

# Assignment-7

## VERTEX AI

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

The screenshot shows the Google Cloud Platform Vertex AI interface. On the left, there's a sidebar with options like Dashboard, Datasets, Features, Labeling tasks, Workbench (which is selected), Pipelines, Training, Experiments, Models, Endpoints, Batch predictions, and Metadata. The main area is titled 'Notebooks' and shows a table of managed notebooks. One notebook is listed: 'tensorflow-2-3-20211210-223253'. The table includes columns for Notebook name, Zone, Auto-upgrade, Environment, and Machine type. A note at the top states: 'As of the M80 DLVM release, all environments will include JupyterLab 3 by default. To continue using an existing environment's JupyterLab 1.x version, disable auto-upgrade (if enabled) and do not manually upgrade the environment to a new environment version. To create new Notebooks with JupyterLab 1.x installed, see creating specific versions of Notebooks.' A 'Filter' bar allows searching by property name or value.

The screenshot shows a Jupyter Notebook cell with the file name 'Untitled.ipynb'. The code in the cell is as follows:

```
# You can change the `text` and `emoji_str` parameters here to update the pipeline output
def intro_pipeline(text: str = "Vertex Pipelines", emoji_str: str = "sparkles"):
    product_task = product_name(text)
    emoji_task = emoji(emoji_str)
