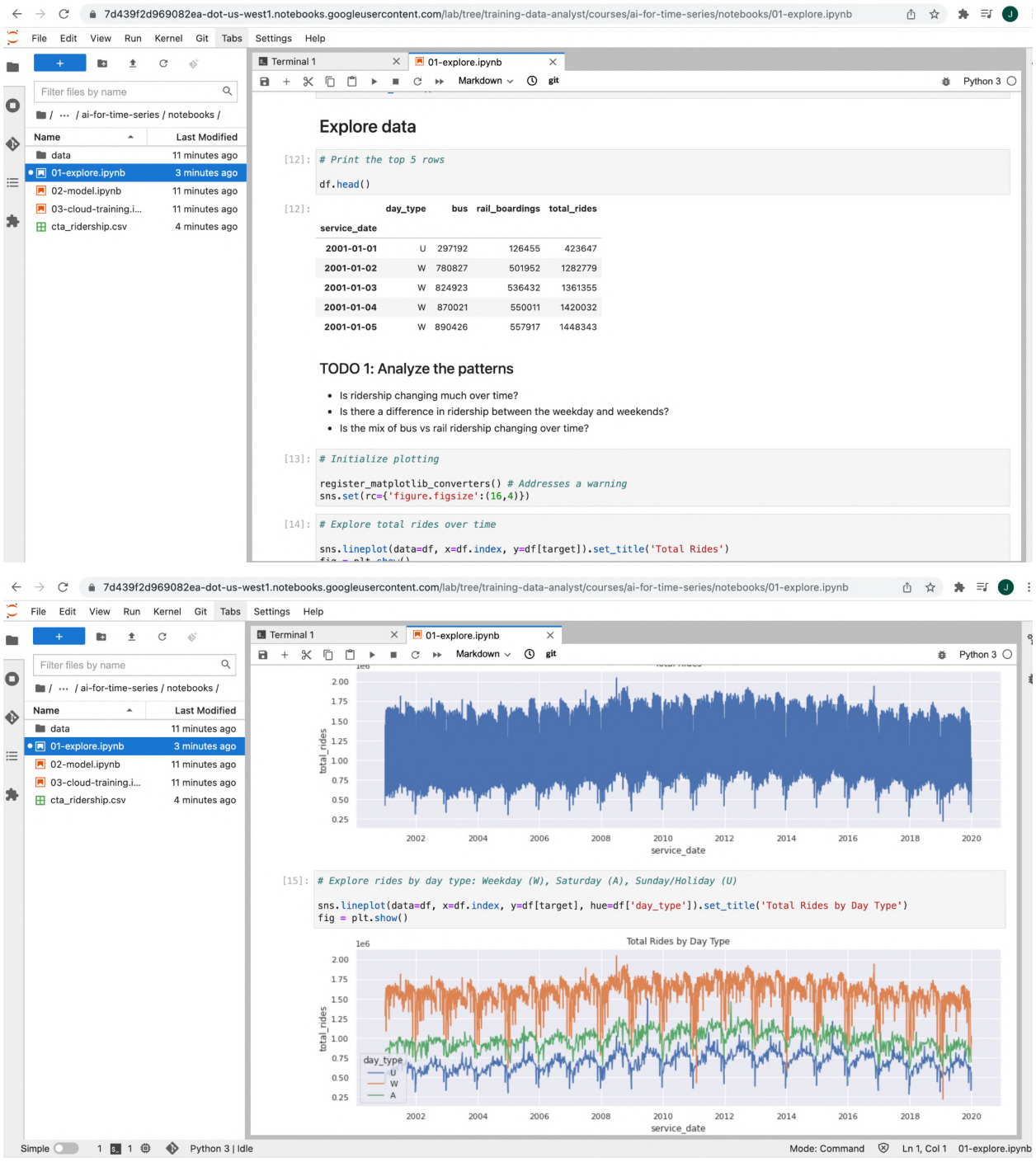


m)https://codelabs.developers.google.com/codelabs/time-series-forecasting-with-cloud-ai-platform#0


