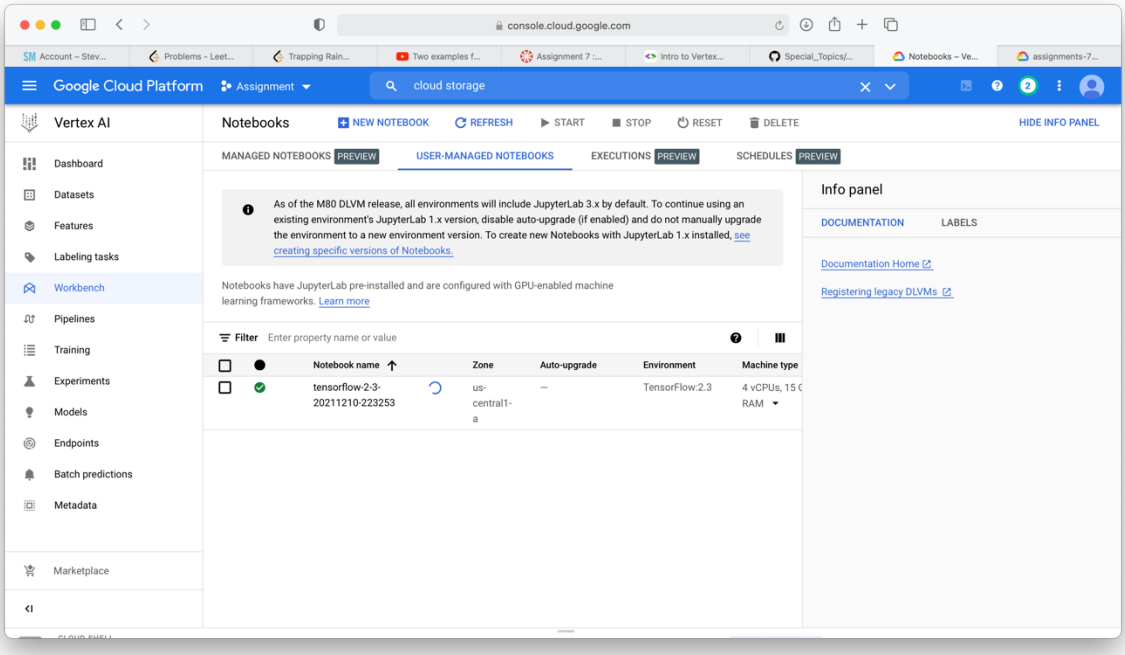
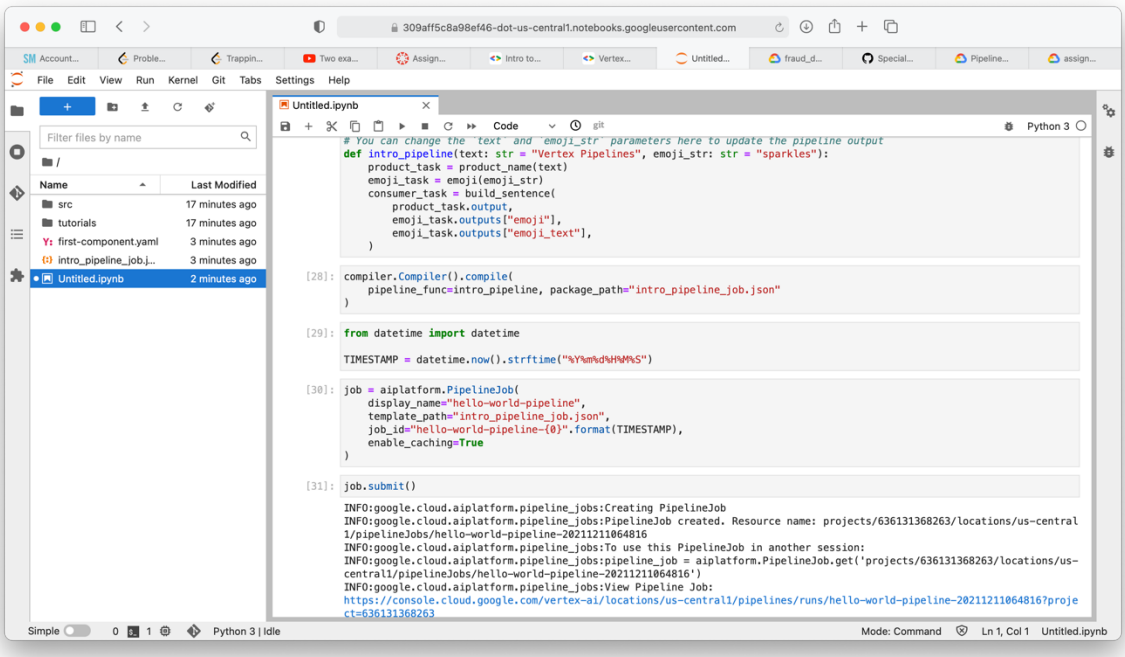
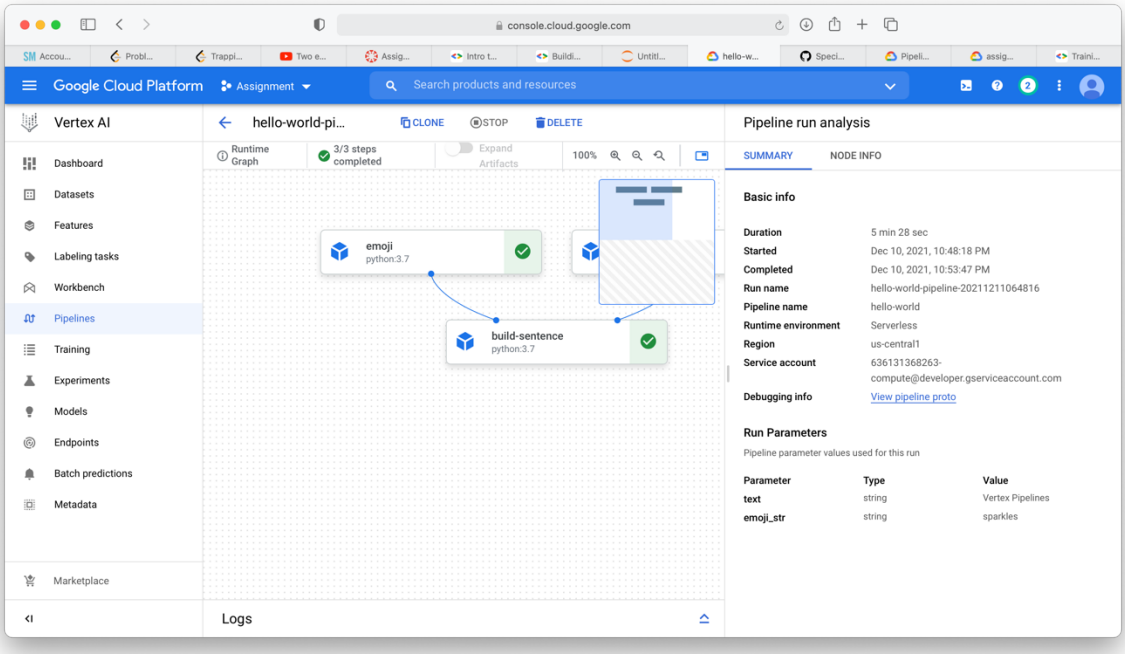
Assignment-7

VERTEX AI

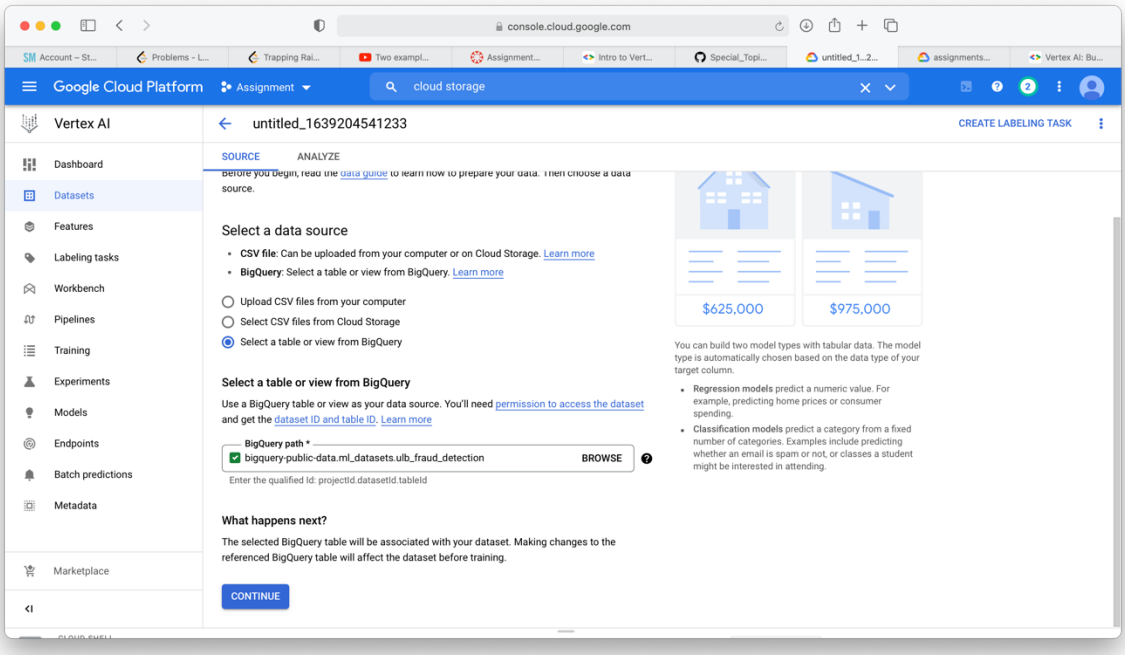
Part a) [https://codelabs.developers.google.com/vertex-pipelines-intro#0 (Links to an external site.)](https://codelabs.developers.google.com/vertex-pipelines-intro#0)