    consumer_task = build_sentence(
        product_task.output,
        emoji_task.outputs["emoji"],
        emoji_task.outputs["emoji_text"],
    )

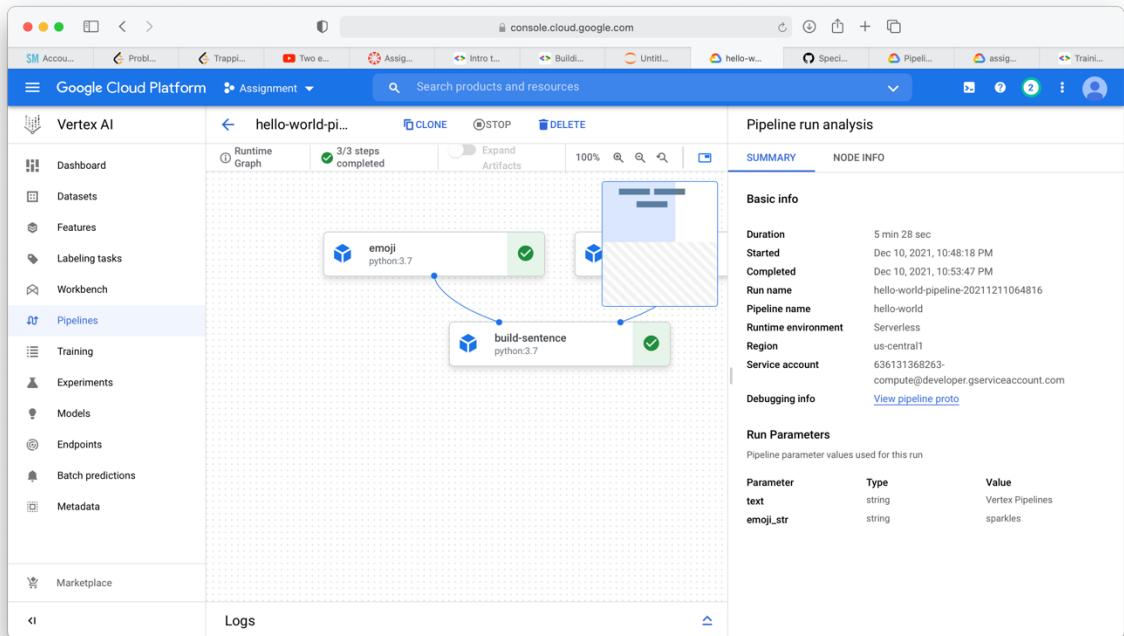
[28]: compiler.Compiler().compile(
    pipeline_func=intro_pipeline,
    package_path="intro_pipeline_job.json"
)

[29]: from datetime import datetime
TIMESTAMP = datetime.now().strftime("%Y%m%d%H%M%S")

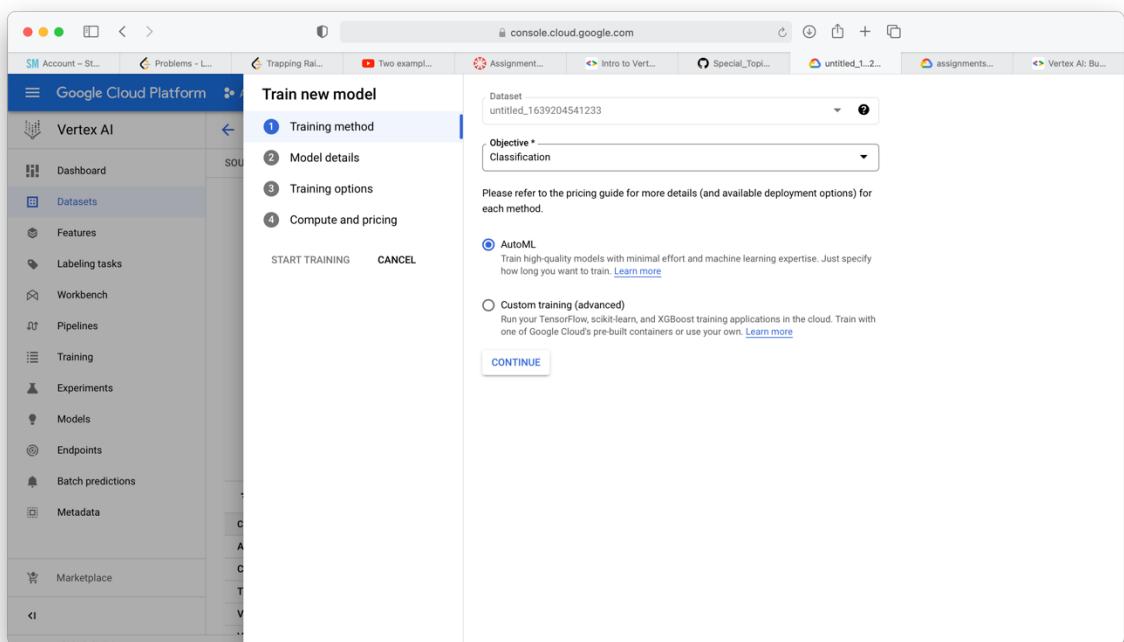
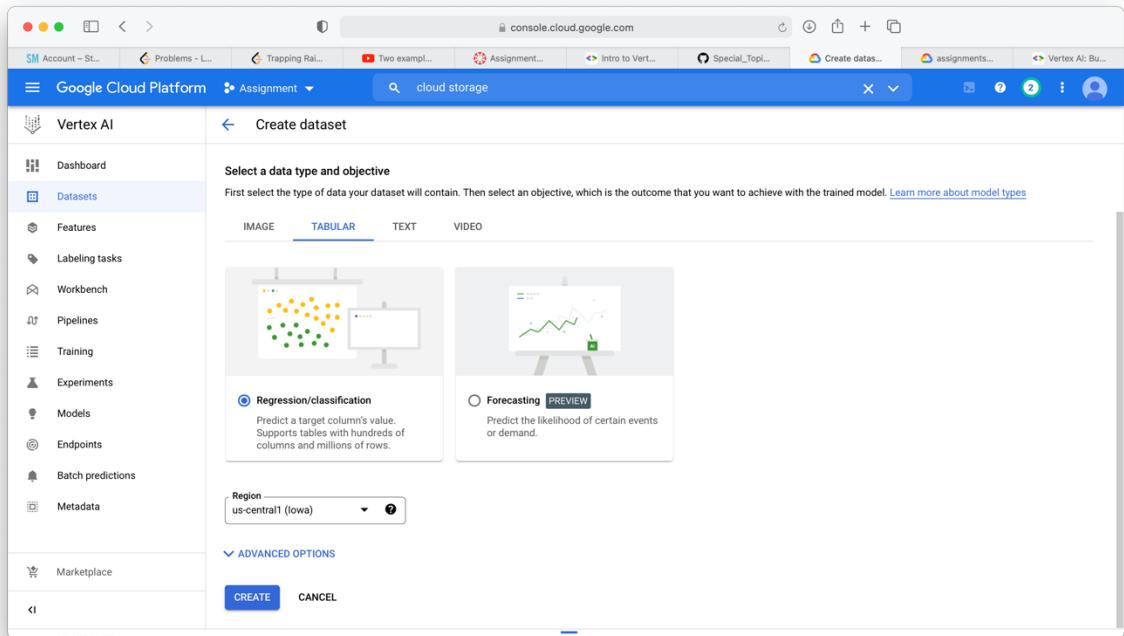
[30]: job = aiplatform.PipelineJob(
    display_name="hello-world-pipeline",
    template_path="intro_pipeline_job.json",
    job_id="hello-world-pipeline-{}".format(TIMESTAMP),
    enable_caching=True
)

[31]: job.submit()
INFO:google.cloud.aiplatform.pipeline_jobs:Creating PipelineJob
INFO:google.cloud.aiplatform.pipeline_jobs:PipelineJob created. Resource name: projects/636131368263/locations/us-central1/pipelineJobs/hello-world-pipeline-20211211064816
INFO:google.cloud.aiplatform.pipeline_jobs:This is the PipelineJob in another session:
INFO:google.cloud.aiplatform.pipeline_jobs: pipeline_job = aiplatform.PipelineJob.get('projects/636131368263/locations/us-central1/pipelineJobs/hello-world-pipeline-20211211064816')
INFO:google.cloud.aiplatform.pipeline_jobs:View Pipeline Job:
https://console.cloud.google.com/vertex-ai/locations/us-central1/pipelines/runs/hello-world-pipeline-20211211064816?projec
```