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/
Name
Last Modified
src
26 minutes ago
training-data-analyst
seconds ago
tutorials
26 minutes ago
Terminal 1
(base) jupyter@tensorflow-2-3-20211203-133926:~$ git clone https://github.com/GoogleCloudPlatform/training-data-analyst
Cloning into 'training-data-analyst'...
remote: Enumerating objects: 54506, done.
remote: Counting objects: 100% (375/375), done.
remote: Compressing objects: 100% (246/246), done.
remote: Total 54506 (delta 147), reused 266 (delta 83), pack-reused 54131
Receiving objects: 100% (54506/54506), 627.40 MiB | 22.17 MiB/s, done.
Resolving deltas: 100% (34518/34518), done.
Checking out files: 100% (12334/12334), done.
(base) jupyter@tensorflow-2-3-20211203-133926:~$


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name Last Modified
data 7 minutes ago
01-explore.ipynb a minute ago
02-model.ipynb 7 minutes ago
03-cloud-training.i... 7 minutes ago
cta_ridership.csv seconds ago
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
Restarting the kernel may be required to use new packages.
[1]: %pip install -U statsmodels scikit-learn --user
Requirement already satisfied: statsmodels in /opt/conda/lib/python3.7/site-packages (0.13.1)
Requirement already satisfied: scikit-learn in /opt/conda/lib/python3.7/site-packages (1.0.1)
Requirement already satisfied: numpy>=1.17 in /opt/conda/lib/python3.7/site-packages (from statsmodels) (1.19.5)
Requirement already satisfied: pandas>=0.25 in /opt/conda/lib/python3.7/site-packages (from statsmodels) (1.3.4)
Requirement already satisfied: patsy>=0.5.2 in /opt/conda/lib/python3.7/site-packages (from statsmodels) (0.5.2)
Requirement already satisfied: scipy>=1.3 in /opt/conda/lib/python3.7/site-packages (from statsmodels) (1.7.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/conda/lib/python3.7/site-packages (from scikit-learn) (3.0.0)
Requirement already satisfied: joblib>=0.11 in /opt/conda/lib/python3.7/site-packages (from scikit-learn) (1.1.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /opt/conda/lib/python3.7/site-packages (from pandas>=0.25->statsmodels) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-packages (from pandas>=0.25->statsmodels) (2021.3)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from patsy>=0.5.2->statsmodels) (1.16.0)
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
[3]: 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
[5]: # 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.


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name Last Modified
data 7 minutes ago
01-explore.ipynb a minute ago
02-model.ipynb 7 minutes ago
03-cloud-training.i... 7 minutes ago
cta_ridership.csv seconds ago
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
[5]: # 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 = 'Vertex-AI' # 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 != 'Vertex-AI', 'Don''t forget to change the project variables!'
[7]: 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 ¶
[8]: # Import CSV file
df = pd.read_csv(raw_data_file, index_col=[ts_col], parse_dates=[ts_col])
[9]: # Model data prior to 2020
df = df[df.index < '2020-01-01']
[10]: # Drop duplicates
df = df.drop_duplicates()
[11]: # Sort by date


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 11 minutes ago
01-explore.ipynb 3 minutes ago
02-model.ipynb 11 minutes ago
03-cloud-training.i... 11 minutes ago
cta_ridership.csv 4 minutes ago
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
Explore data
[12]:
# Print the top 5 rows
df.head()
[12]:
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
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?
[13]:
Initialize plotting
register_matplotlib_converters() # Addresses a warning
sns.set(rc={'figure.figsize':(16,4)})
[14]:
# Explore total rides over time
sns.lineplot(data=df, x=df.index, y=df[target]).set_title('Total Rides')

7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 11 minutes ago
01-explore.ipynb 3 minutes ago
02-model.ipynb 11 minutes ago
03-cloud-training.i... 11 minutes ago
cta_ridership.csv 4 minutes ago
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
total_rides
service_date
[15]:
# 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()
Total Rides by Day Type
1e6
day_type
U
W
A
Simple
Python 3 | Idle
Mode: Command
Ln 1, Col 1
01-explore.ipynb