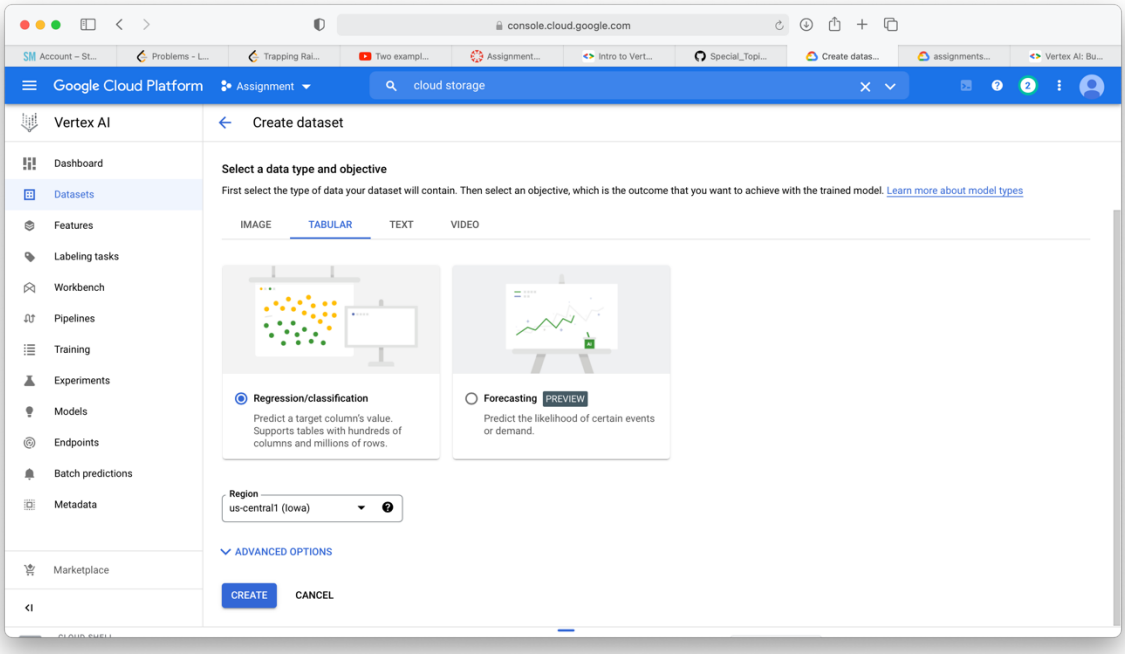


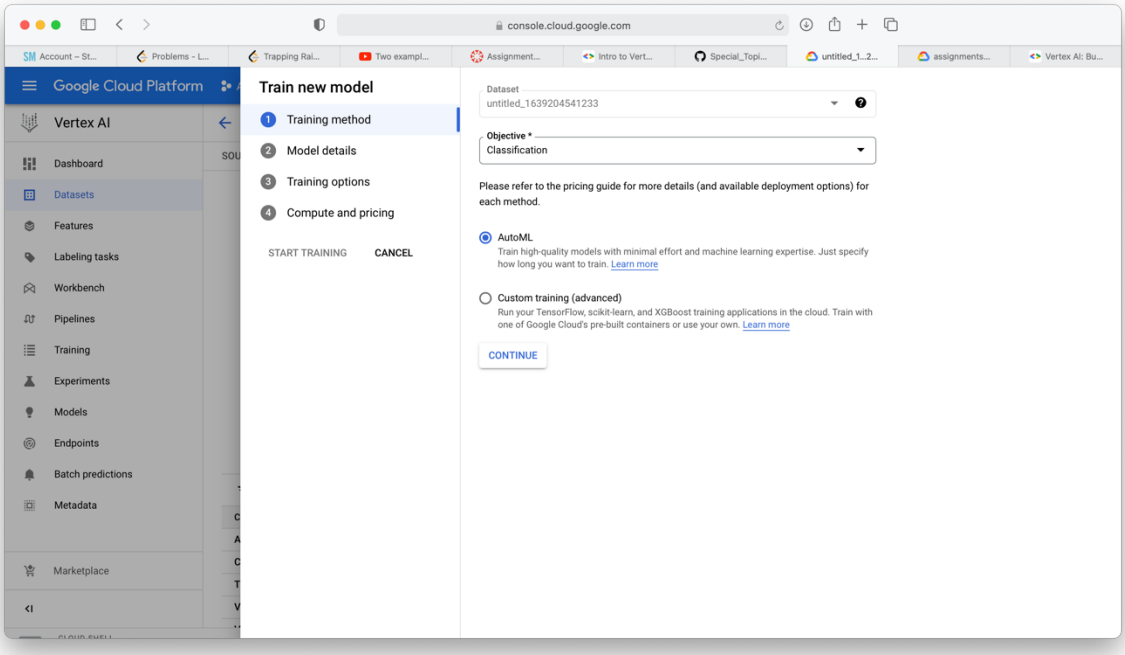


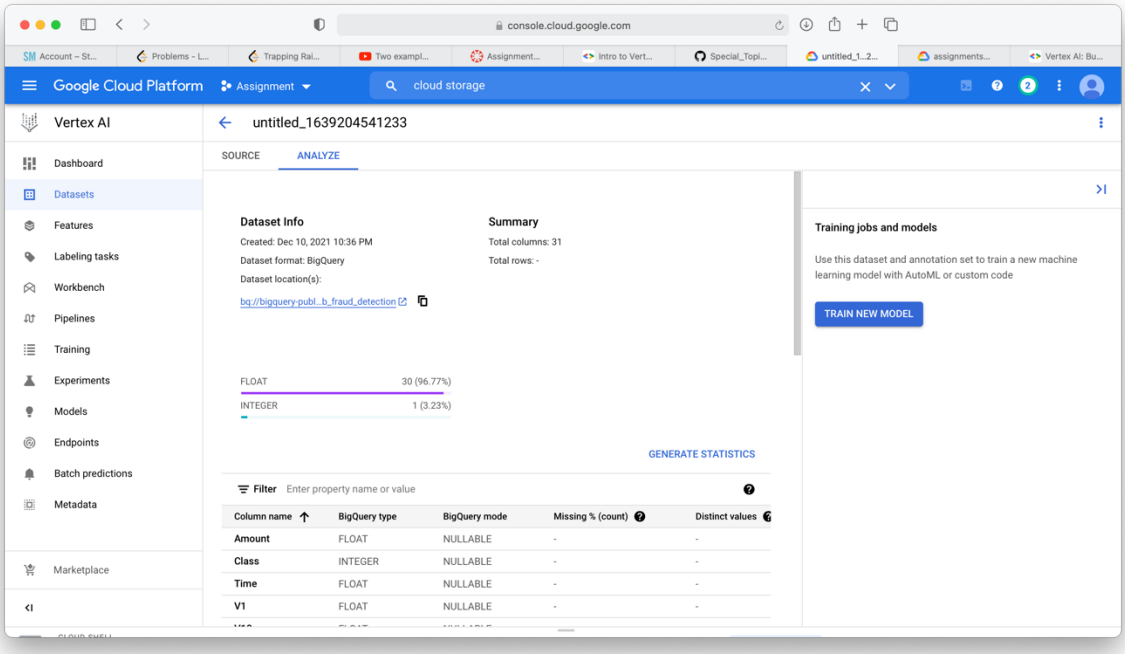


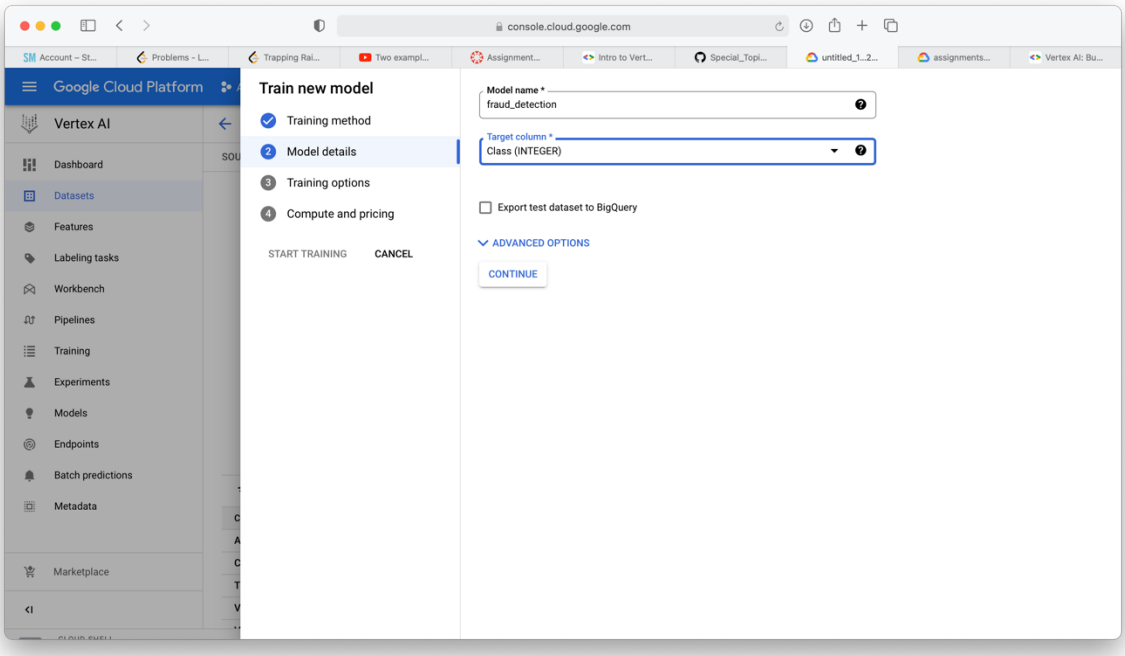
Part b) https://codelabs.developers.google.com/vertex-automl-tabular#0

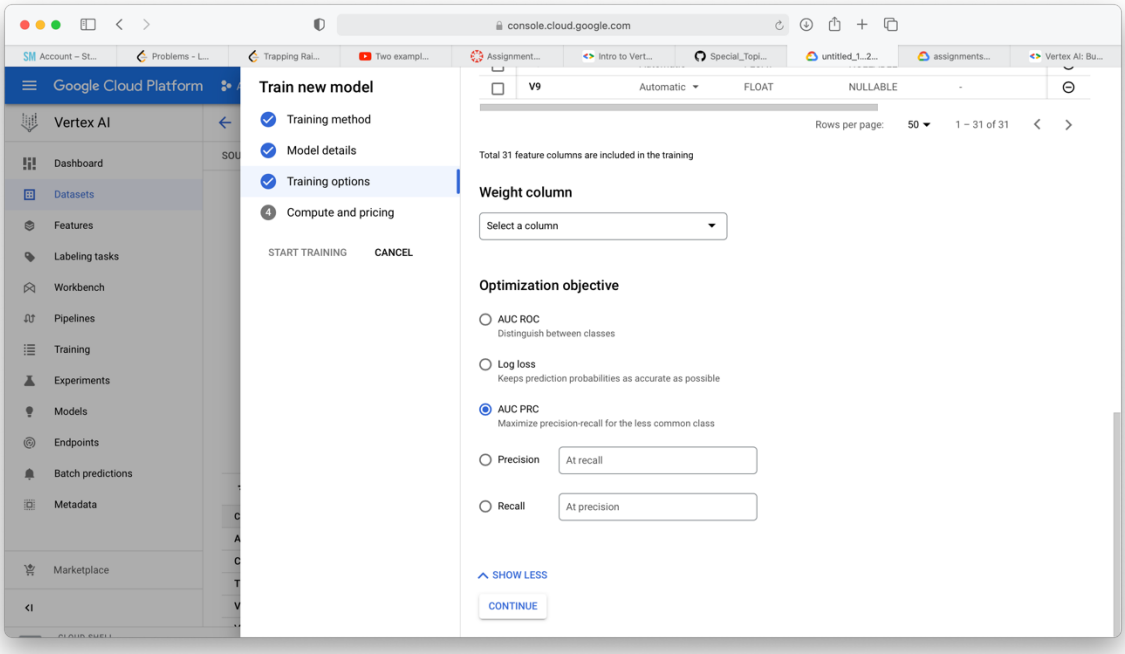



console.cloud.google.com
Google Cloud Platform
Assignment
cloud storage
Vertex AI
Create dataset
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
Select a data type and objective
First select the type of data your dataset will contain. Then select an objective, which is the outcome that you want to achieve with the trained model. Learn more about model types
IMAGE
TABULAR
TEXT
VIDEO
Regression/classification
Predict a target column's value. Supports tables with hundreds of columns and millions of rows.
Forecasting PREVIEW
Predict the likelihood of certain events or demand.
Region
us-central1 (Iowa)
ADVANCED OPTIONS
CREATE
CANCEL

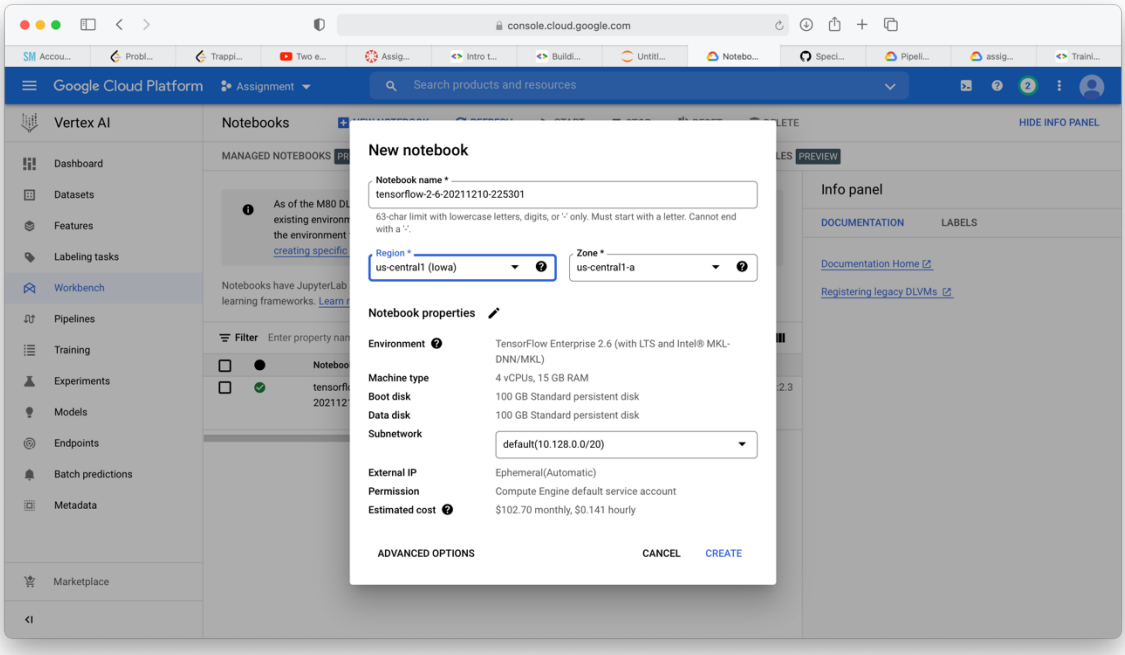

console.cloud.google.com
Google Cloud Platform
Vertex AI
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
Train new model
1 Training method
2 Model details
3 Training options
4 Compute and pricing
START TRAINING
CANCEL
Dataset
untitled_1639204541233
Objective *
Classification
Please refer to the pricing guide for more details (and available deployment options) for each method.
AutoML
Train high-quality models with minimal effort and machine learning expertise. Just specify how long you want to train. Learn more
Custom training (advanced)
Run your TensorFlow, scikit-learn, and XGBoost training applications in the cloud. Train with one of Google Cloud's pre-built containers or use your own. Learn more
CONTINUE

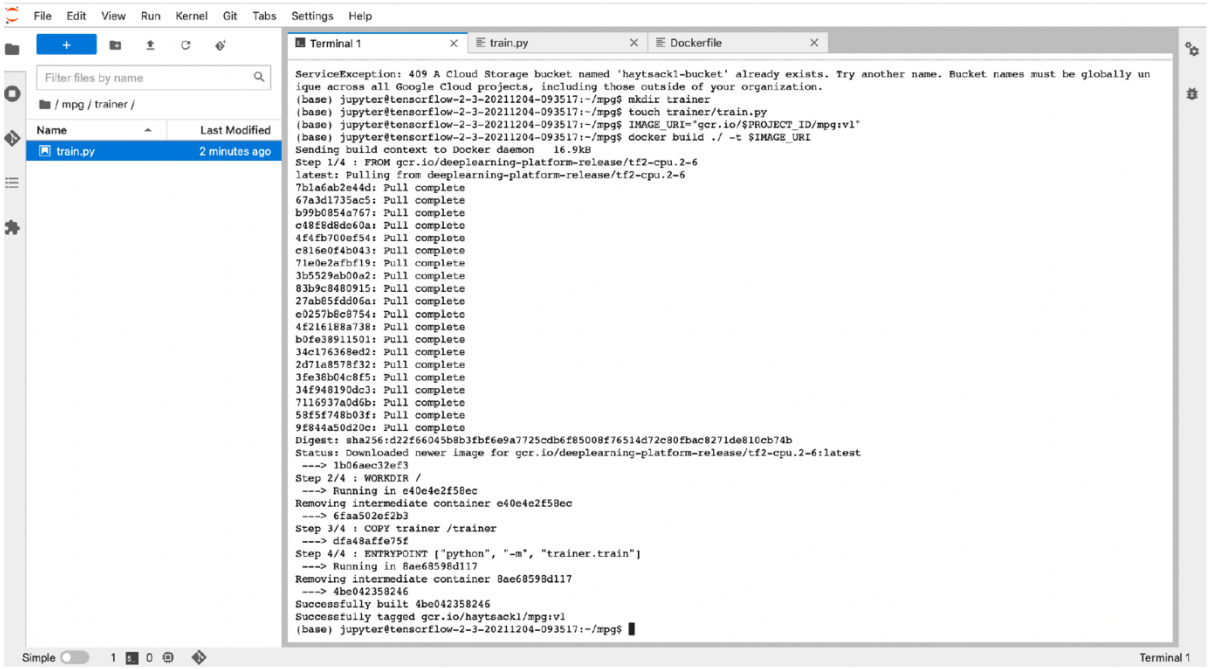

console.cloud.google.com
Google Cloud Platform
Assignment
cloud storage
Vertex AI
untitled_1639204541233
SOURCE
ANALYZE
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
Dataset Info
Created: Dec 10, 2021 10:36 PM
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
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