The cell has a status bar at the bottom indicating 'Simple' mode, 'Python 3 | Idle', and 'Mode: Command'. The status bar also shows 'Ln 1, Col 1' and 'Untitled.ipynb'.

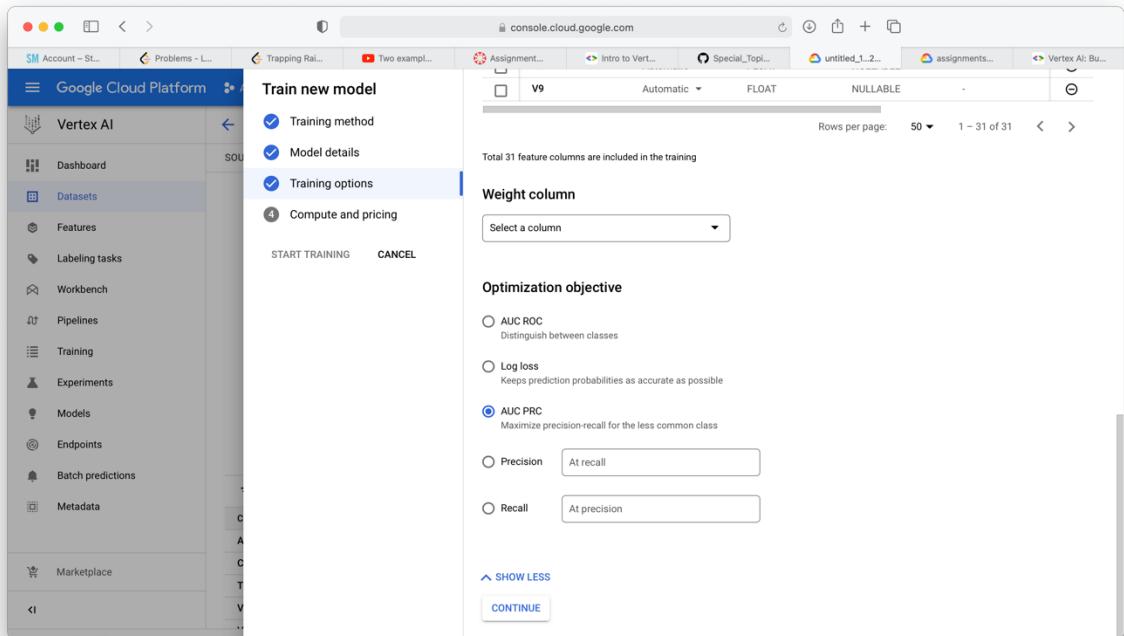


Part b) <https://codelabs.developers.google.com/vertex-automl-tabular#0>

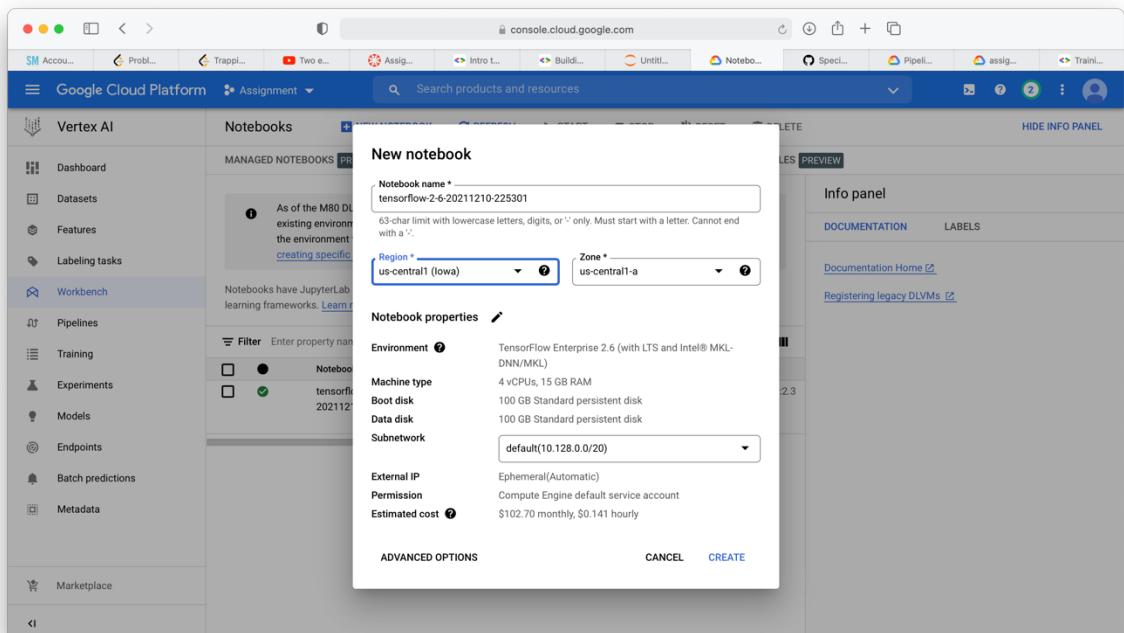


The screenshot shows the Google Cloud Platform Vertex AI interface. On the left, a sidebar menu includes options like Dashboard, Datasets, Features, Labeling tasks, Workbench, Pipelines, Training, Experiments, Models, Endpoints, Batch predictions, and Metadata. The main area is titled "untitled\_1639204541233". It has tabs for "SOURCE" and "ANALYZE". Under "Dataset Info", it shows the dataset was created on Dec 10, 2021 at 10:36 PM, is a BigQuery dataset with 31 columns and 1 row, and is located at [bq://bigquery-public-data\\_fraud\\_detection](bq://bigquery-public-data_fraud_detection). A summary table shows the distribution of data types: FLOAT (30, 96.77%) and INTEGER (1, 3.23%). Below this is a "GENERATE STATISTICS" button. A "Training jobs and models" section includes a "TRAIN NEW MODEL" button.

The screenshot shows the "Train new model" dialog. The sidebar on the left is identical to the previous screenshot. The main dialog has four steps: 1. Training method (selected), 2. Model details, 3. Training options, and 4. Compute and pricing. Step 1 is completed with "Model name \* fraud\_detection" and "Target column \* Class (INTEGER)". Step 2 is partially visible. Step 3 and 4 are collapsed. There is a checkbox for "Export test dataset to BigQuery" which is unchecked. Below the steps is a "CONTINUE" button.



Part c) [https://codelabs.developers.google.com/vertex\\_custom\\_training\\_prediction#0](https://codelabs.developers.google.com/vertex_custom_training_prediction#0)



The screenshot shows a Jupyter Notebook interface with two terminal tabs open. The left terminal tab displays the command to create a Cloud Storage bucket named 'haytsack1-bucket' and the subsequent steps to build a Docker image using the 'train.py' script. The right terminal tab shows the Docker build process, which fails due to a Cloud Storage bucket already existing.

```

ServiceException: 409 A Cloud Storage bucket named 'haytsack1-bucket' already exists. Try another name. Bucket names must be globally unique across all Google Cloud projects, including those outside of your organization.
(base) jupyter@tensorflow-2-3-20211204-093517:~/mpg$ mkdir trainer
(base) jupyter@tensorflow-2-3-20211204-093517:~/mpg$ touch trainer/train.py
(base) jupyter@tensorflow-2-3-20211204-093517:~/mpg$ IMAGE_URI="gcr.io/PROJECT_ID/mpg:v1"
(base) jupyter@tensorflow-2-3-20211204-093517:~/mpg$ docker build ./ -t $IMAGE_URI
Sending build context to Docker daemon 16.9kB
Step 1/4 : FROM gcr.io/deeplearning-platform-release/tf2-cpu.2-6
latest: Pulling from gcr.io/deeplearning-platform-release/tf2-cpu.2-6
7ba5a6b144c: Pull complete
67a3d1735ac5: Pull complete
b99b0854a767: Pull complete
c18f8ddde60a: Pull complete
4f4fb700ef54: Pull complete
c816e0f4b043: Pull complete
71e0e2afbf19: Pull complete
3b5529eb00a2: Pull complete
83b9c8488915: Pull complete
27a6a205064c: Pull complete
c0257bd89738: Pull complete
4f216188a738: Pull complete
b0fe38911501: Pull complete
34c176368ed2: Pull complete
2dt1a8578f32: Pull complete
3fe3b0b4c0ff5: Pull complete
34f948190dc3: Pull complete
7116937aa0d6b: Pull complete
58f5748b03f1: Pull complete
92844a5b80c0: Pull complete
Digest: sha256:d22f66045b83fbff6ea9a7725cdbe85008f76514d72c80fbac8271de810cb74b
Status: Downloaded newer image for gcr.io/deeplearning-platform-release/tf2-cpu.2-6:latest
--> 1b056acc32ef3
Step 2/4 : WORKDIR /
--> Running in e40e4e2f58ec
Removing intermediate container e40e4e2f58ec
--> 6faa502ef2b3
Step 3/4 : COPY trainer /trainer
--> cfa84eff15f
Step 4/4 : RUN ["python", "-m", "trainer.train"]
--> Running in 8ae68598d117
Removing intermediate container 8ae68598d117
--> 4be042358246
Successfully built 4be042358246
Successfully tagged gcr.io/haytsack1/mpg:v1
(base) jupyter@tensorflow-2-3-20211204-093517:~/mpg$ 