7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb

File Edit View Run Kernel Git Tabs Settings Help

Filter files by name

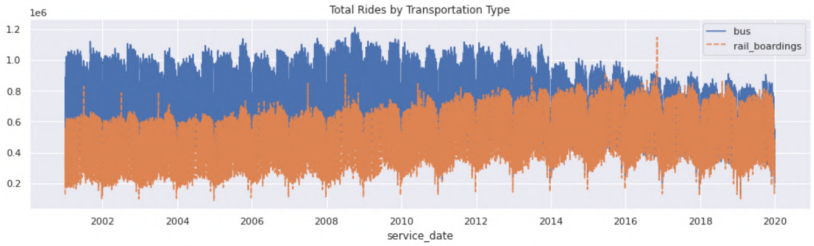
/ ... / ai-for-time-series / notebooks /

Name	Last Modified
data	11 minutes ago
01-explore.ipynb	3 minutes ago
02-model.ipynb	11 minutes ago
03-cloud-training.i...	11 minutes ago
cta_ridership.csv	4 minutes ago

Terminal 1 | 01-explore.ipynb

Markdown | git | Python 3

[16]:
```
# 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?

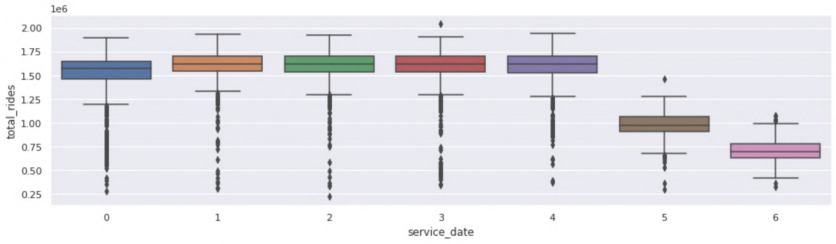
[17]:
```
df[target].describe().apply(lambda x: round(x))
```

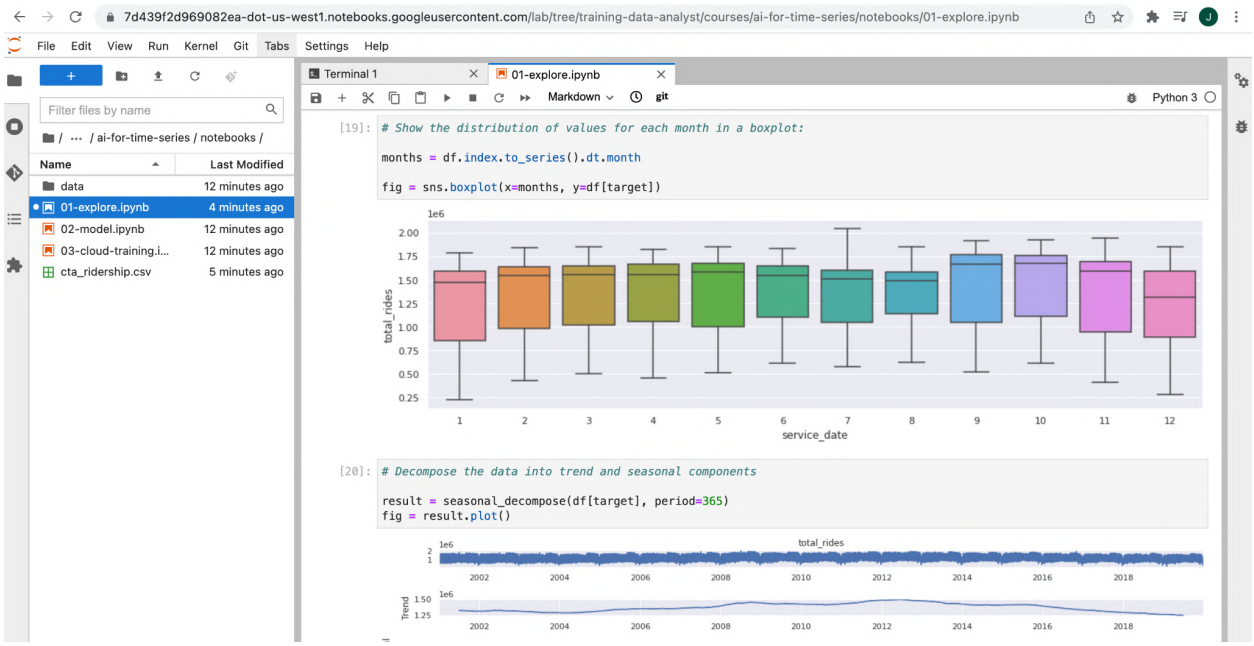
[17]:
```
count       6939
mean     1368761
std       391443
min       222071
25%      1005394
50%      1548343
75%      1660947
max      2049519
Name: total_rides, dtype: int64
```