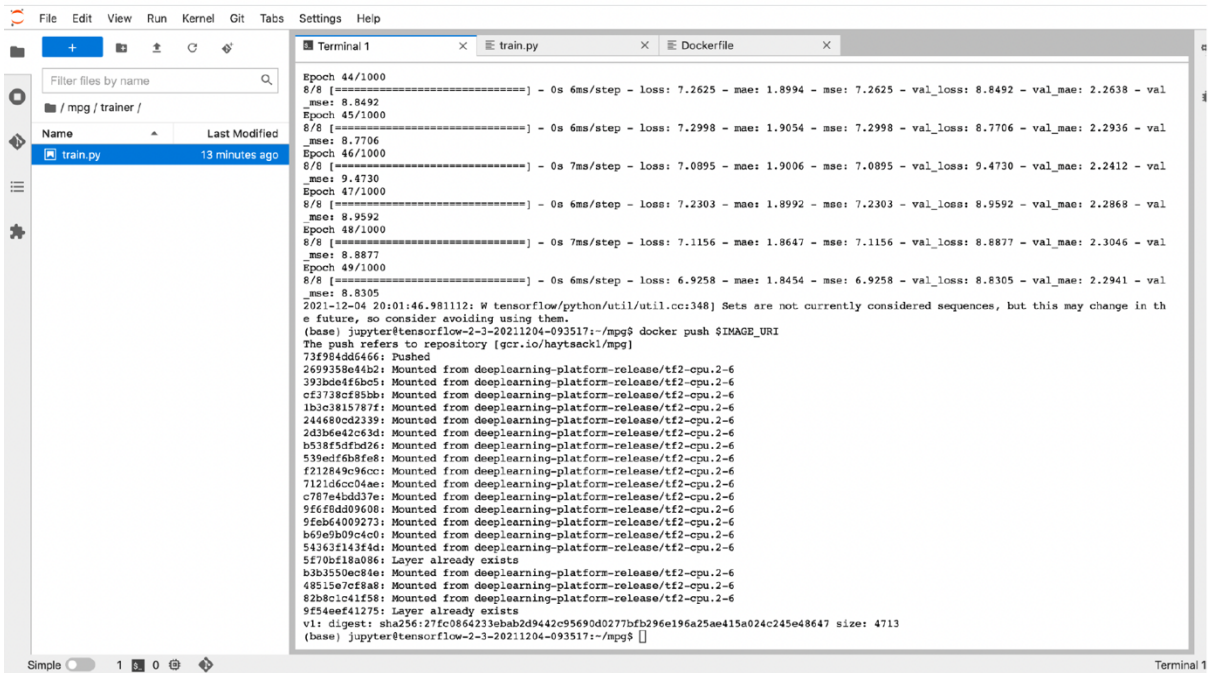

console.cloud.google.com
Google Cloud Platform
Vertex AI
Train new model
Training method
Model details
Training options
Compute and pricing
START TRAINING
CANCEL
Model name *
fraud_detection
Target column *
Class (INTEGER)
Export test dataset to BigQuery
ADVANCED OPTIONS
CONTINUE



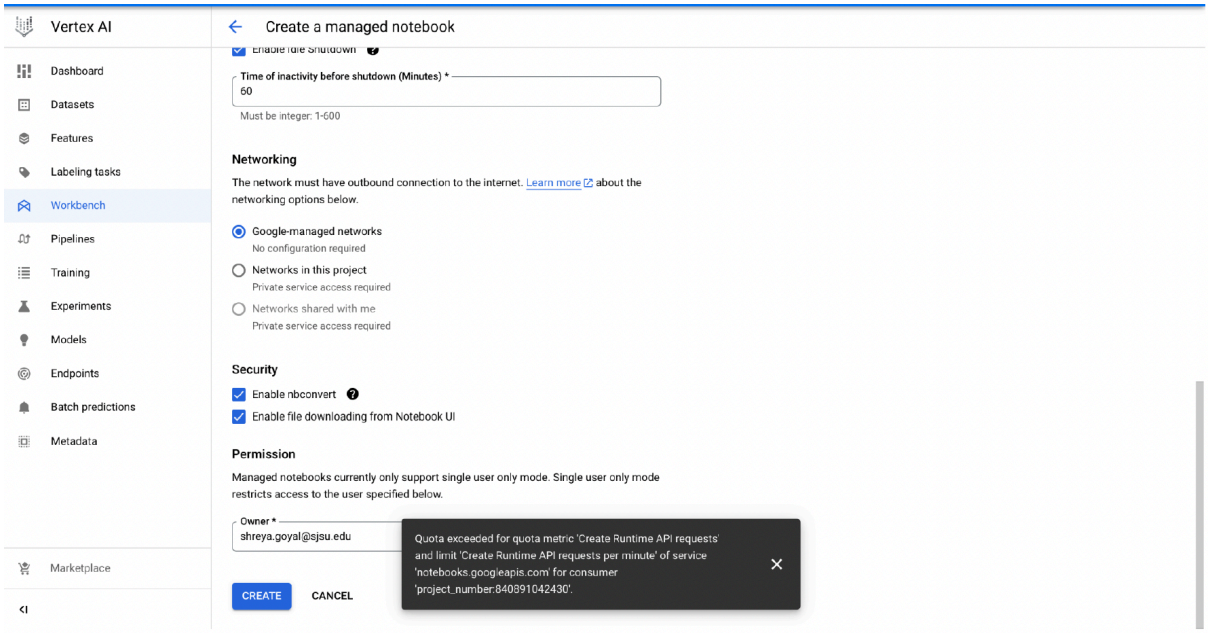
Part c) https://codelabs.developers.google.com/vertex_custom_training_prediction#0

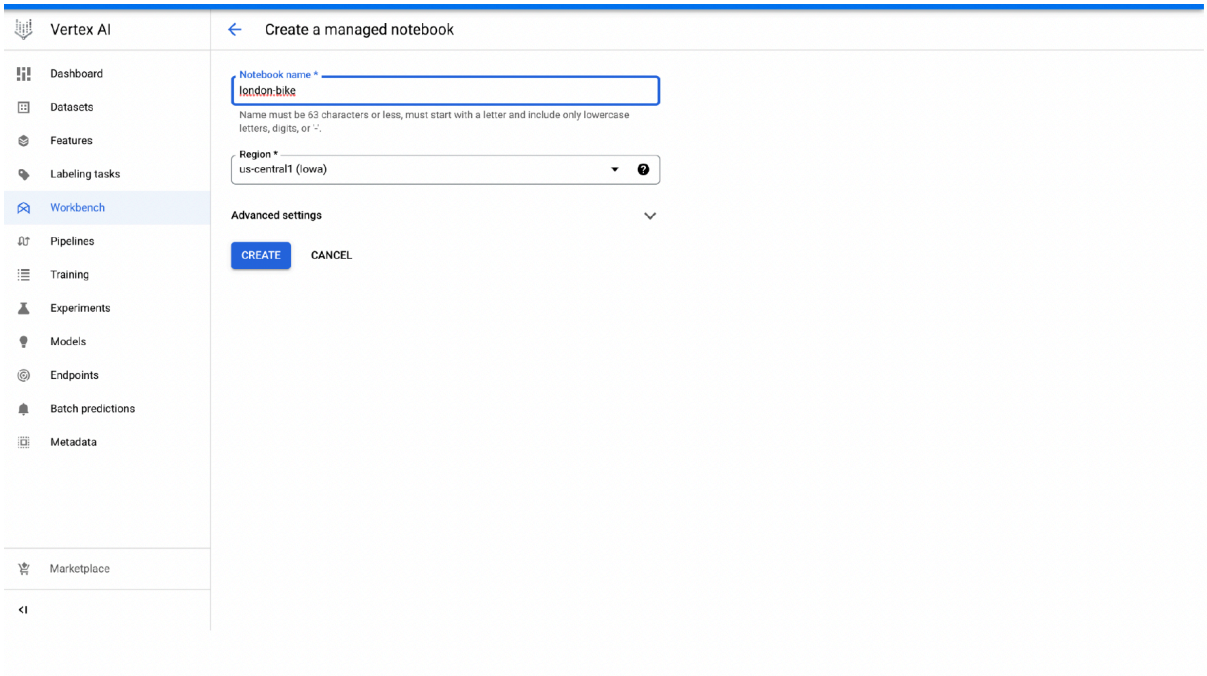




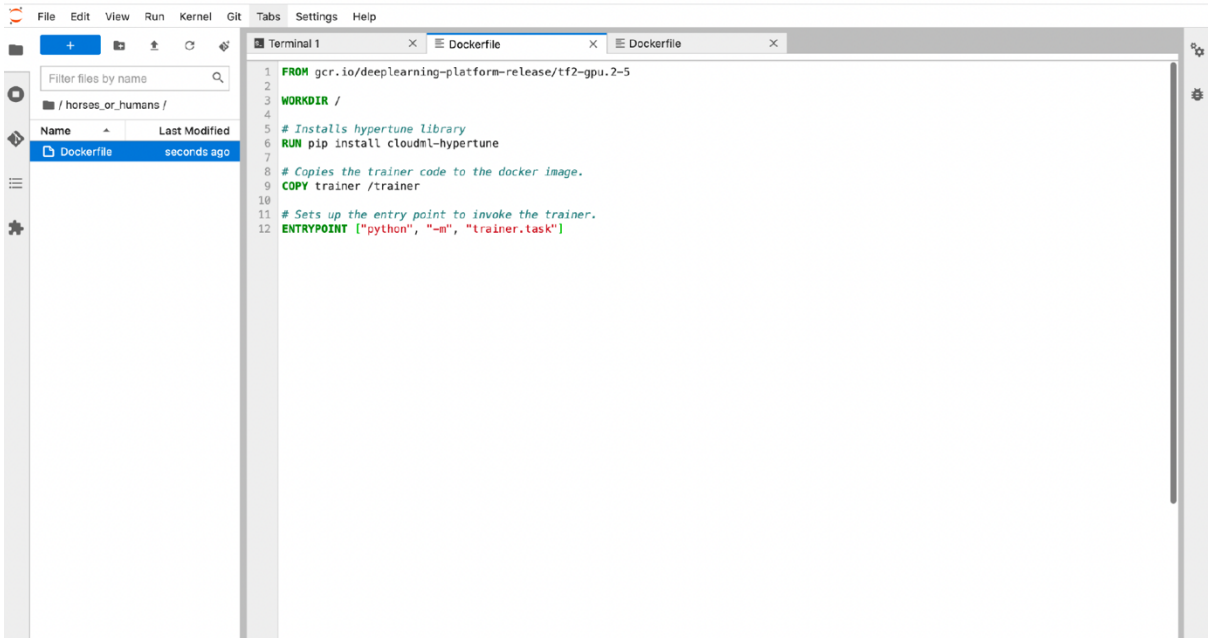


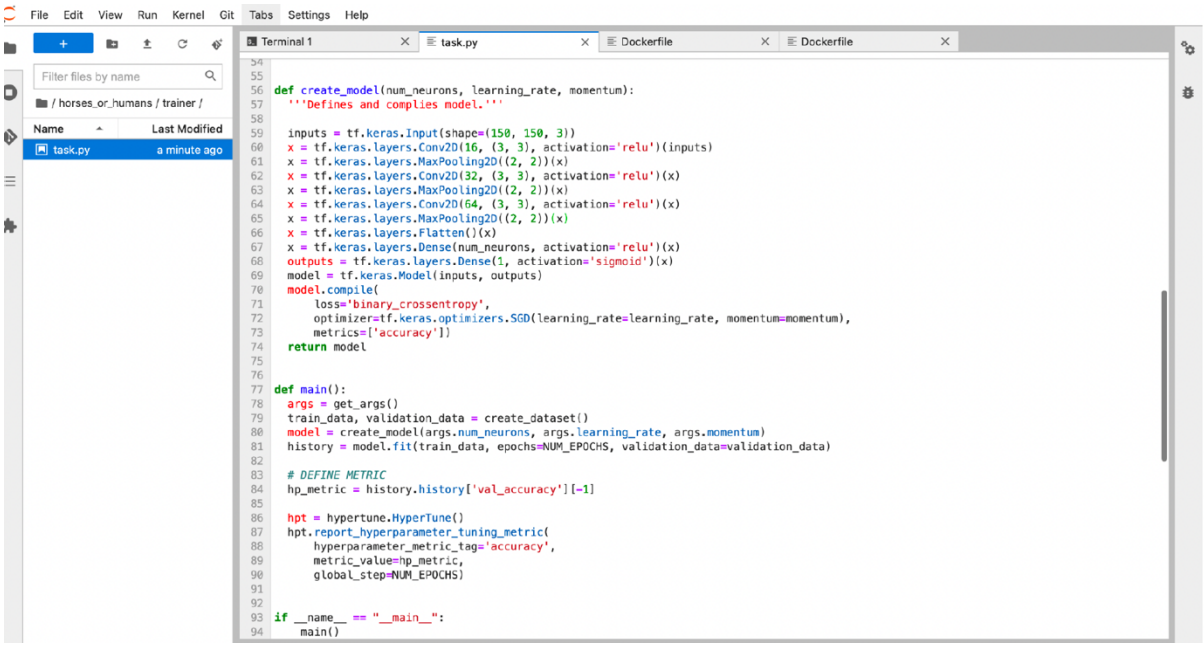
Part d) https://codelabs.developers.google.com/vertex-workbench-intro#0

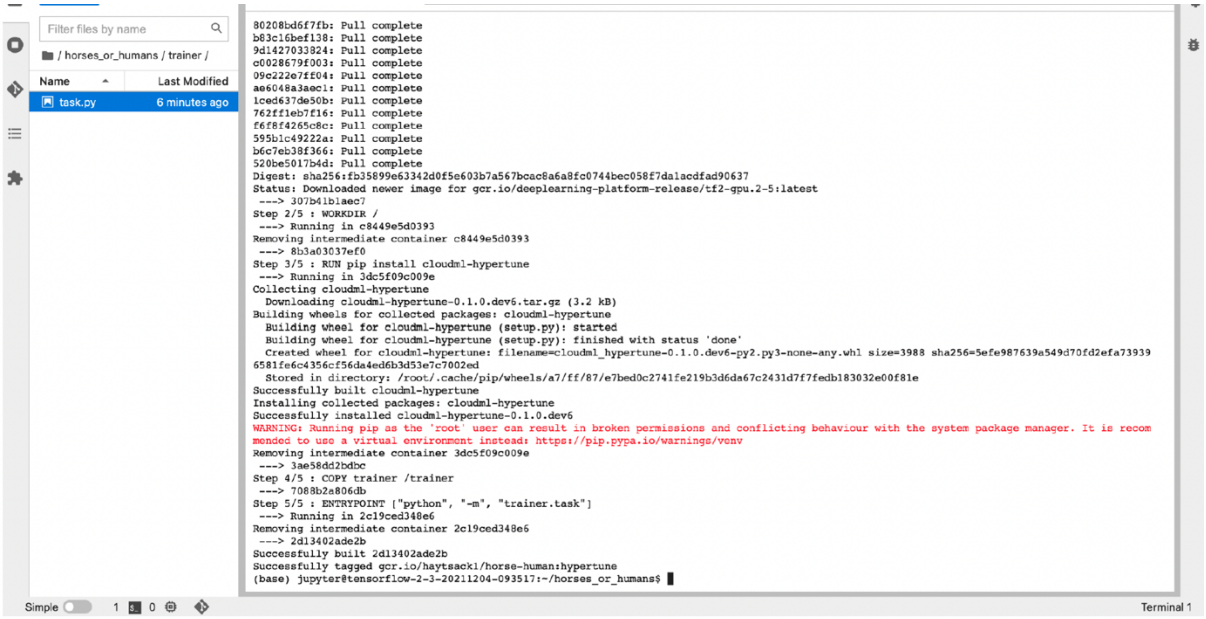




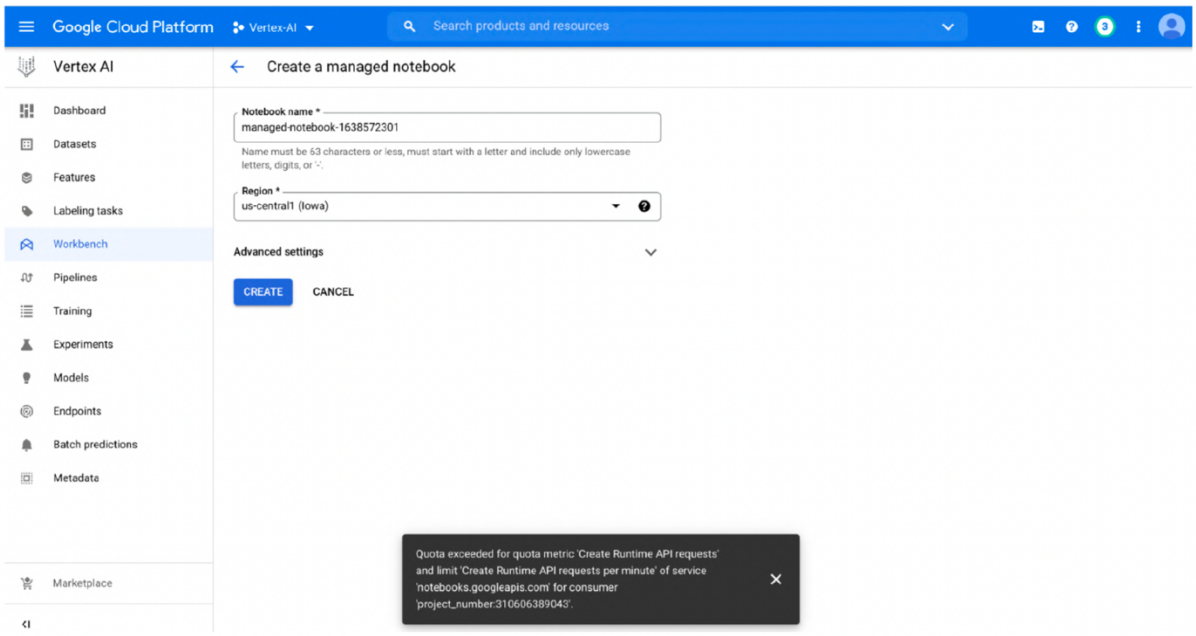
Part f)https://codelabs.developers.google.com/vertex_hyperparameter_tuning#0