```

The screenshot shows a Jupyter Notebook interface with three terminal tabs open. The left terminal tab displays the command to push the Docker image to Google Container Registry. The middle terminal tab shows the training logs for epoch 44, which ends with an error message about sequences being considered as sets. The right terminal tab shows the command to push the Docker image.

```

Epoch 44/1000
8/8 [=====] - 0s 6ms/step - loss: 7.2625 - mae: 1.8994 - mse: 7.2625 - val_loss: 8.8492 - val_mae: 2.2638 - val_mse: 8.8492
Epoch 45/1000
8/8 [=====] - 0s 6ms/step - loss: 7.2998 - mae: 1.9054 - mse: 7.2998 - val_loss: 8.7706 - val_mae: 2.2936 - val_mse: 8.7706
Epoch 46/1000
8/8 [=====] - 0s 7ms/step - loss: 7.0895 - mae: 1.9006 - mse: 7.0895 - val_loss: 9.4730 - val_mae: 2.2412 - val_mse: 9.4730
Epoch 47/1000
8/8 [=====] - 0s 6ms/step - loss: 7.2303 - mae: 1.8992 - mse: 7.2303 - val_loss: 8.9592 - val_mae: 2.2868 - val_mse: 8.9592
Epoch 48/1000
8/8 [=====] - 0s 7ms/step - loss: 7.1156 - mae: 1.8647 - mse: 7.1156 - val_loss: 8.8877 - val_mae: 2.3046 - val_mse: 8.8877
Epoch 49/1000
8/8 [=====] - 0s 6ms/step - loss: 6.9258 - mae: 1.8454 - mse: 6.9258 - val_loss: 8.8305 - val_mae: 2.2941 - val_mse: 8.8305
2021-12-04 20:01:46.981112: W tensorflow/python/util/util.cc:348] Sets are not currently considered sequences, but this may change in the future, so consider avoiding using them.
(base) jupyter@tensorflow-2-3-20211204-093517:~/mpg$ docker push $IMAGE_URI
The push refers to repository [gcr.io/haytsack1/mpg]
73f984dd6466: Pushed
2699358e44d2: Mounted from deeplearning-platform-release/tf2-cpu.2-6
393bde4f6bc5: Mounted from deeplearning-platform-release/tf2-cpu.2-6
cf3738cf85bb1: Mounted from deeplearning-platform-release/tf2-cpu.2-6
1b3cd3815787f: Mounted from deeplearning-platform-release/tf2-cpu.2-6
2446e42c63d1: Mounted from deeplearning-platform-release/tf2-cpu.2-6
2d3b6ee42c63d1: Mounted from deeplearning-platform-release/tf2-cpu.2-6
b538ff5dfbd06: Mounted from deeplearning-platform-release/tf2-cpu.2-6
539edf6b8f68: Mounted from deeplearning-platform-release/tf2-cpu.2-6
f212849c96cc: Mounted from deeplearning-platform-release/tf2-cpu.2-6
7121d6cc04ae: Mounted from deeplearning-platform-release/tf2-cpu.2-6
c787e4bdd37e: Mounted from deeplearning-platform-release/tf2-cpu.2-6
9f6f8d0d96008: Mounted from deeplearning-platform-release/tf2-cpu.2-6
9f6f64009273: Mounted from deeplearning-platform-release/tf2-cpu.2-6
b69e9b09c4e0: Mounted from deeplearning-platform-release/tf2-cpu.2-6
54363f143fd1: Mounted from deeplearning-platform-release/tf2-cpu.2-6
5379358e44d2: Mounted from deeplearning-platform-release/tf2-cpu.2-6
b2b1550e0e98e1: Mounted from deeplearning-platform-release/tf2-cpu.2-6
48515e7cf8a9: Mounted from deeplearning-platform-release/tf2-cpu.2-6
82bb8c1c41f58: Mounted from deeplearning-platform-release/tf2-cpu.2-6
9f54eef41275: Layer already exists
v1: digest: sha256:27fc0864233ebab2d9442c95690d0277bfb296e196a25ae415a024c245e48647 size: 4713
(base) jupyter@tensorflow-2-3-20211204-093517:~/mpg$ 

```

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

**Vertex AI**

[Create a managed notebook](#)

[Enable idle shutdown](#)

Time of inactivity before shutdown (Minutes) \*   
Must be integer: 1-600

**Networking**  
The network must have outbound connection to the internet. [Learn more](#) about the networking options below.

Google-managed networks  
No configuration required

Networks in this project  
Private service access required

Networks shared with me  
Private service access required

**Security**

Enable nbconvert

Enable file downloading from Notebook UI

**Permission**  
Managed notebooks currently only support single user only mode. Single user only mode restricts access to the user specified below.

Owner \*

Quota exceeded for quota metric 'Create Runtime API requests' and limit 'Create Runtime API requests per minute' of service 'notebooks.googleapis.com' for consumer 'project\_number:840891042430'. [X](#)

[CREATE](#) [CANCEL](#)

**Vertex AI**

[Create a managed notebook](#)

Notebook name \*  
Name must be 63 characters or less, must start with a letter and include only lowercase letters, digits, or '-'.