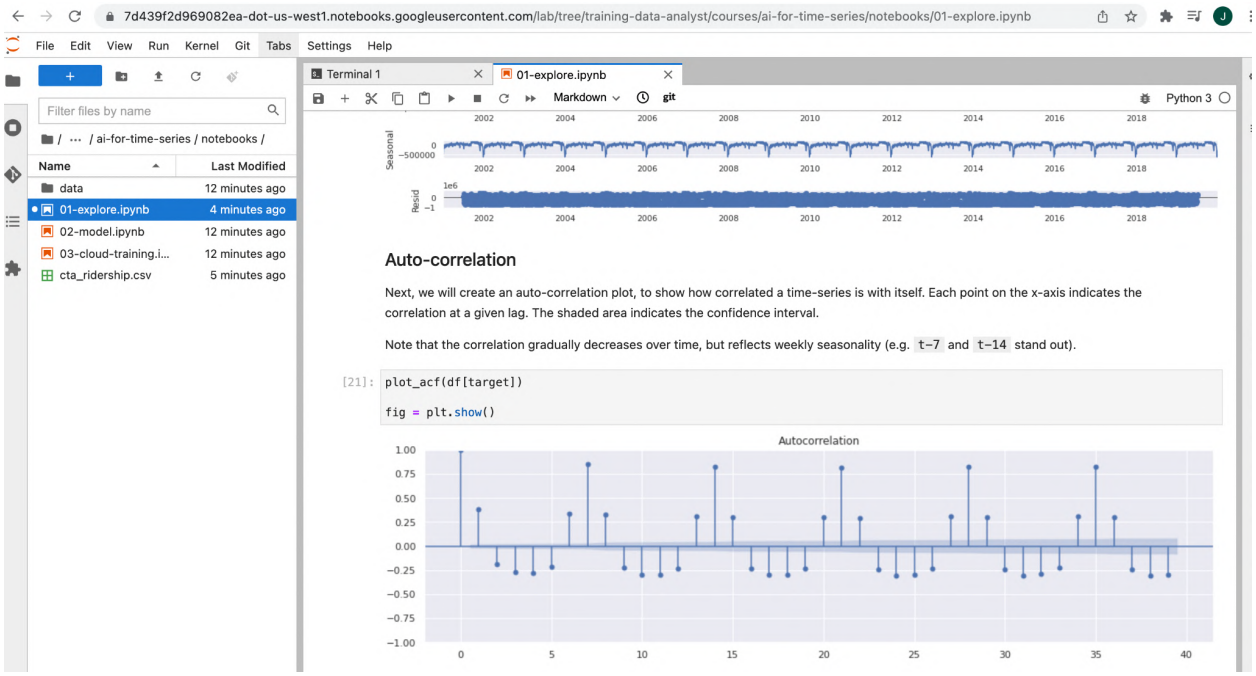
TODO 3: Explore seasonality

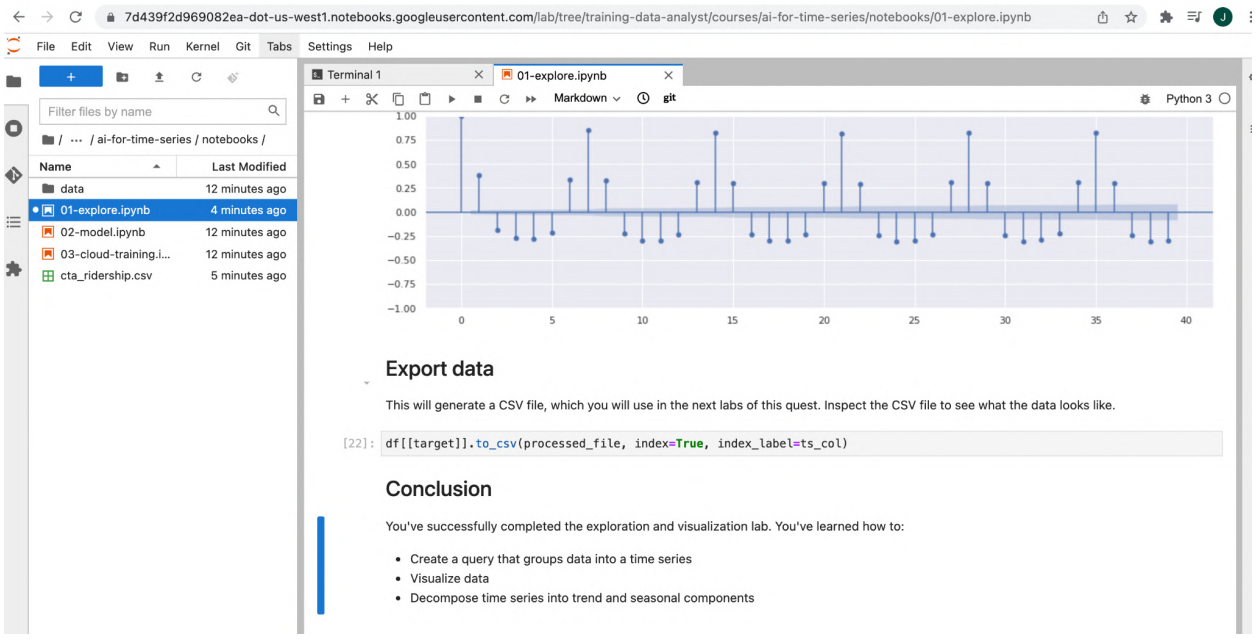
- Is there much difference between months?
- Can you extract the trend and seasonal pattern from the data?

[18]:
```
# 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])
```




7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 12 minutes ago
01-explore.ipynb 4 minutes ago
02-model.ipynb 12 minutes ago
03-cloud-training.i... 12 minutes ago
cta_ridership.csv 5 minutes ago
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
[19]:
# 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])
1e6
total_rides
service_date
[20]:
# Decompose the data into trend and seasonal components
result = seasonal_decompose(df[target], period=365)
fig = result.plot()
total_rides
Trend


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 12 minutes ago
01-explore.ipynb 4 minutes ago
02-model.ipynb 12 minutes ago
03-cloud-training.i... 12 minutes ago
cta_ridership.csv 5 minutes ago
Seasonal
Resid
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).
[21]:
plot_acf(df[target])
fig = plt.show()
Autocorrelation


7d439f2d969082ea-dot-us-west1.notebooks.googleusercontent.com/lab/tree/training-data-analyst/courses/ai-for-time-series/notebooks/01-explore.ipynb
File Edit View Run Kernel Git Tabs Settings Help
Terminal 1
01-explore.ipynb
Markdown
git
Python 3
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 12 minutes ago
01-explore.ipynb 4 minutes ago
02-model.ipynb 12 minutes ago
03-cloud-training.i... 12 minutes ago
cta_ridership.csv 5 minutes ago
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.
[22]:
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