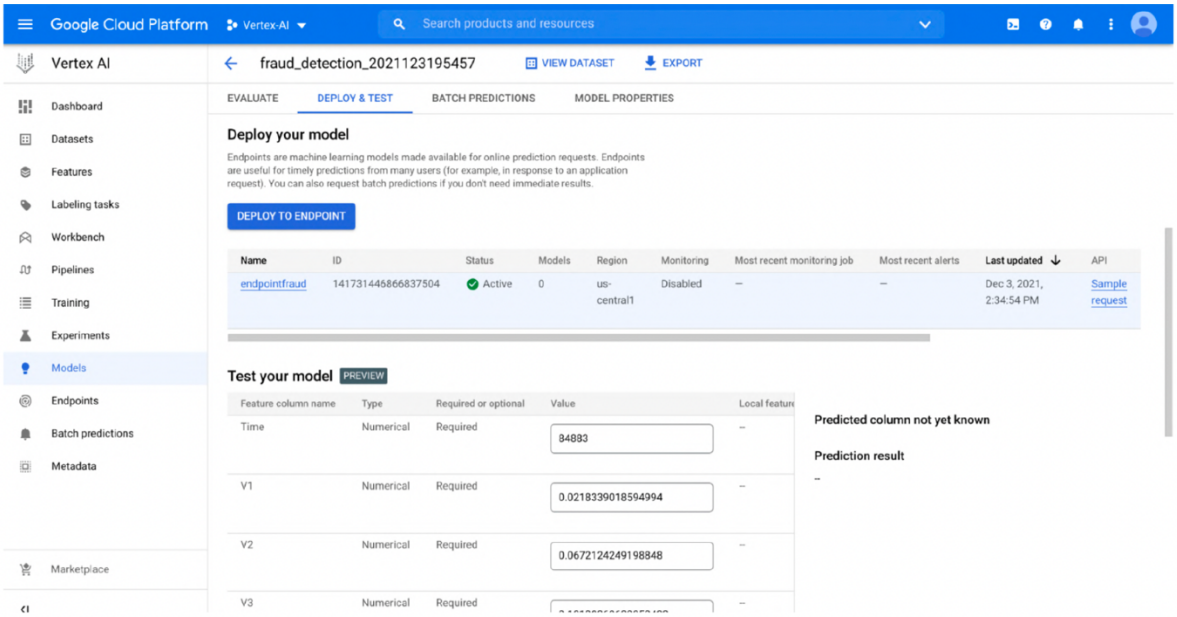

File Edit View Run Kernel Git Tabs Settings Help
Terminal 1
Dockerfile
Dockerfile
Filter files by name
/ horses_or_humans /
Name
Last Modified
Dockerfile
seconds ago
FROM gcr.io/deeplearning-platform-release/tf2-gpu.2-5
WORKDIR /
Installs hypertune library
RUN pip install cloudml-hypertune
Copies the trainer code to the docker image.
COPY trainer /trainer
Sets up the entry point to invoke the trainer.
ENTRYPOINT ["python", "-m", "trainer.task"]


File Edit View Run Kernel Git Tabs Settings Help
Terminal 1
task.py
Dockerfile
Dockerfile
Filter files by name
/ horses_or_humans / trainer /
Name
Last Modified
task.py
a minute ago
def create_model(num_neurons, learning_rate, momentum):
'''Defines and complies model.'''
inputs = tf.keras.Input(shape=(150, 150, 3))
x = tf.keras.layers.Conv2D(16, (3, 3), activation='relu')(inputs)
x = tf.keras.layers.MaxPooling2D((2, 2))(x)
x = tf.keras.layers.Conv2D(32, (3, 3), activation='relu')(x)
x = tf.keras.layers.MaxPooling2D((2, 2))(x)
x = tf.keras.layers.Conv2D(64, (3, 3), activation='relu')(x)
x = tf.keras.layers.MaxPooling2D((2, 2))(x)
x = tf.keras.layers.Flatten()(x)
x = tf.keras.layers.Dense(num_neurons, activation='relu')(x)
outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)
model = tf.keras.Model(inputs, outputs)
model.compile(
loss='binary_crossentropy',
optimizer=tf.keras.optimizers.SGD(learning_rate=learning_rate, momentum=momentum),
metrics=['accuracy'])
return model
def main():
args = get_args()
train_data, validation_data = create_dataset()
model = create_model(args.num_neurons, args.learning_rate, args.momentum)
history = model.fit(train_data, epochs=NUM_EPOCHS, validation_data=validation_data)
# DEFINE METRIC
hp_metric = history.history['val_accuracy'][-1]
hpt = hypertune.HyperTune()
hpt.report_hyperparameter_tuning_metric(
hyperparameter_metric_tag='accuracy',
metric_value=hp_metric,
global_step=NUM_EPOCHS)
if __name__ == "__main__":
main()

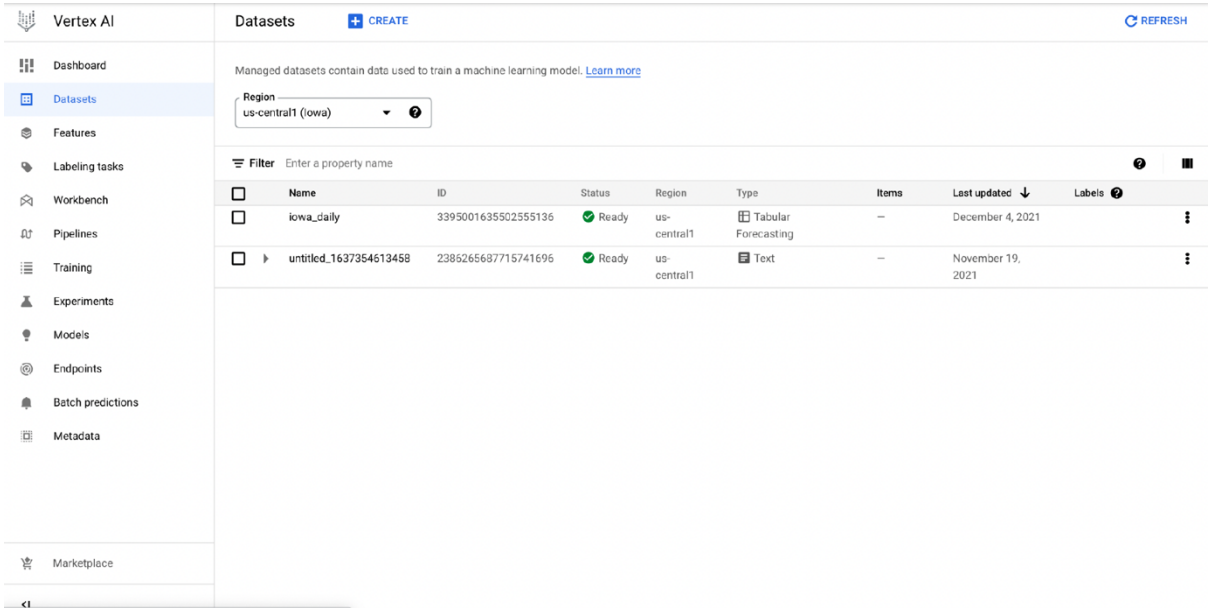


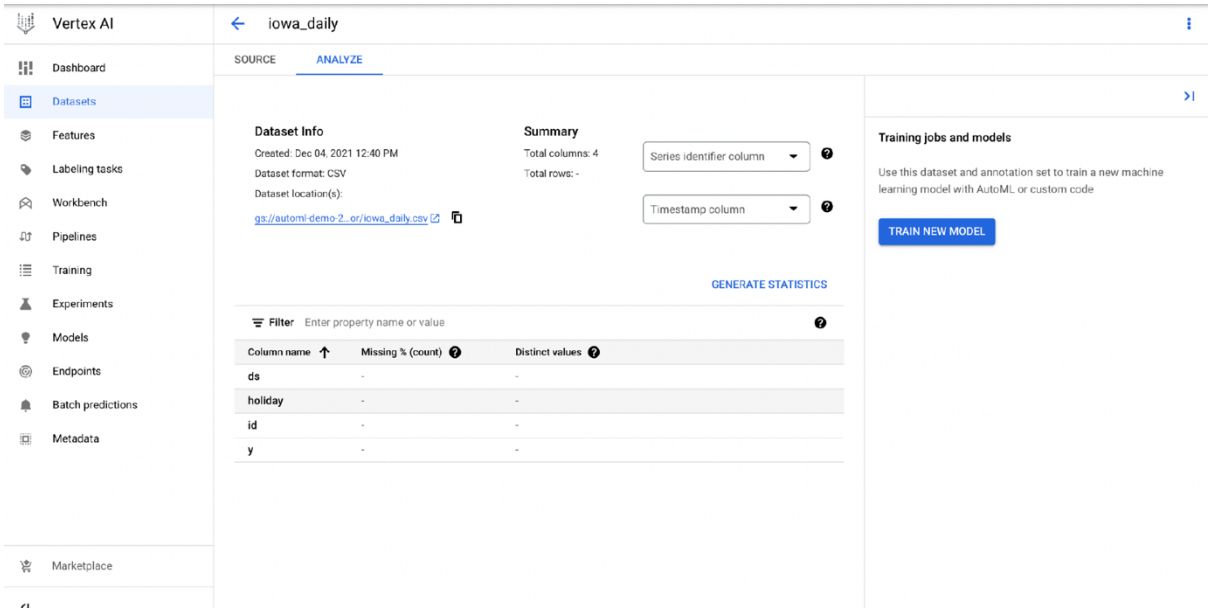
Part g) https://codelabs.developers.google.com/vertex-mlmd-pipelines#0

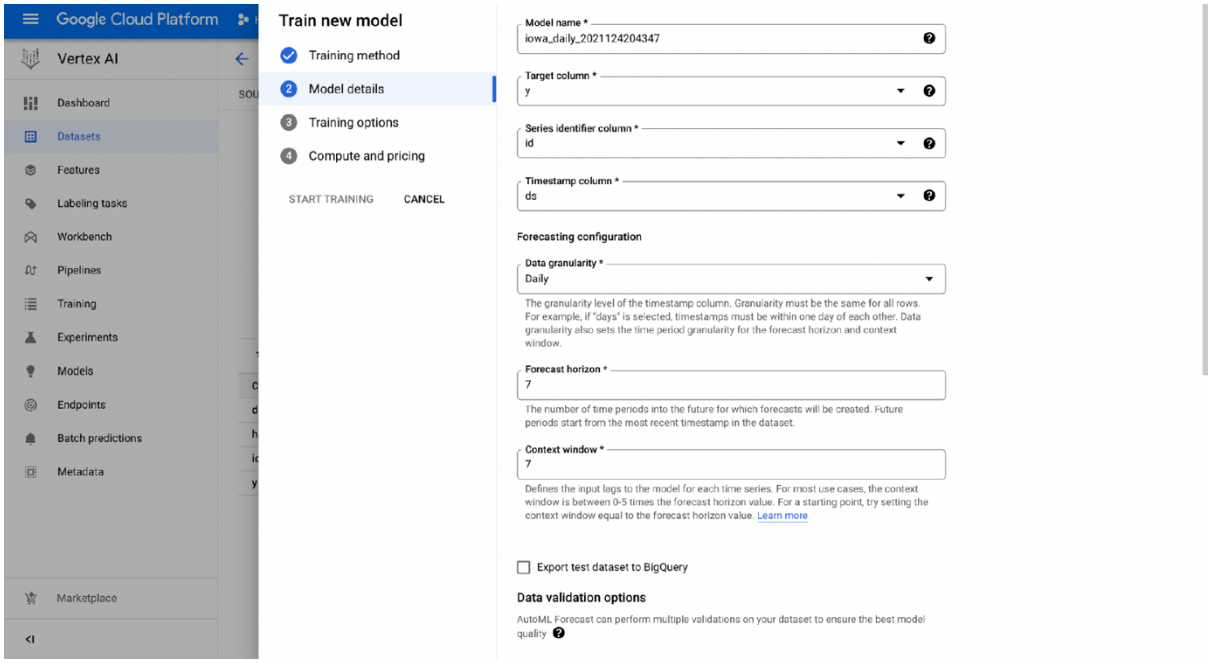


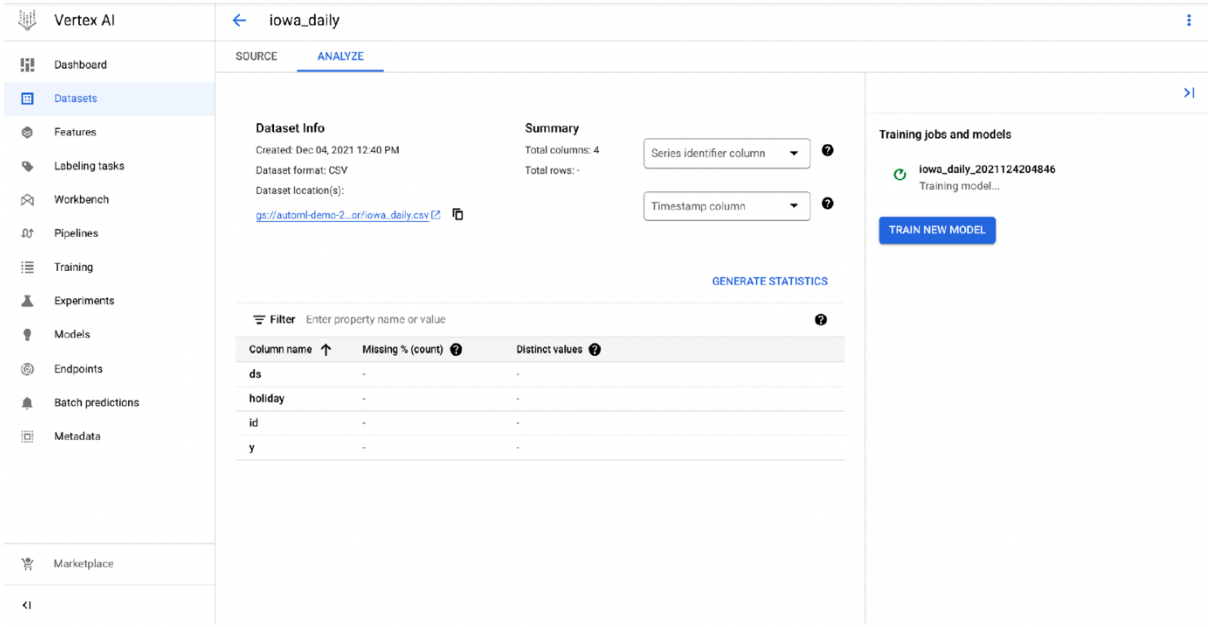


Part h)
https://codelabs.developers.google.com/codelabs/automl-forecasting-with-vertex-ai#0

