Ch08

Machine Learning on GCP

CHAPTER ASSIGNMENT

Name _____ Class _____ No. _____ Date _____

> Answer every question in the space given. Show your working. You may use the chapter page to help you.

Questions from *Data Engineering with Google Cloud Platform*, 1st edition, Adi Wijaya (Packt, 2022), and from the Guru Jazzy chapter page, for classroom use.

Exercises from the book

Ex1 **Exercise ? practicing ML code using Python**

Create a machine learning model in Python (`scikit-learn` Random Forest Classifier) to predict `default_payment_next_month` on the Credit Card Default dataset (`bigquery-public-data.ml_datasets.credit_card_default`):

(1) Preparing the ML dataset: load the public dataset into `chapter_8.credit_card_default_train` (`WHERE RAND() < 0.8`) and `chapter_8.credit_card_default_test` (`WHERE id NOT IN (SELECT id FROM chapter_8.credit_card_default_train)`). Explain why the dataset is split into training and testing tables.

(2) Training the ML model using Random Forest in Python: run `ml_train_model.py` in Cloud Shell to load `limit_balance`, `education_level`, `age`, and `default_payment_next_month` from `credit_card_default_train`, fit `RandomForestClassifier(n_estimators=100)`, print the test accuracy (`metrics.accuracy_score`), and save `cc_default_rf_model.joblib`.

(3) Creating Batch prediction: run `ml_batch_predict.py` to load `cc_default_rf_model.joblib`, score `credit_card_default_test`, and write `prediction.csv`.

Ex2 **Exercise ? leveraging pre-built GCP models as a service**

Use the Google Cloud Vision API pre-built model to detect handwritten text from `handwriting_sample.jpg`:

(1) Uploading the image file to a GCS bucket: upload `chapter-8/data/handwriting_sample.jpg` to a regional `US-CENTRAL1` GCS bucket and record its `gs://` URI.

(2) Installing Python packages in the local environment: install `google-cloud-vision` in Cloud Shell and enable the Cloud Vision API.

(3) Creating a Python function to call the Vision API: run `vision_handwriting_detection.py` (`client.document_text_detection(image=image)`) on your GCS URI and record the detected text and confidence outputs.

Ex3 **Exercise ? using GCP in AutoML to train an ML model**

Walk through setting up a codeless AutoML tabular classification model in Vertex AI using `chapter_8.credit_card_default_train`:

(1) Create Vertex AI datasets: create a Tabular dataset in `us-central1` pointing to `chapter_8.credit_card_default_train` and generate column statistics.

(2) Set up AutoML training: choose Classification, AutoML, and `default_payment_next_month` as the Target column.

(3) Choose compute and pricing: explain why AutoML tabular training requires a minimum node-hour budget even on a small dataset, and why we stop before clicking **START TRAINING** in a practice project.

Ex4 **Exercise ? deploying a dummy workflow with Vertex AI Pipeline**

(1) Installing Python packages: install `kfp` and `google-cloud-aiplatform` in Cloud Shell and create a regional `us-central1` GCS bucket.

(2) Developing the Vertex AI pipeline in Python: inspect `ai-pipeline-dummy.py` (`step_one` and `step_two` decorated with `@component`, compiled with `compiler.Compiler().compile`, and submitted with `AIPlatformClient.create_run_from_job_spec`).

(3) Monitoring the pipeline in the GCP console: verify the run in Vertex AI Pipelines and record the contents of `output.txt` in your GCS bucket.

Ex5 **Exercise ? deploying a scikit-learn model pipeline with Vertex AI**

Deploy two separate Vertex AI Pipelines for the Credit Card Default dataset:

(1) Creating and running the first pipeline (`ai-pipeline-credit-default-train.py`): explain why `load_data_from_bigquery` exports `train.csv` to GCS and returns the GCS path string instead of passing an in-memory `pandas` DataFrame to `train_model`, and verify that `cc_default_rf_model.joblib` is saved in GCS.

(2) Creating and running the second pipeline (`ai-pipeline-credit-default-predict.py`): run the batch prediction pipeline to produce `prediction.csv` in GCS. Explain why training and prediction pipelines are kept separate, and what a green arrow icon on a Vertex AI Pipeline step means (`enable_caching=False`).

Sorting activities from the chapter page

S1 Match the term

ML is a data process whose output is a generalised formula for one objective, called a model. Your data skills apply directly. The new part is the vocabulary.

Options: **Feature** ? **Target** ? **Hyperparameter** ? **Batch prediction** ? **Online prediction**

No.	Scenario	Answer
1	Humidity and temperature: the columns used to train a rain predictor.	
2	Rain / No Rain: the value you want to predict for future data.	
3	The number of trees in a Random Forest, chosen by trial and error.	
4	Next week's Monday-to-Sunday rain forecast, computed in one go from this week's readings.	
5	An API takes two numbers and instantly returns "Rain" or "No Rain".	

S2 Which would you pick?

The approaches trade off three things: development time, expertise and cost. Ordered by least to most experimenting time: pre-built, AutoML, BigQuery ML, custom.

Options: **Pre-built model** ? **AutoML** ? **BigQuery ML** ? **Custom model**

No.	Scenario	Answer
1	Read handwritten text from an image with one Python API call and no training.	
2	A team with no data scientists needs a credit-default model from a tabular dataset this week.	
3	SQL-fluent analysts want to train a model without leaving BigQuery.	
4	An experienced team has time, needs a specific algorithm and wants to keep infrastructure spend low.	

S3 Training or prediction?

Vertex AI Pipelines is Kubeflow Pipelines, managed. Each step is a Python function wrapped as a container, and the pipeline definition is compiled and submitted to run on Google's infrastructure, not on your machine.

Options: **Training pipeline** ? **Prediction pipeline** ? **Both**

No.	Scenario	Answer
1	Fit a Random Forest on the training rows and write a versioned .joblib file to Cloud Storage.	
2	Load the saved model file and score today's unlabeled rows into prediction.csv.	
3	Query features from BigQuery and write them to a file in Cloud Storage for the next step.	

Quiz from the chapter page: Check yourself

1 In a production ML system, roughly how does the ML code compare with everything around it?

A. It is most of the work

B. It is a small part: data collection, verification, features, resources, serving and monitoring take far more effort

C. It is exactly half

D. It does not exist

---

2 Why do Vertex AI Pipeline steps pass Cloud Storage paths rather than a pandas DataFrame?

A. DataFrames are not allowed in Python

B. Every step runs in a different container on a different machine, so in-memory objects cannot be handed over

C. Cloud Storage is faster than memory

D. To avoid using BigQuery

---

3 Why keep training and prediction in separate pipelines?

A. GCP requires it

B. They can run at different frequencies, and the newest model is not always the best

C. To double the cost

D. Prediction cannot read models

---

4 You call predict on the test set inside your training script. What is that step for?

A. Batch prediction for the business

B. Online prediction

C. Calculating the model's accuracy on data it has not seen

D. Saving the model

---

5 Which approach is the fastest to start with, and what is its main limit?

A. Custom model; limited by budget

B. Pre-built model; limited to what Google offers and costly at high call volume

C. BigQuery ML; limited to Python

D. AutoML; limited to images

---

6 A pipeline step shows a green arrow instead of a tick. What does it mean?

A. The step failed

B. The step reused a cached result from an earlier identical run

C. The step was skipped by an error

D. The step is waiting for approval