Region \*  [?](#)

**Advanced settings** [▼](#)

[CREATE](#) [CANCEL](#)

Part f) [https://codelabs.developers.google.com/vertex\\_hyperparameter\\_tuning#0](https://codelabs.developers.google.com/vertex_hyperparameter_tuning#0)

A screenshot of a Jupyter Notebook interface. On the left is a file browser showing a directory structure with a 'Dockerfile' file selected. In the center, there are two tabs: 'Terminal 1' and 'Dockerfile'. The 'Dockerfile' tab contains the following code:

```
1 FROM gcr.io/deeplearning-platform-release/tf2-gpu.2-5
2 WORKDIR /
3 # Installs hypertune library
4 RUN pip install cloudml-hypertune
5 # Copies the trainer code to the docker image.
6 COPY trainer /trainer
7 # Sets up the entry point to invoke the trainer.
8 ENTRYPOINT ["python", "-m", "trainer.task"]
```

A screenshot of a Jupyter Notebook interface. On the left is a file browser showing a directory structure with a 'task.py' file selected. In the center, there are three tabs: 'Terminal 1', 'task.py', and 'Dockerfile'. The 'task.py' tab contains the following code:

```
54
55
56 def create_model(num_neurons, learning_rate, momentum):
57     '''Defines and compiles model.'''
58
59     inputs = tf.keras.Input(shape=(150, 150, 3))
60     x = tf.keras.layers.Conv2D(16, (3, 3), activation='relu')(inputs)
61     x = tf.keras.layers.MaxPooling2D((2, 2))(x)
62     x = tf.keras.layers.Conv2D(32, (3, 3), activation='relu')(x)
63     x = tf.keras.layers.MaxPooling2D((2, 2))(x)
64     x = tf.keras.layers.Conv2D(64, (3, 3), activation='relu')(x)
65     x = tf.keras.layers.MaxPooling2D((2, 2))(x)
66     x = tf.keras.layers.Flatten()(x)
67     x = tf.keras.layers.Dense(num_neurons, activation='relu')(x)
68     outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)
69     model = tf.keras.Model(inputs, outputs)
70     model.compile(
71         loss='binary_crossentropy',
72         optimizer=tf.keras.optimizers.SGD(learning_rate=learning_rate, momentum=momentum),
73         metrics=['accuracy'])
74     return model
75
76
77 def main():
78     args = get_args()
79     train_data, validation_data = create_dataset()
80     model = create_model(args.num_neurons, args.learning_rate, args.momentum)
81     history = model.fit(train_data, epochs=NUM_EPOCHS, validation_data=validation_data)
82
83     # DEFINE METRIC
84     hp_metric = history.history['val_accuracy'][-1]
85
86     hpt = hypertune.HyperTune()
87     hpt.report_hyperparameter_tuning_metric(
88         hyperparameter_metric_tag='accuracy',
89         metric_value=hp_metric,
90         global_step=NUM_EPOCHS)
91
92
93 if __name__ == "__main__":
94     main()
```

```

Filter files by name
Search: / horses_or_humans / trainer /
Name Last Modified
task.py 6 minutes ago

80208bd6f7fb: Pull complete
b33c1ghsf138: Pull complete
9d427033324: Pull complete
c028679f003: Pull complete
09c22e7f04: Pull complete
ae6048a3aec1: Pull complete
lced37de508e: Pull complete
762ff1eb7f16: Pull complete
f6f8f42465c0e: Pull complete
5321c49222a: Pull complete
5321c49222a: Pull complete
5321c49222a: Pull complete
5321c49222a: Pull complete
Digest: sha256:f2b5899e63342d0f5e603b7a567bcac8a6a9fc0744bec058f7dalacfad90637
Status: Downloaded newer image for gcr.io/deeplearning-platform-release/tf2-gpu.2-5:latest
--> 307b41b1aec7
Step 2/5 : WORKDIR /
--> Running in c8449e5d0393
Removing intermediate container c8449e5d0393
--> 8b3a03037ef0
Step 3/5 : RUN pip install cloudml-hypertune
--> Running in 3dc569a900c6
Collecting cloudml-hypertune
  Downloading cloudml-hypertune-0.1.0.dev6.tar.gz (3.2 kB)
Building wheels for collected packages: cloudml-hypertune
  Building wheel for cloudml-hypertune (setup.py): started
  Building wheel for cloudml-hypertune (setup.py): finished with status 'done'
    Created wheel for cloudml-hypertune: filename=cloudml_hypertune-0.1.0.dev6-py2.py3-none-any.whl size=3988 sha256=5efe987639a549d70fd2efa73939
6581fe6c4356cf56da4ed63d3d53e7c7002ed
  Stored in directory: /root/.cache/pip/wheels/a7/ff/87/e7bed0c2741fe219b3d6da67c2431d7f7fdb183032e0f81e
Successfully installed cloudml-hypertune-0.1.0.dev6
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Removing intermediate container 3dc569a900c6
--> 3ae58dd2bdbc
Step 4/5 : COPY trainer /trainer
--> 7088b2a806db
Step 5/5 : RUN python -m trainer.task
--> Running in 2c19ced348e6
Removing intermediate container 2c19ced348e6
--> 2d13402ade2b
Successfully built 2d13402ade2b
Successfully tagged gcr.io/haystack1/horse-human:hypertune
(base) jupyter@tensorflow-2-3-20211204-093517:/horses_or_humans$ 

```

Simple 1 0 Terminal 1

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

Google Cloud Platform Vertex AI

**Create a managed notebook**

Vertex AI

- Dashboard
- Datasets
- Features
- Labeling tasks
- Workbench**
- Pipelines
- Training
- Experiments
- Models
- Endpoints
- Batch predictions
- Metadata

Notebook name \* managed-notebook-1638572301  
Name must be 63 characters or less, must start with a letter and include only lowercase letters, digits, or '-'.