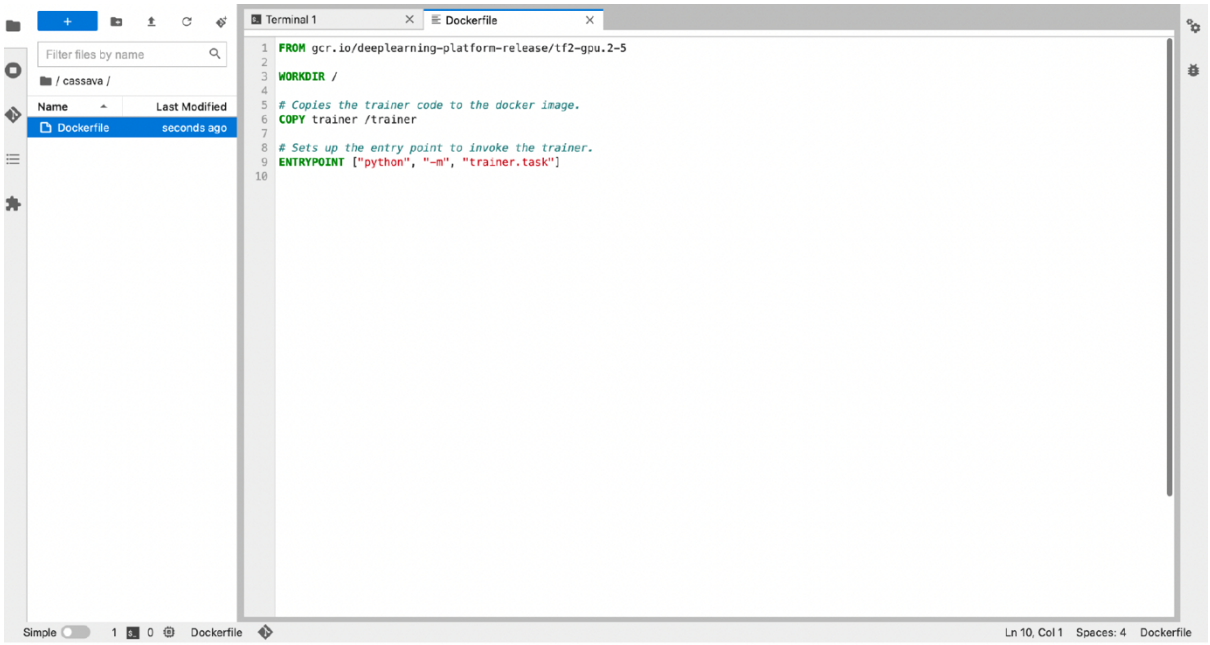


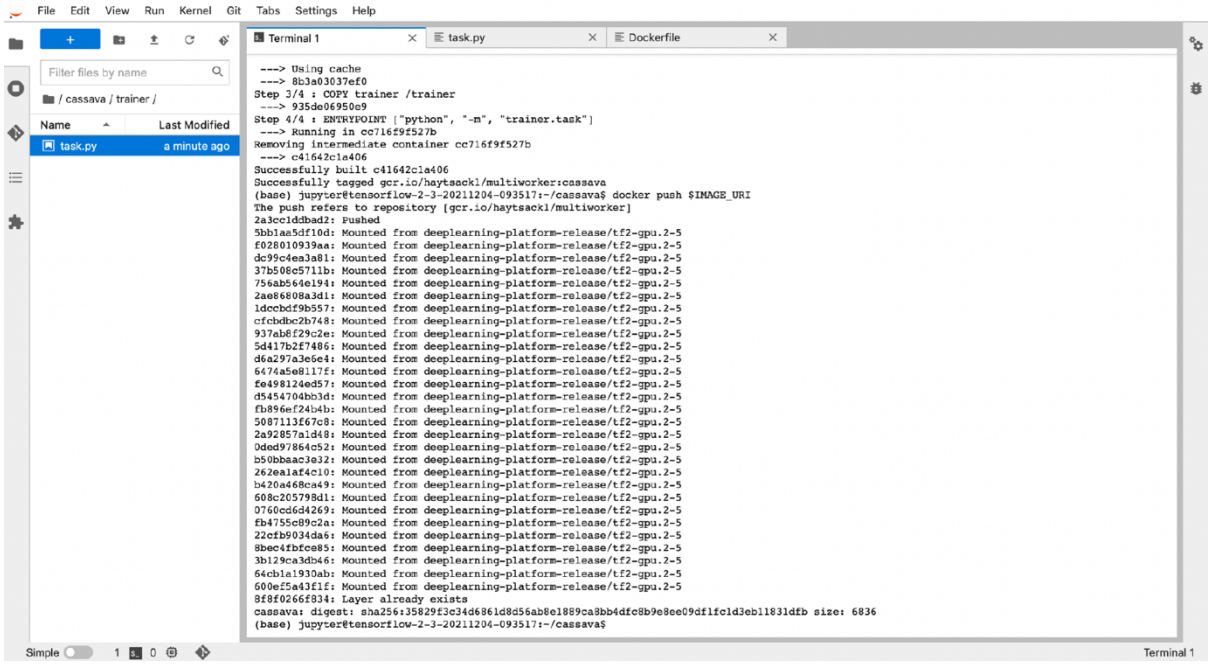

Vertex AI
Dashboard
Datasets
Features
Labeling tasks
Workbench
Pipelines
Training
Experiments
Models
Endpoints
Batch predictions
Metadata
Marketplace
iowa_daily
SOURCE
ANALYZE
Dataset Info
Created: Dec 04, 2021 12:40 PM
Dataset format: CSV
Dataset location(s):
gs://automl-demo-2...or/iowa_daily.csv
Summary
Total columns: 4
Total rows: -
Series identifier column
Timestamp column
GENERATE STATISTICS
Filter
Enter property name or value
Column name
Missing % (count)
Distinct values
ds
holiday
id
y
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

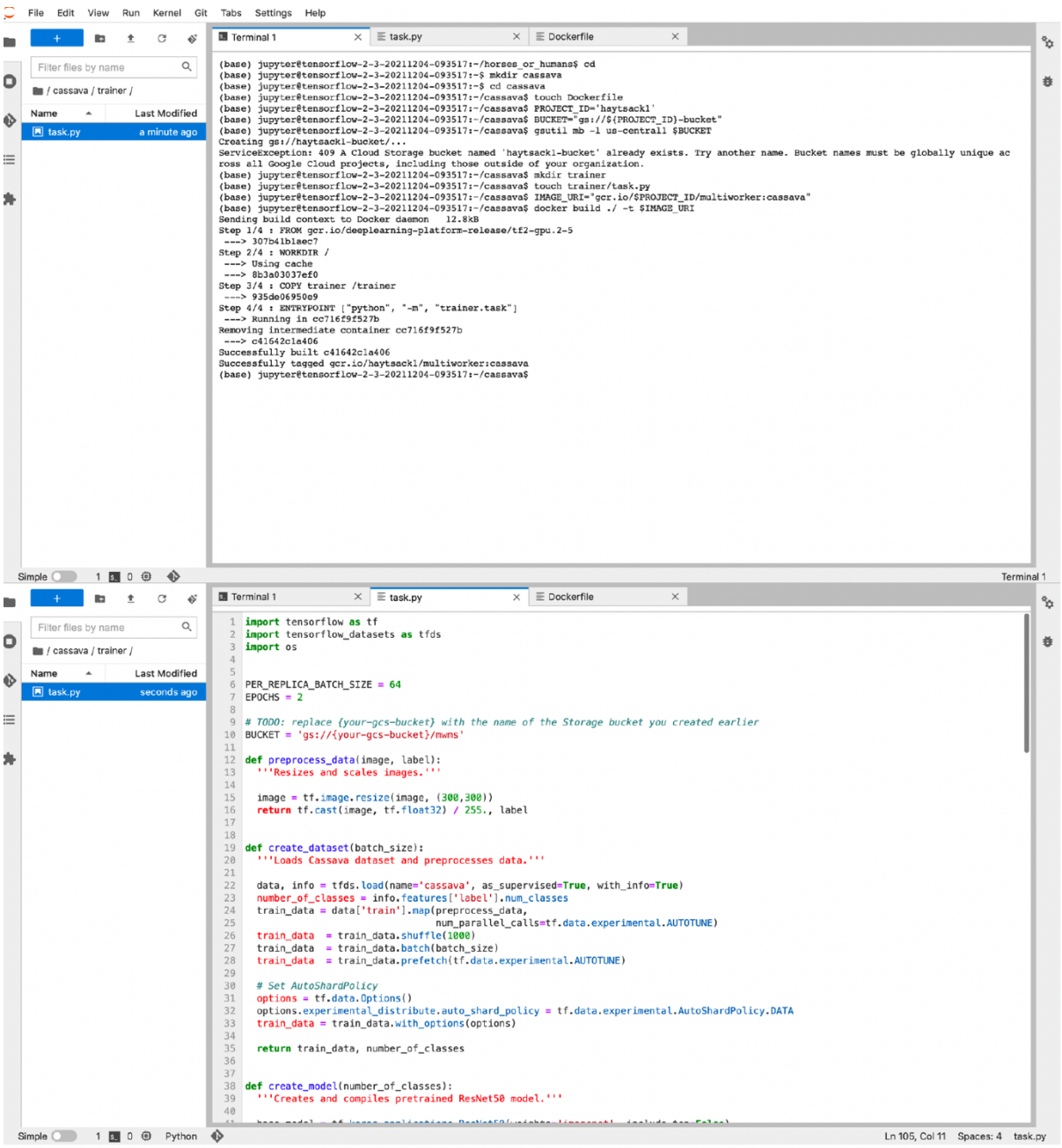




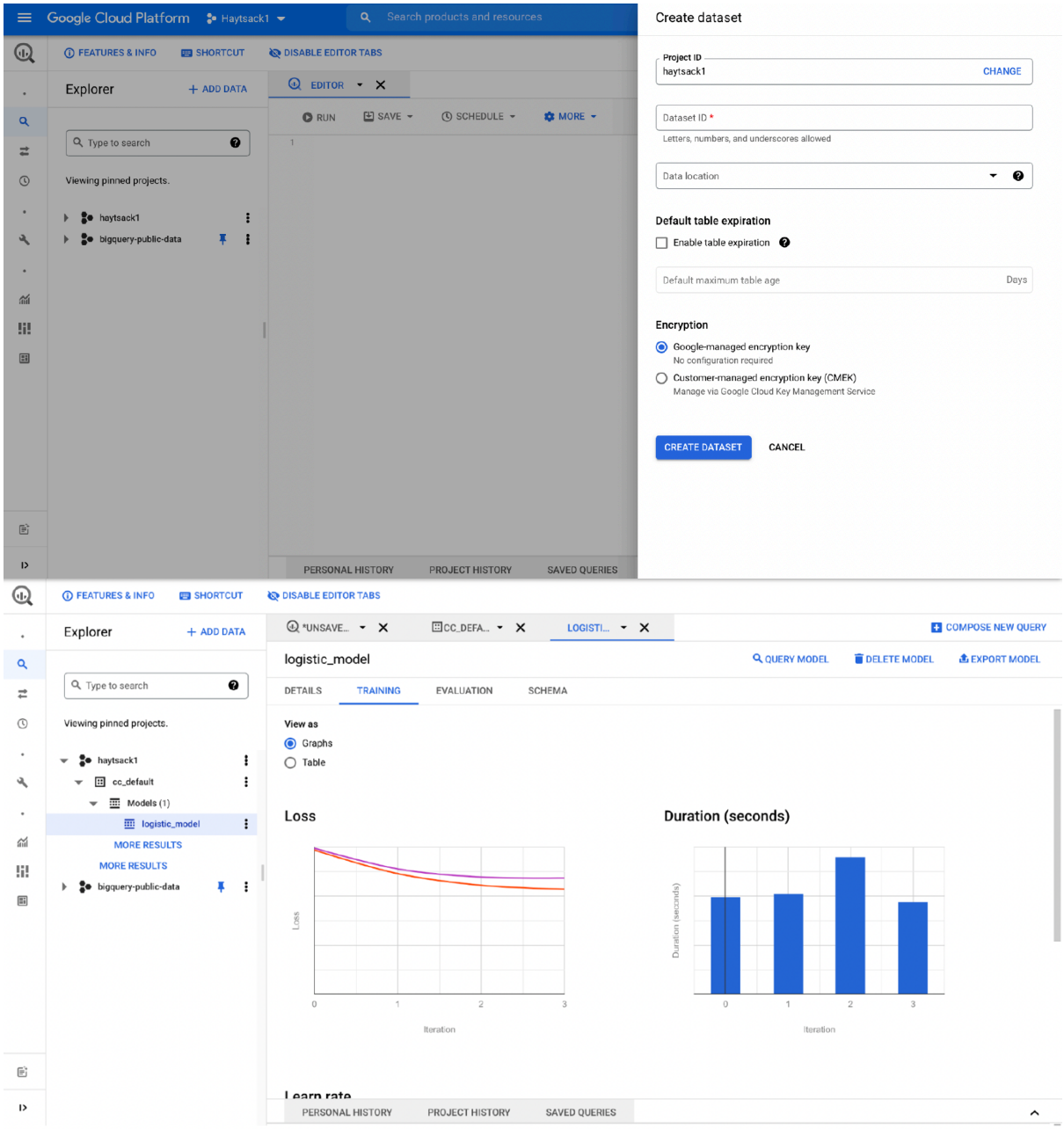
Part i) https://codelabs.developers.google.com/vertex_multiworker_training#0

