

Google Cloud Professional Machine Learning Engineer

Introduction

A Professional Machine Learning Engineer designs builds, and productions ML models to solve business challenges using Google Cloud technologies and knowledge of proven ML models and techniques. The ML Engineer considers responsible AI throughout the ML development process and collaborates closely with other job roles to ensure the long-term success of models. The ML Engineer should be proficient in all aspects of model architecture, data pipeline interaction, and metrics interpretation. The ML Engineer needs familiarity with foundational concepts of application development, infrastructure management, data engineering, and data governance. Through an understanding of training, retraining, deploying, scheduling, monitoring, and improving models, the ML Engineer designs and creates scalable solutions for optimal performance.

The Professional Machine Learning Engineer exam assesses your ability to:

- Frame ML problems
- Architect ML solutions
- Design data preparation and processing systems
- Develop ML models
- Automate & orchestrate ML pipelines
- Monitor, optimize, and maintain ML solutions

Course Objective

- Frame machine learning problems
- Design a machine learning solution architecture
- Prepare and process data
- Develop machine learning models
- Automate and orchestrate machine learning pipelines
- Monitor, optimize, and maintain machine learning solutions

Course Outline

Section 1: Framing ML problems

1.1 Translate business challenges into an ML use case. Considerations include:

- Defining business problems
- Identifying nonML solutions
- Defining output use
- Managing incorrect results
- Identifying data sources

1.2 Define ML problem. Considerations include:

- Defining problem type (classification, regression, clustering, etc.)
- Defining outcome of model predictions
- Defining the input (features) and predicted output format

1.3 Define business success criteria. Considerations include:

- Success metrics

- Key results
- Determination of when a model is deemed unsuccessful

1.4 Identify risks to feasibility and implementation of ML solution. Considerations include

- Assessing and communicating business impact
- Assessing ML solution readiness
- Assessing data readiness
- Aligning with Google AI principles and practices (e.g. different biases)

Section 2: Architecting ML solutions

2.1 Design reliable, scalable, highly available ML solutions. Considerations include:

- Optimizing data use and storage
- Data connections
- Automation of data preparation and model training/deployment
- SDLC best practices

2.2 Choose appropriate Google Cloud software components. Considerations include:

- A variety of component types - data collection; data management
- Exploration/analysis
- Feature engineering
- Logging/management
- Automation
- Monitoring
- Serving

2.3 Choose appropriate Google Cloud hardware components. Considerations include:

- Selection of quotas and compute/accelerators with components

2.4 Design architecture that complies with regulatory and security concerns. Considerations include:

- Building secure ML systems
- Privacy implications of data usage
- Identifying potential regulatory issues

Section 3: Designing data preparation and processing systems

3.1 Data ingestion. Considerations include:

- Ingestion of various file types (e.g. Csv, json, img, parquet or databases, Hadoop/Spark)
- Database migration
- Streaming data (e.g. from IoT devices)

3.2 Data exploration (EDA). Considerations include:

- Visualization
- Statistical fundamentals at scale
- Evaluation of data quality and feasibility

3.3 Design data pipelines. Considerations include:

- Batching and streaming data pipelines at scale
- Data privacy and compliance
- Monitoring/changing deployed pipelines

3.4 Build data pipelines. Considerations include:

- Data validation
- Handling missing data
- Handling outliers
- Managing large samples (TFRecords)
- Transformations (TensorFlow Transform)

3.5 Feature engineering. Considerations include:

- Data leakage and augmentation
- Encoding structured data types
- Feature selection
- Class imbalance
- Feature crosses

Section 4: Developing ML models

4.1 Build a model. Considerations include:

- Choice of framework and model
- Modeling techniques are given interpretability requirements
- Transfer learning
- Model generalization
- Overfitting

4.2 Train a model. Considerations include:

- Productionizing
- Training a model as a job in different environments
- Tracking metrics during training
- Retraining/redeployment evaluation

4.3 Test a model. Considerations include:

- Unit tests for model training and serving
- Model performance against baselines, simpler models, and across the time dimension
- Model explainability on Cloud AI Platform

4.4 Scale model training and serving. Considerations include:

- Distributed training
- Hardware accelerators
- Scalable model analysis (e.g. Cloud Storage output files, Dataflow, BigQuery, Google Data Studio)

Section 5: Automating and orchestrating ML pipelines

5.1 Design pipeline. Considerations include:

- Identification of components, parameters, triggers, and compute needs
- Orchestration framework
- Hybrid or multi-cloud strategies

5.2 Implement training pipeline. Considerations include:

- Decoupling components with Cloud Build
- Constructing and testing of parameterized pipeline definition in SDK
- Tuning compute performance
- Performing data validation
- Storing data and generated artifacts

5.3 Implement serving pipeline. Considerations include:

- Model binary options
- Google Cloud serving options
- Testing for target performance
- Setup of trigger and pipeline schedule

5.4 Track and audit metadata. Considerations include:

- Organization and tracking experiments and pipeline runs
- Hooking into model and dataset versioning
- Model/dataset lineage

5.5 Use CI/CD to test and deploy models. Considerations include:

- Hooking models into existing CI/CD deployment system
- A/B and canary testing

Section 6: Monitoring, optimizing, and maintaining ML solutions

6.1 Monitor ML solutions. Considerations include:

- Performance and business quality of ML model predictions
- Logging strategies
- Establishing continuous evaluation metrics

6.2 Troubleshoot ML solutions. Considerations include:

- Permission issues (IAM)
- Common training and serving errors (TensorFlow)
- ML system failure and biases

6.3 Tune performance of ML solutions for training & serving in production. Considerations include:

- Optimization and simplification of input pipeline for training
- Simplification techniques
- Identification of appropriate retraining policy

Prerequisites

- Basic understanding of cloud concepts
- Experience writing Python code

Recommended Experience

3+ years of industry experience including 1+ years designing & managing solutions using Google Cloud.

Target Audience

- Engineers who successfully wish to work on google workshop's powerful & efficient tools
- AI Engineers/ML Engineers
- AI Developers
- System/Software Engineers

Duration

3 days training course