Region \* us-central1 (Iowa)

Advanced settings

**CREATE** CANCEL

Quota exceeded for quota metric 'Create Runtime API requests' and limit 'Create Runtime API requests per minute' of service notebooks.googleapis.com for consumer project\_number:310606389043.

Google Cloud Platform Vertex AI Search products and resources

**fraud\_detection\_2021123195457**

EVALUATE DEPLOY & TEST BATCH PREDICTIONS MODEL PROPERTIES

**Deploy your model**

Endpoints are machine learning models made available for online prediction requests. Endpoints are useful for timely predictions from many users (for example, in response to an application request). You can also request batch predictions if you don't need immediate results.

**DEPLOY TO ENDPOINT**

Name	ID	Status	Models	Region	Monitoring	Most recent monitoring job	Most recent alerts	Last updated	API
endpointfraud	141731446866837504	Active	0	us-central1	Disabled	—	—	Dec 3, 2021, 2:34:54 PM	Sample request

**Test your model** PREVIEW

Feature column name	Type	Required or optional	Value	Local feature
Time	Numerical	Required	84883	—
V1	Numerical	Required	0.0218339018594994	—
V2	Numerical	Required	0.0672124249198848	—
V3	Numerical	Required	.....	—

Predicted column not yet known

Prediction result

## Part h)

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

Vertex AI Datasets CREATE REFRESH

Managed datasets contain data used to train a machine learning model. [Learn more](#)

Region us-central1 (iowa) Filter Enter a property name

Name	ID	Status	Region	Type	Items	Last updated	Labels
iowa_daily	3395001635502555136	Ready	us-central1	Tabular Forecasting	—	December 4, 2021	...
untitled_1637354613458	2385265687715741696	Ready	us-central1	Text	—	November 19, 2021	...

Vertex AI

↳ 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](gs://automl-demo-2/or/iowa_daily.csv)

**Summary**

Total columns: 4  
Total rows: -

Series identifier column:

Timestamp column:

**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**

**GENERATE STATISTICS**

Filter Enter property name or value

Column name ↑	Missing % (count) ⓘ	Distinct values ⓘ
ds	-	-
holiday	-	-
id	-	-
y	-	-

Marketplace

**Train new model**

**Training method** (checked)

**Model details** (selected)

**Training options**

**Compute and pricing**

**START TRAINING**   **CANCEL**

**Model name \***: iowa\_daily\_2021124204347

**Target column \***: y

**Series identifier column \***: id

**Timestamp column \***: ds

**Forecasting configuration**

**Data granularity \***: Daily

The granularity level of the timestamp column. Granularity must be the same for all rows. For example, if "days" is selected, timestamps must be within one day of each other. Data granularity also sets the time period granularity for the forecast horizon and context window.

**Forecast horizon \***: 7

The number of time periods into the future for which forecasts will be created. Future periods start from the most recent timestamp in the dataset.

**Context window \***: 7

Defines the input lags to the model for each time series. For most use cases, the context window is between 0-5 times the forecast horizon value. For a starting point, try setting the context window equal to the forecast horizon value. [Learn more](#)

Export test dataset to BigQuery

**Data validation options**

AutoML Forecast can perform multiple validations on your dataset to ensure the best model quality. [?](#)

**Vertex AI**

**iowa\_daily**

**SOURCE**   **ANALYZE** (selected)

**Dataset Info**

Created: Dec 04, 2021 12:40 PM  
Dataset format: CSV  
Dataset location(s): [gs://automl-demo-2\\_0r/iowa\\_daily.csv](gs://automl-demo-2_0r/iowa_daily.csv)

**Summary**

Total columns: 4  
Total rows: -

**Series identifier column**: id

**Timestamp column**: ds

**GENERATE STATISTICS**

**Filter**: Enter property name or value

Column name	Missing % (count)	Distinct values
ds	-	-
holiday	-	-
id	-	-
y	-	-

**Training jobs and models**

**iowa\_daily\_2021124204846**  
Training model...

**TRAIN NEW MODEL**

Part i) [https://codelabs.developers.google.com/vertex\\_multiworker\\_training#0](https://codelabs.developers.google.com/vertex_multiworker_training#0)

Terminal 1 Dockerfile

```

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

```

--> Using cache
--> 833a03037ef0
Step 3/4 : COPY trainer /trainer
--> 935de6950c9
Step 4/4 : ENTRYPOINT ["python", "-m", "trainer.task"]
--> Removing intermediate container cc716f9f527b
--> c41542c1a06
Successfully built c41542c1a06
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]
2a3cc1dbd2: Pushed
5bb1a55df10d: Mounted from deeplearning-platform-release/tf2-gpu.2-5
f028010939a: Mounted from deeplearning-platform-release/tf2-gpu.2-5
d523a33939a: Mounted from deeplearning-platform-release/tf2-gpu.2-5
37a506e571b: Mounted from deeplearning-platform-release/tf2-gpu.2-5
35ab564e194: Mounted from deeplearning-platform-release/tf2-gpu.2-5
2aa86808a3d1: Mounted from deeplearning-platform-release/tf2-gpu.2-5
1dcfcdf9b557: Mounted from deeplearning-platform-release/tf2-gpu.2-5
cfchdhbc2b748: Mounted from deeplearning-platform-release/tf2-gpu.2-5
937abff29c2e: Mounted from deeplearning-platform-release/tf2-gpu.2-5
5d417bf2f486: Mounted from deeplearning-platform-release/tf2-gpu.2-5
d6a297a3e5e1: Mounted from deeplearning-platform-release/tf2-gpu.2-5
6474ac8117f: Mounted from deeplearning-platform-release/tf2-gpu.2-5
fe1422a5d74: Mounted from deeplearning-platform-release/tf2-gpu.2-5
83545704bd4: Mounted from deeplearning-platform-release/tf2-gpu.2-5
fb396ef2414b: Mounted from deeplearning-platform-release/tf2-gpu.2-5
5087113e67c0: 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
b50bb8ac3e32: Mounted from deeplearning-platform-release/tf2-gpu.2-5
262ea1af4c10: Mounted from deeplearning-platform-release/tf2-gpu.2-5
b4270a468ca949: Mounted from deeplearning-platform-release/tf2-gpu.2-5
608c205798d1: Mounted from deeplearning-platform-release/tf2-gpu.2-5
0933a333a: Mounted from deeplearning-platform-release/tf2-gpu.2-5
4b1755c0992a: Mounted from deeplearning-platform-release/tf2-gpu.2-5
22fbfb034da6: Mounted from deeplearning-platform-release/tf2-gpu.2-5
8bec4fbfc085: Mounted from deeplearning-platform-release/tf2-gpu.2-5
3b129aca3d046: 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
8ff0f266f6834: Layer already exists
cassava: digest: sha256:35829f3c34d6861d8d56ab8e1889ca8b4dfc8b9e8e09df1fc1d3eb11831dfb size: 6836
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$
```

Simple 1 0 Dockerfile Terminal 1

The screenshot shows a Jupyter Notebook environment with two terminals and a file browser.

**File Browser:**

- Path: / cassava / trainer /
- Selected file: task.py
- Last Modified: a minute ago

**Terminal 1 (Dockerfile):**