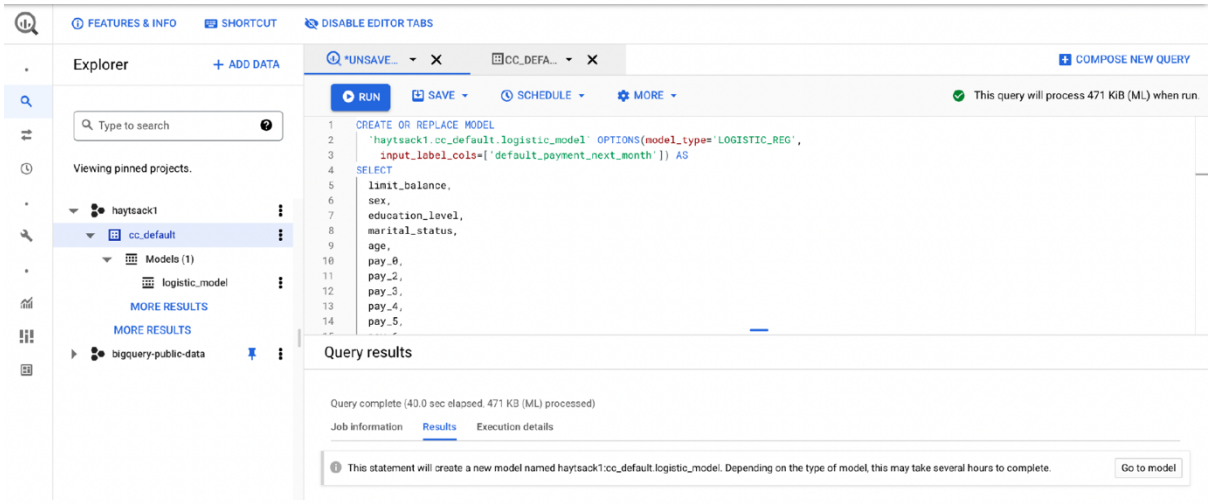

Terminal 1
Dockerfile
Filter files by name
/ cassava /
Name
Last Modified
Dockerfile
seconds ago
1 FROM gcr.io/deeplearning-platform-release/tf2-gpu.2-5
2
3 WORKDIR /
4
5 # Copies the trainer code to the docker image.
6 COPY trainer /trainer
7
8 # Sets up the entry point to invoke the trainer.
9 ENTRYPOINT ["python", "-m", "trainer.task"]
10
Simple
1
0
Dockerfile
Ln 10, Col 1
Spaces: 4
Dockerfile


File
Edit
View
Run
Kernel
Git
Tabs
Settings
Help
Terminal 1
task.py
Dockerfile
Filter files by name
/ cassava / trainer /
Name
Last Modified
task.py
a minute ago
---> Using cache
---> 8b3a03037ef0
Step 3/4 : COPY trainer /trainer
---> 935da06950e9
Step 4/4 : ENTRYPOINT ["python", "-m", "trainer.task"]
---> Running in cc716f9f527b
Removing intermediate container cc716f9f527b
---> c41642c1a406
Successfully built c41642c1a406
Successfully tagged gcr.io/haytsack1/multiworker:cassava
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ docker push $IMAGE_URI
The push refers to repository [gcr.io/haytsack1/multiworker]
2a3cc1d8bad2: Pushed
5bb1aa5df10d: Mounted from deeplearning-platform-release/tf2-gpu.2-5
f028010939aa: Mounted from deeplearning-platform-release/tf2-gpu.2-5
dc99c4ea3a81: Mounted from deeplearning-platform-release/tf2-gpu.2-5
37b508c5711b: Mounted from deeplearning-platform-release/tf2-gpu.2-5
756ab564e194: Mounted from deeplearning-platform-release/tf2-gpu.2-5
2ae86808a3d1: Mounted from deeplearning-platform-release/tf2-gpu.2-5
1dcebdf9b557: Mounted from deeplearning-platform-release/tf2-gpu.2-5
cfcbdbc2b748: Mounted from deeplearning-platform-release/tf2-gpu.2-5
937ab8f29c2e: Mounted from deeplearning-platform-release/tf2-gpu.2-5
5d417b2f7486: Mounted from deeplearning-platform-release/tf2-gpu.2-5
d6a297a3e6e4: Mounted from deeplearning-platform-release/tf2-gpu.2-5
6474a5e8117f: Mounted from deeplearning-platform-release/tf2-gpu.2-5
fe498124ed57: Mounted from deeplearning-platform-release/tf2-gpu.2-5
d5454704bb3d: Mounted from deeplearning-platform-release/tf2-gpu.2-5
fb896ef24b4b: Mounted from deeplearning-platform-release/tf2-gpu.2-5
5087113f67c8: Mounted from deeplearning-platform-release/tf2-gpu.2-5
2a92857a1d48: Mounted from deeplearning-platform-release/tf2-gpu.2-5
0ded97864c52: Mounted from deeplearning-platform-release/tf2-gpu.2-5
b50bbaac3a32: Mounted from deeplearning-platform-release/tf2-gpu.2-5
262ea1af4c10: Mounted from deeplearning-platform-release/tf2-gpu.2-5
b420a468ca49: Mounted from deeplearning-platform-release/tf2-gpu.2-5
608c205798d1: Mounted from deeplearning-platform-release/tf2-gpu.2-5
0760cd6d4269: Mounted from deeplearning-platform-release/tf2-gpu.2-5
fb4755c89c2a: Mounted from deeplearning-platform-release/tf2-gpu.2-5
22cfb9034da6: Mounted from deeplearning-platform-release/tf2-gpu.2-5
8bec4fbfce85: Mounted from deeplearning-platform-release/tf2-gpu.2-5
3b129ca3db46: Mounted from deeplearning-platform-release/tf2-gpu.2-5
64cb1a1930ab: Mounted from deeplearning-platform-release/tf2-gpu.2-5
600ef5a43f1f: Mounted from deeplearning-platform-release/tf2-gpu.2-5
8f8f0266f834: Layer already exists
cassava: digest: sha256:35829f3c34d6861d8d56ab8e1889ca8bb4dfc8b9e8ee09df1fc1d3eb11831dfb size: 6836
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$
Simple
1
0
Terminal 1

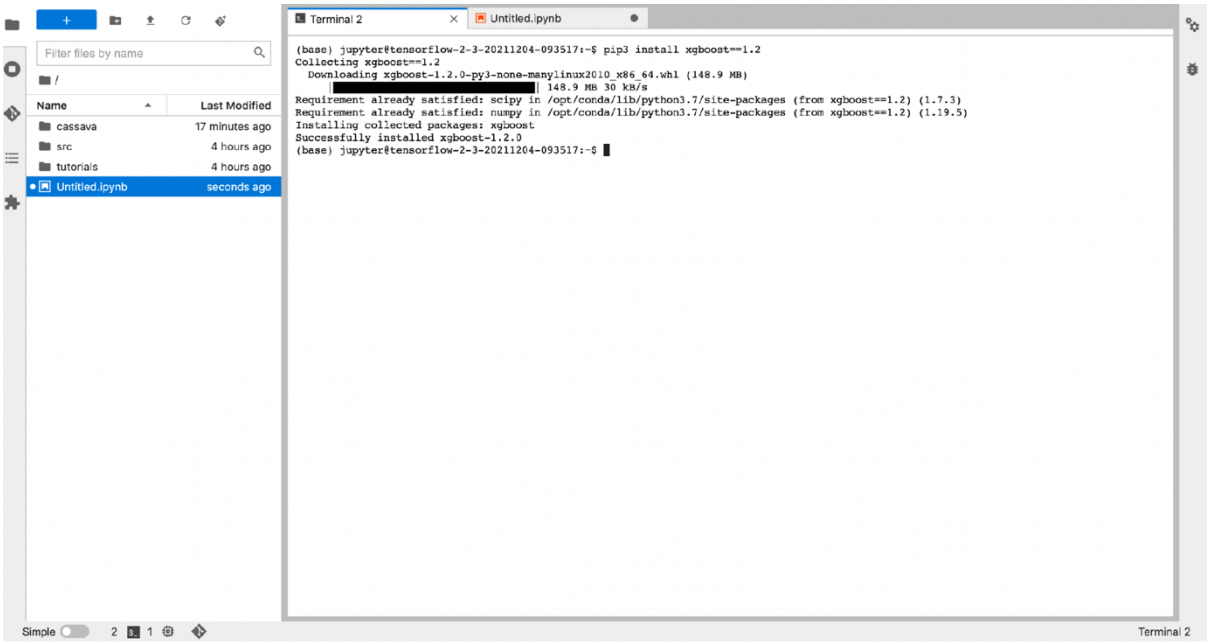


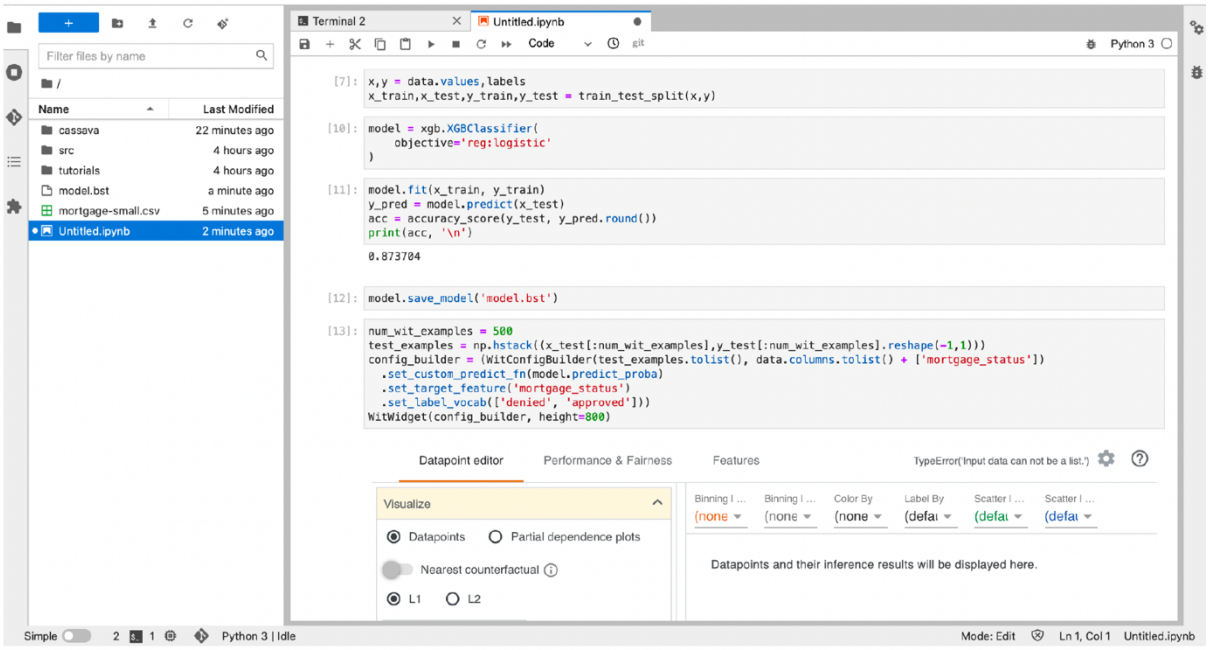
Part j) https://codelabs.developers.google.com/codelabs/bqml-vertex-prediction#0

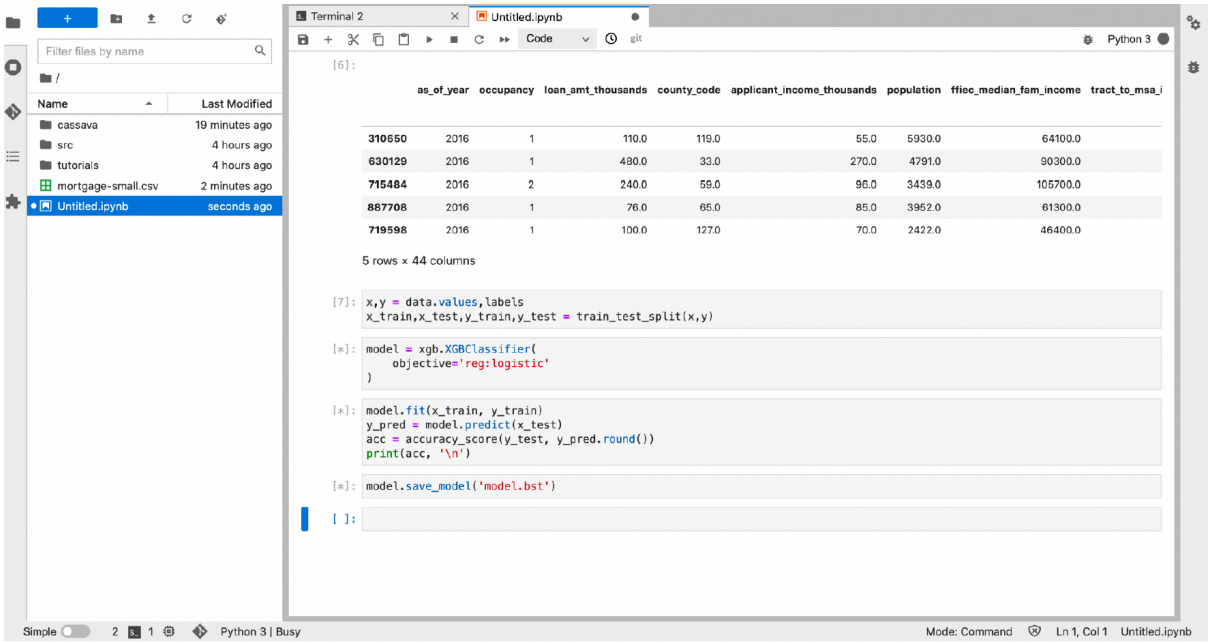

Google Cloud Platform
Haytsack1
Search products and resources
FEATURES & INFO
SHORTCUT
DISABLE EDITOR TABS
Explorer
ADD DATA
Type to search
Viewing pinned projects.
haytsack1
bigquery-public-data
EDITOR
RUN
SAVE
SCHEDULE
MORE
PERSONAL HISTORY
PROJECT HISTORY
SAVED QUERIES
Create dataset
Project ID
haytsack1
CHANGE
Dataset ID *
Letters, numbers, and underscores allowed
Data location
Default table expiration
Enable table expiration
Default maximum table age
Days
Encryption
Google-managed encryption key
No configuration required
Customer-managed encryption key (CMEK)
Manage via Google Cloud Key Management Service
CREATE DATASET
CANCEL
FEATURES & INFO
SHORTCUT
DISABLE EDITOR TABS
Explorer
ADD DATA
Type to search
Viewing pinned projects.
haytsack1
cc_default
Models (1)
logistic_model
MORE RESULTS
MORE RESULTS
bigquery-public-data
*UNSAVE...
CC_DEFA...
LOGISTI...
COMPOSE NEW QUERY
logistic_model
QUERY MODEL
DELETE MODEL
EXPORT MODEL
DETAILS
TRAINING
EVALUATION
SCHEMA
View as
Graphs
Table
Loss
Loss
Iteration
Duration (seconds)
Duration (seconds)
Iteration
Learn rate
PERSONAL HISTORY
PROJECT HISTORY
SAVED QUERIES

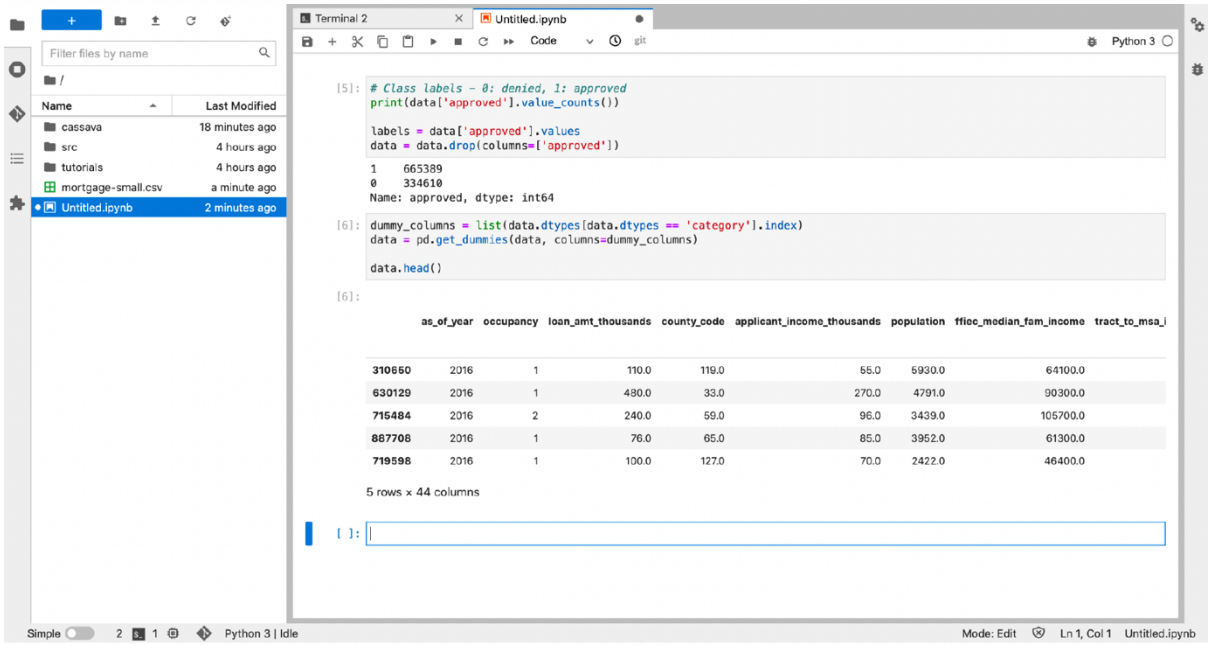


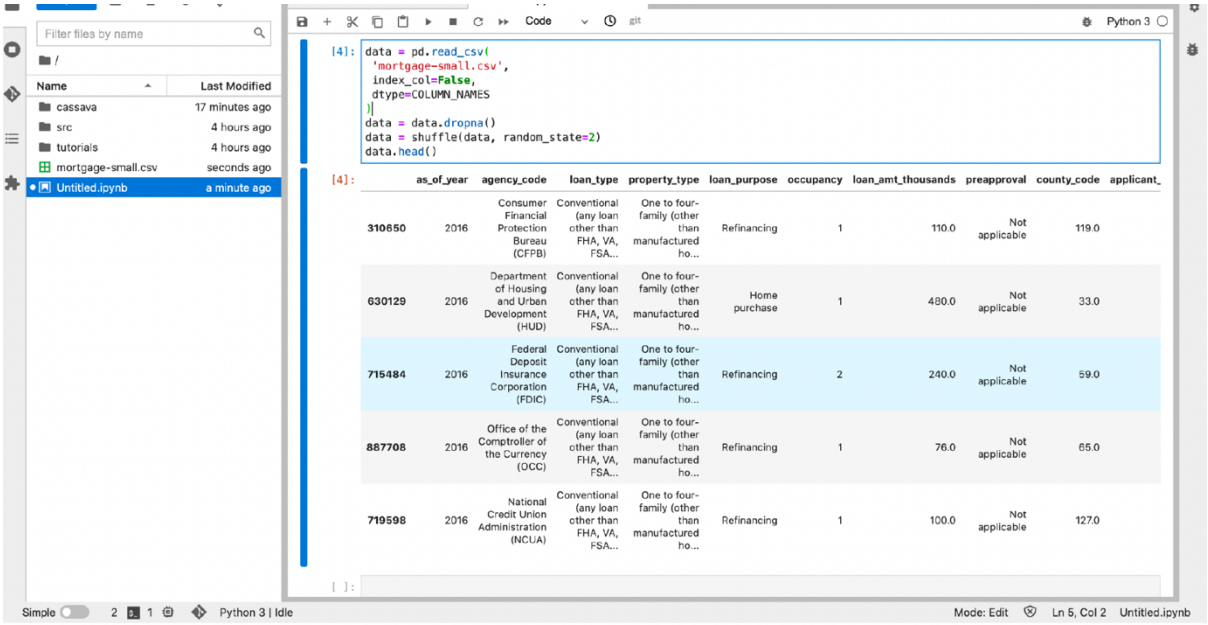
Part k) https://codelabs.developers.google.com/vertex-xgb-wit#0


Terminal 2
Untitled.ipynb
Filter files by name
/
Name
Last Modified
cassava 17 minutes ago
src 4 hours ago
tutorials 4 hours ago
Untitled.ipynb seconds ago
(base) jupyter@tensorflow-2-3-20211204-093517:~$ pip3 install xgboost==1.2
Collecting xgboost==1.2
Downloading xgboost-1.2.0-py3-none-manylinux2010_x86_64.whl (148.9 MB)
148.9 MB 30 kB/s
Requirement already satisfied: scipy in /opt/conda/lib/python3.7/site-packages (from xgboost==1.2) (1.7.3)
Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from xgboost==1.2) (1.19.5)
Installing collected packages: xgboost
Successfully installed xgboost-1.2.0
(base) jupyter@tensorflow-2-3-20211204-093517:~$
Simple
Terminal 2

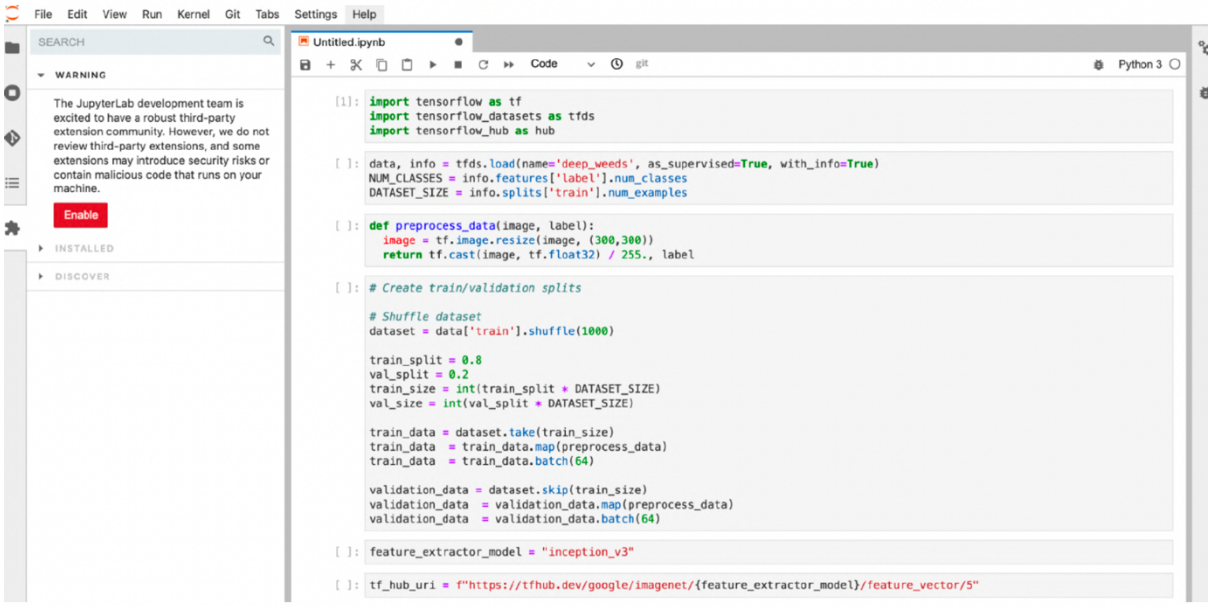

Terminal 2
Untitled.ipynb
Code
Python 3
Filter files by name
/
Name
Last Modified
cassava 22 minutes ago
src 4 hours ago
tutorials 4 hours ago
model.bst a minute ago
mortgage-small.csv 5 minutes ago
Untitled.ipynb 2 minutes ago
[7]: x,y = data.values,labels
x_train,x_test,y_train,y_test = train_test_split(x,y)
[10]: model = xgb.XGBClassifier(
objective='reg:logistic'
)
[11]: model.fit(x_train, y_train)
y_pred = model.predict(x_test)
acc = accuracy_score(y_test, y_pred.round())
print(acc, '\n')
0.873704
[12]: model.save_model('model.bst')
[13]: num_wit_examples = 500
test_examples = np.hstack((x_test[:num_wit_examples],y_test[:num_wit_examples].reshape(-1,1)))
config_builder = (WitConfigBuilder(test_examples.tolist(), data.columns.tolist() + ['mortgage_status'])
.set_custom_predict_fn(model.predict_proba)
.set_target_feature('mortgage_status')
.set_label_vocab(['denied', 'approved']))
WitWidget(config_builder, height=800)
Datapoint editor
Performance & Fairness
Features
TypeError('Input data can not be a list.')
Visualize
Datapoints
Partial dependence plots
Nearest counterfactual
L1
L2
Binning I ...
Binning I ...
Color By
Label By
Scatter I ...
Scatter I ...
(none
(none
(none
(defai
(defai
(defai
Datapoints and their inference results will be displayed here.
Simple
Python 3 | Idle
Mode: Edit
Ln 1, Col 1
Untitled.ipynb

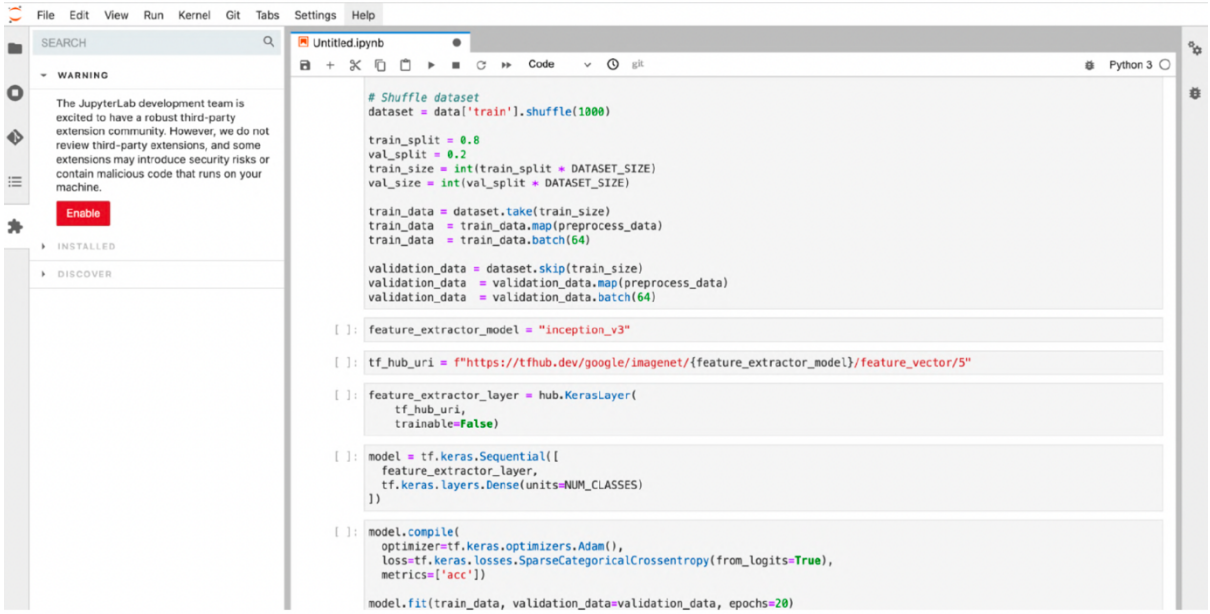

Terminal 2
Untitled.ipynb
Code
Python 3
Filter files by name
/
Name
Last Modified
cassava 19 minutes ago
src 4 hours ago
tutorials 4 hours ago
mortgage-small.csv 2 minutes ago
Untitled.ipynb seconds ago
[6]:
as_of_year occupancy loan_amt_thousands county_code applicant_income_thousands population ffiec_median_fam_income tract_to_msa_i
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
[7]:
x,y = data.values,labels
x_train,x_test,y_train,y_test = train_test_split(x,y)
[*]:
model = xgb.XGBClassifier(
objective='reg:logistic'
)
[*]:
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
acc = accuracy_score(y_test, y_pred.round())
print(acc, '\n')
[*]:
model.save_model('model.bst')
[ ]:
Simple
2 1
Python 3 | Busy
Mode: Command
Ln 1, Col 1
Untitled.ipynb


Terminal 2
Untitled.ipynb
Code
Python 3
Filter files by name
/
Name
Last Modified
cassava 18 minutes ago
src 4 hours ago
tutorials 4 hours ago
mortgage-small.csv a minute ago
Untitled.ipynb 2 minutes ago
[5]:
# 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
[6]:
dummy_columns = list(data.dtypes[data.dtypes == 'category'].index)
data = pd.get_dummies(data, columns=dummy_columns)
data.head()
[6]:
as_of_year occupancy loan_amt_thousands county_code applicant_income_thousands population ffiec_median_fam_income tract_to_msa_i
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
[ ]:
Simple
2 1
Python 3 | Idle
Mode: Edit
Ln 1, Col 1
Untitled.ipynb