```
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ cd
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ touch Dockerfile
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ PROJECT_ID="haystack1"
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ BUCKET="gs://${PROJECT_ID}-bucket"
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ gutil mb -l us-central1 $BUCKET
Creating gs://haystack1-bucket...
ServiceException: 409 A Cloud Storage bucket named 'haystack1-bucket' already exists. Try another name. Bucket names must be globally unique across all Google Cloud projects, including those outside of your organization.
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ mkdir trainer
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ touch trainer/task.py
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ IMAGE_URI="gcr.io/${PROJECT_ID}/multiworker:cassava"
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$ docker build ./ -t $IMAGE_URI
Sending build context to Docker daemon 12.8kB
Step 1/4 : FROM gcr.io/deeplearning-platform-release/tf2-gpu.2-5
--> 3075a1b1aec7
Step 2/4 : WORKDIR /
--> Using cache
--> 8b3a03037efc
Step 3/4 : COPY trainer /trainer
--> 935d900460a9
Step 4/4 : ENTRYPOINT ["python", "-m", "trainer.task"]
--> Running in cc71642c1a406
Successfully built c41642c1a406
Successfully tagged gcr.io/haystack1/multiworker:cassava
(base) jupyter@tensorflow-2-3-20211204-093517:~/cassava$
```

**Terminal 2 (task.py):**

```
1 import tensorflow as tf
2 import tensorflow_datasets as tfds
3 import os
4
5 PER_REPLICA_BATCH_SIZE = 64
6 EPOCHS = 2
7
8 # TODO: replace {your-gcs-bucket} with the name of the Storage bucket you created earlier
9 BUCKET = 'gs://{your-gcs-bucket}/mnms'
10
11 def preprocess_data(image, label):
12     '''Resizes and scales images.'''
13
14     image = tf.image.resize(image, (300,300))
15     return tf.cast(image, tf.float32) / 255., label
16
17
18 def create_dataset(batch_size):
19     '''Loads Cassava dataset and preprocesses data.'''
20
21     data, info = tfds.load(name='cassava', as_supervised=True, with_info=True)
22     number_of_classes = info.features['label'].num_classes
23     train_data = data['train'].map(preprocess_data,
24                                  num_parallel_calls=tf.data.experimental.AUTOTUNE)
25     train_data = train_data.shuffle(1000)
26     train_data = train_data.batch(batch_size)
27     train_data = train_data.prefetch(tf.data.experimental.AUTOTUNE)
28
29     # Set AutoShardPolicy
30     options = tf.data.Options()
31     options.experimental_distribute.auto_shard_policy = tf.data.experimental.AutoShardPolicy.DATA
32     train_data = train_data.with_options(options)
33
34     return train_data, number_of_classes
35
36
37 def create_model(number_of_classes):
38     '''Creates and compiles pretrained ResNet50 model.'''
39
40     base_model = tf.keras.applications.ResNet50(weights='imagenet', include_top=False)
```

Ln 105, Col 11 Spaces: 4 task.py

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

Google Cloud Platform Haystack1 Search products and resources

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

**Editor** RUN SAVE SCHEDULE MORE

**Explorer** + ADD DATA

Viewing pinned projects.

- haystack1
- bigrquery-public-data

### Create dataset

Project ID haystack1 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 (CMK) Manage via Google Cloud Key Management Service

CREATE DATASET CANCEL

FEATURES & INFO SHORTCUT DISABLE EDITOR TABS

+ UNSAVE... CC\_DEFAL... LOGISTI...

COMPOSE NEW QUERY

**Explorer** + ADD DATA

Viewing pinned projects.

- haystack1
- cc\_default
- Models (1)
  - logistic\_model
- bigrquery-public-data

QUERY MODEL DELETE MODEL EXPORT MODEL

**logistic\_model**

DETAILS TRAINING EVALUATION SCHEMA

**View as**

Graphs

Table

**Loss**

Iteration	Loss
0	0.75
1	0.45
2	0.35
3	0.25

**Duration (seconds)**

Iteration	Duration (seconds)
0	0.5
1	0.55
2	0.75
3	0.5

The screenshot shows the Google Cloud BigQuery interface. On the left, the sidebar displays pinned projects, with 'haystack1' selected. Under 'haystack1', the 'cc.default' dataset is selected, showing a single 'logistic.model' table. The main area is a query editor with the following SQL code:

```
1 CREATE OR REPLACE MODEL
2   `haystack1.cc_default.logistic_model` OPTIONS(model_type='LOGISTIC_REG',
3     input_label_cols=['default_payment_next_month']) AS
4   SELECT
5     limit_balance,
6     sex,
7     education_level,
8     marital_status,
9     age,
10    pay_0,
11    pay_2,
12    pay_3,
13    pay_4,
14    pay_5,
```

The status bar at the top right indicates: "This query will process 471 kB (ML) when run". Below the code, the results tab is selected, showing the message: "Query complete (40.0 sec elapsed, 471 kB (ML) processed)". A note below the results states: "This statement will create a new model named haystack1:cc\_default.logistic\_model. Depending on the type of model, this may take several hours to complete." There is also a link "Go to model".

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

File tree:

```
Filter files by name
Name Last Modified
/ 
cassava 17 minutes ago
src 4 hours ago
tutorials 4 hours ago
Untitled.ipynb seconds ago
```

Terminal 2:

```
(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:~$
```

File tree:

```
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
```

Terminal 2:

```
[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')
0.873784

[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)
```

Python 3 | Idle

Datapoint editor

Visualize

- Datapoints
- Partial dependence plots
- Nearest counterfactual
- L1
- L2

Performance & Fairness

Features

TypeError('Input data can not be a list.')

Binning I ... Binning I ... Color By Label By Scatter I ... Scatter I ...
 (none) (none) (none) (defal) (defal) (defal)

Data points and their inference results will be displayed here.

Mode: Edit Ln 1, Col 1 Untitled.ipynb

File browser sidebar:

```
[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	490.0	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	3962.0	61300.0	
719598	2016	1	100.0	127.0	70.0	2422.0	46400.0	

5 rows x 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

File browser sidebar:

```
[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	490.0	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	3962.0	61300.0	
719598	2016	1	100.0	127.0	70.0	2422.0	46400.0	

5 rows x 44 columns

```
[ ]:
```

Simple 2 1 Python 3 | Idle Mode: Edit ⌘ Ln 1, Col 1 Untitled.ipynb

The screenshot shows a Jupyter Notebook interface with a file browser on the left and a code editor/terminal on the right.

**File Browser:**

- Filter files by name: /
- Name: Last Modified
- cassava (17 minutes ago)
- src (4 hours ago)
- tutorial's (4 hours ago)
- mortgage-small.csv (seconds ago)
- Untitled.ipynb (a minute ago)

**Code Editor (Cell 4):**

```

[4]: data = pd.read_csv(
    'mortgage-small.csv',
    index_col=False,
    dtype=COLUMN_NAMES
)
data = data.dropna()
data = shuffle(data, random_state=2)
data.head()

```

**Data Preview (Cell 4 output):**

	as_of_year	agency_code	loan_type	property_type	loan_purpose	occupancy	loan_amt_thousands	preapproval	county_code	applicant...
310850	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 (HUD)	Conventional (any loan other than FHA, VA, FSA...)	One to four-family (other than manufactured ho...)	Home purchase	1	480.0	Not applicable	33.0	
715484	2016	Federal Deposit Insurance Corporation (FDIC)	Conventional (any loan other than FHA, VA, FSA...)	One to four-family (other than manufactured ho...)	Refinancing	2	240.0	Not applicable	59.0	
887708	2016	Office of the Comptroller of the Currency (OCC)	Conventional (any loan other than FHA, VA, FSA...)	One to four-family (other than manufactured ho...)	Refinancing	1	76.0	Not applicable	65.0	
719598	2016	National Credit Union Administration (NCUA)	Conventional (any loan other than FHA, VA, FSA...)	One to four-family (other than manufactured ho...)	Refinancing	1	100.0	Not applicable	127.0	

Part 1) [https://codelabs.developers.google.com/vertex\\_notebook\\_executor#0](https://codelabs.developers.google.com/vertex_notebook_executor#0)

```

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 Python 3
[1]: import tensorflow as tf
import tensorflow_datasets as tfds
import tensorflow_hub as hub

[2]: data, info = tfds.load(name='deep_weeds', as_supervised=True, with_info=True)
NUM_CLASSES = info.features['label'].num_classes
DATASET_SIZE = info.splits['train'].num_examples

[3]: def preprocess_data(image, label):
    image = tf.image.resize(image, (300,300))
    return tf.cast(image, tf.float32) / 255., label

[4]: # Create train/validation splits

# Shuffle dataset
dataset = data['train'].shuffle(1000)

train_split = 0.8
val_split = 0.2
train_size = int(train_split * DATASET_SIZE)
val_size = int(val_split * DATASET_SIZE)

train_data = dataset.take(train_size)
train_data = train_data.map(preprocess_data)
train_data = train_data.batch(64)

validation_data = dataset.skip(train_size)
validation_data = validation_data.map(preprocess_data)
validation_data = validation_data.batch(64)

[5]: feature_extractor_model = "inception_v3"

[6]: tf_hub_uri = f"https://tfhub.dev/google/imagenet/{feature_extractor_model}/feature_vector/5"

```

```

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 Python 3
[1]: # Shuffle dataset
dataset = data['train'].shuffle(1000)

train_split = 0.8
val_split = 0.2
train_size = int(train_split * DATASET_SIZE)
val_size = int(val_split * DATASET_SIZE)

train_data = dataset.take(train_size)
train_data = train_data.map(preprocess_data)
train_data = train_data.batch(64)

validation_data = dataset.skip(train_size)
validation_data = validation_data.map(preprocess_data)
validation_data = validation_data.batch(64)

[2]: feature_extractor_model = "inception_v3"

[3]: tf_hub_uri = f"https://tfhub.dev/google/imagenet/{feature_extractor_model}/feature_vector/5"

[4]: feature_extractor_layer = hub.KerasLayer(
    tf_hub_uri,
    trainable=False)

[5]: model = tf.keras.Sequential([
    feature_extractor_layer,
    tf.keras.layers.Dense(units=NUM_CLASSES)
])

[6]: model.compile(
    optimizer=tf.keras.optimizers.Adam(),
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['acc'])

model.fit(train_data, validation_data=validation_data, epochs=20)

```

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

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 File Edit View Run Kernel Git Tabs Settings Help

Filter files by name

- / ... /ai-for-time-series/notebooks/
- Name Last Modified
- 01-explore... a minute ago
- 02-model.i... 5 minutes ago
- 03-cloud-t... 5 minutes ago
- cta\_riders... a minute ago

Terminal 1 01-explore.ipynb Python 3

fig = plt.show()

Autocorrelation

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.

```
[10]: 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 File Edit View Run Kernel Git Tabs Settings Help

Python 3 | Idle

Mode: Command ↵ Ln 1, Col 1 01-explore.ipynb