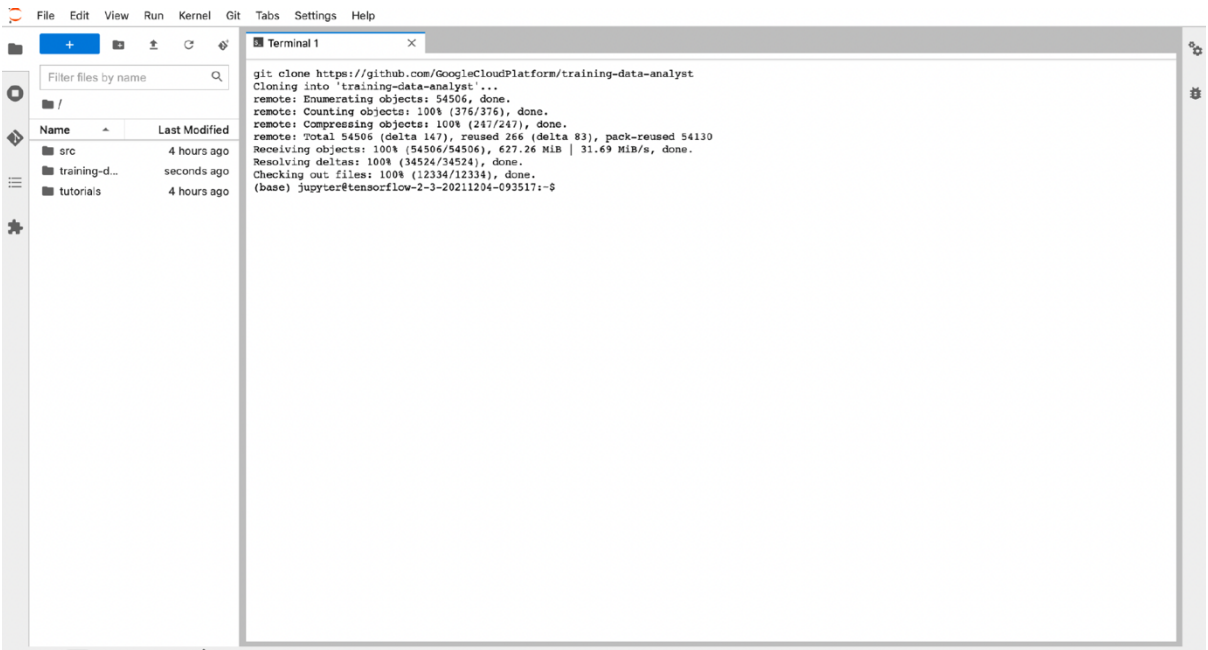


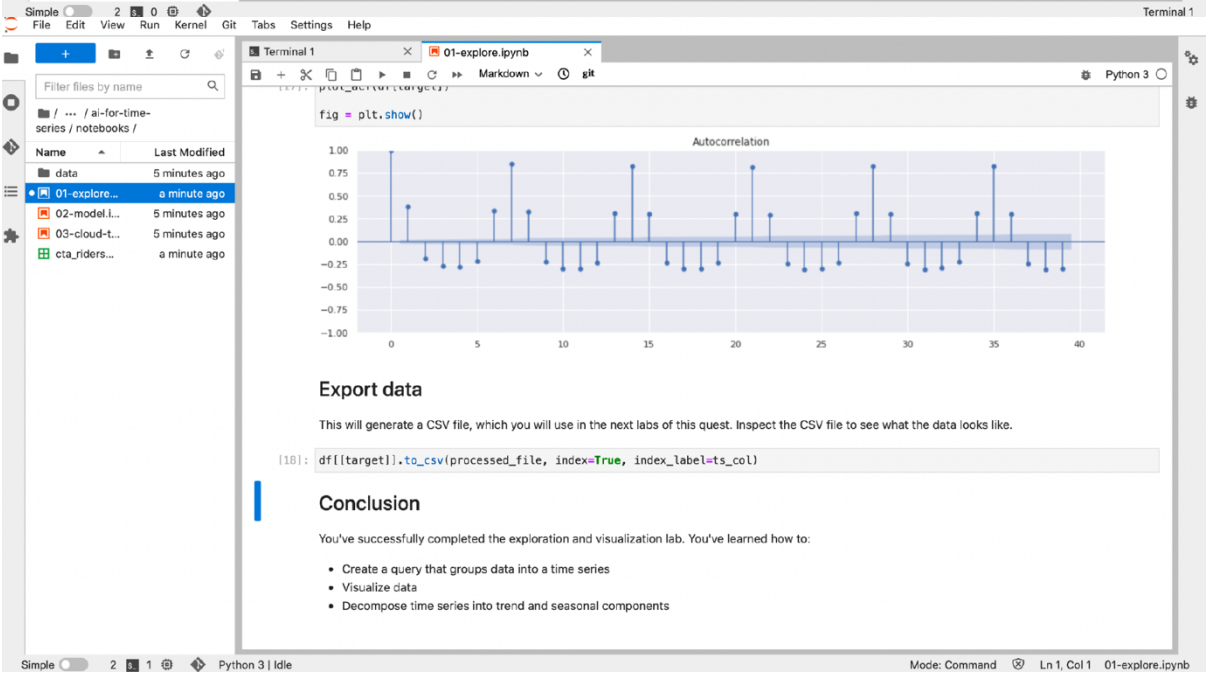
Part 1) https://codelabs.developers.google.com/vertex_notebook_executor#0

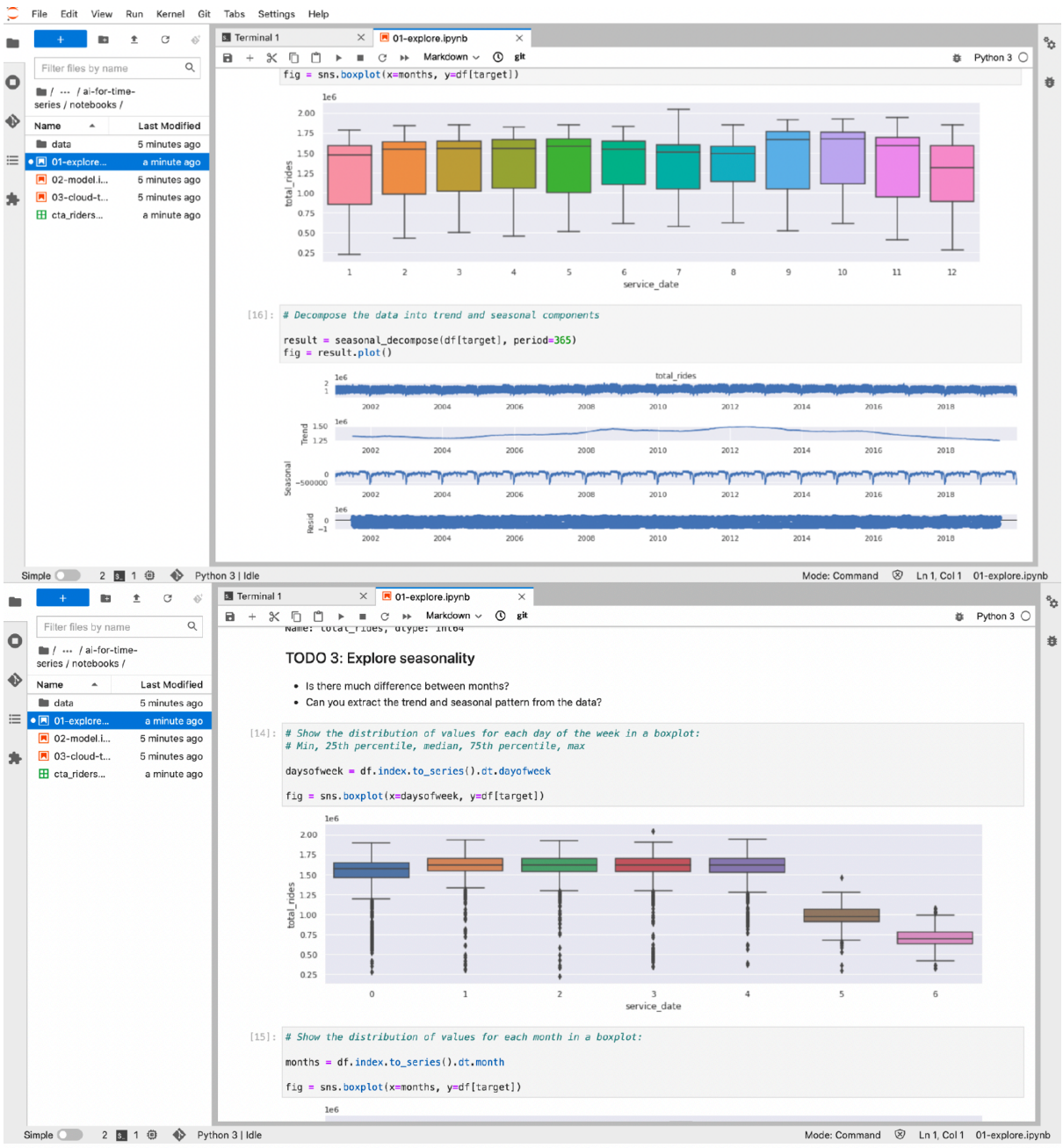




Part m) https://codelabs.developers.google.com/codelabs/time-series-forecasting-with-cloud-ai-platform#0

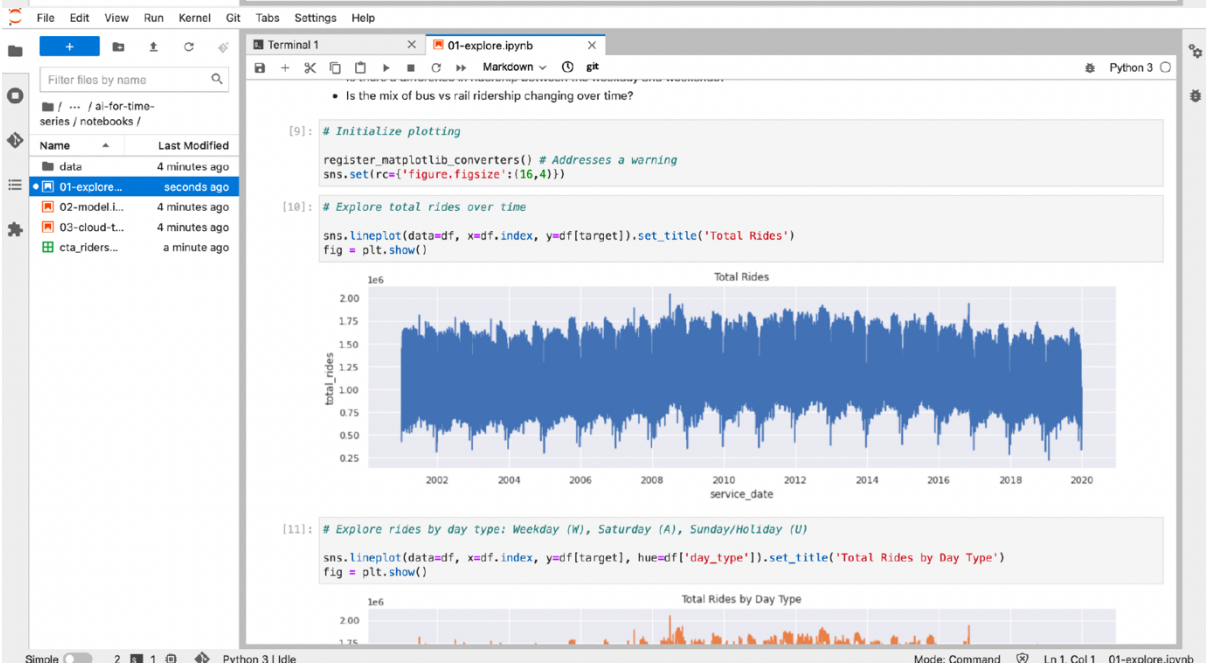

File Edit View Run Kernel Git Tabs Settings Help
Terminal 1
Filter files by name
/
Name
Last Modified
src
4 hours ago
training-d...
seconds ago
tutorials
4 hours ago
git clone https://github.com/GoogleCloudPlatform/training-data-analyst
Cloning into 'training-data-analyst'...
remote: Enumerating objects: 54506, done.
remote: Counting objects: 100% (376/376), done.
remote: Compressing objects: 100% (247/247), done.
remote: Total 54506 (delta 147), reused 266 (delta 83), pack-reused 54130
Receiving objects: 100% (54506/54506), 627.26 MiB | 31.69 MiB/s, done.
Resolving deltas: 100% (34524/34524), done.
Checking out files: 100% (12334/12334), done.
(base) jupyter@tensorflow-2-3-20211204-093517:~$
Simple
2
0
Terminal 1


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
5 minutes ago
01-explore...
a minute ago
02-model.i...
5 minutes ago
03-cloud-t...
5 minutes ago
cta_riders...
a minute ago
fig = plt.show()
Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25
30
35
40
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.
[18]:
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
Simple
2
1
Python 3 | Idle
Mode: Command
Ln 1, Col 1
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 5 minutes ago
01-explore... a minute ago
02-model.i... 5 minutes ago
03-cloud-t... 5 minutes ago
cta_riders... a minute ago
fig = sns.boxplot(x=months, y=df[target])
total_rides
service_date
[16]:
# Decompose the data into trend and seasonal components
result = seasonal_decompose(df[target], period=365)
fig = result.plot()
total_rides
Trend
Seasonal
Resid
Simple
Python 3 | Idle
Mode: Command
Ln 1, Col 1
01-explore.ipynb

Terminal 1
01-explore.ipynb
Markdown
git
Python 3
Filter files by name
/ ... / ai-for-time-series / notebooks /
Name
Last Modified
data 5 minutes ago
01-explore... a minute ago
02-model.i... 5 minutes ago
03-cloud-t... 5 minutes ago
cta_riders... a minute ago
TODO 3: Explore seasonality
Is there much difference between months?
Can you extract the trend and seasonal pattern from the data?
[14]:
# 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])
total_rides
service_date
[15]:
# 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])
Simple
Python 3 | Idle
Mode: Command
Ln 1, Col 1
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 4 minutes ago
01-explore... a minute ago
02-model.i... 4 minutes ago
03-cloud-t... 4 minutes ago
cta_riders... a minute ago
Total Rides by Day Type
1e6
total_rides
day_type
U
W
A
service_date
[12]:
# 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


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 4 minutes ago
01-explore... seconds ago
02-model.i... 4 minutes ago
03-cloud-t... 4 minutes ago
cta_riders... a minute ago
• Is the mix of bus vs rail ridership changing over time?
[9]:
Initialize plotting
register_matplotlib_converters() # Addresses a warning
sns.set(rc={'figure.figsize':(16,4)})
[10]:
# Explore total rides over time
sns.lineplot(data=df, x=df.index, y=df[target]).set_title('Total Rides')
fig = plt.show()
Total Rides
1e6
total_rides
service_date
[11]:
# 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
Simple
Python 3 | Idle
Mode: Command
Ln 1, Col 1
01-explore.ipynb