**Fine Tuning Large Language Models for Enterprise Application Project Report**

**1. Student Information**

* **Student Name:** VAN HIEU NGUYEN
* **Student ID:** 221538422
* **Date Submitted:** 14 May 2025

**2. Project Introduction**

* **Title of the Project: Fine-Tuning Large Language Models for Enterprise Applications: Medical Misinformation Detection**
* **What is the project about?**

This project focuses on the fine-tuning of Large Language Models (LLMs) for domain-specific applications within the healthcare sector, with a specific emphasis on identifying and classifying medical misinformation. The goal is to improve the reliability and trustworthiness of AI-generated responses in medical contexts by customizing LLMs using datasets that include factual and misleading medical information.

* **Why is this project important or useful?**

With the widespread deployment of AI in healthcare from virtual assistants to clinical decision support systems—ensuring the factual correctness of generated responses is critical. Misinformation can lead to harmful outcomes. Fine-tuning LLMs using Supervised Fine-Tuning (SFT) and BERT with Low-Rank Adaptation (LoRA) allows for domain-specific alignment, improves classification accuracy, and helps mitigate hallucinations and bias in medical language processing.

**3. API/Token Setup**

**Objective:** Access and use an LLM through a secure API with Gemini 2.0 Flash model for training and inference.

**Instructions:**

1. Specify which provider you're using:
   * Hugging Face: BERT (Bidirectional Encoder Representations from Transformers) method for pre-trained model
   * Google Vertex AI: implement fine-tuning for Gemini 2.0 Flash Model and deploy an complete AI chatbots for using.
2. List the **steps you followed** to generate the token:

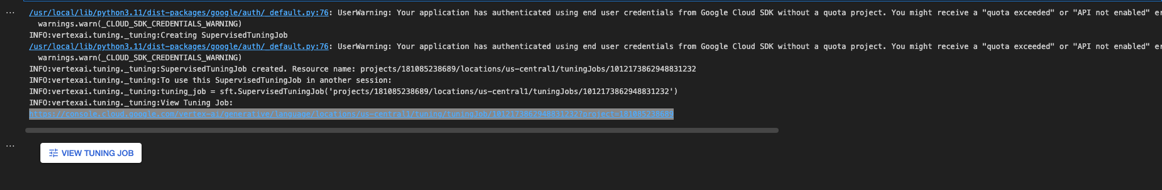
2.1. Pre-train model using BERT method

* + Step 1: Created account at https://huggingface.co/settings/tokens
  + Step 2: Navigated to the API/token section <https://wandb.ai/authorize?ref=models>
  + Step 3: Clicked on "Create new key”
  + Step 4: Copied the key and securely saved it: “7aecd1344b2f77be5dcc08d6ff6ecdb52886e5c9”
  + Step 5: View Run the project: <https://wandb.ai/hieunguyen23032001-deakin-university/huggingface/runs/k7ad0pp8>

2.2. Fine Tuning model using Gemini 2.0 Flash Experiment Model:

* + Step 1: Created account at <https://console.cloud.google.com/vertex-ai/studio/tuning?inv=1&invt=AbxQeA&project=sit319-25t1-nguyen-ae806d0>
  + Step 2: Navigated to “Tuning” and select on “Create Tuned Model”
  + Step 3: Create a model detail:
* Tuned Model name: “covid\_tuning”
* Based model: “gimini-2.0-flash-lite-001”
* Region: “us-central1”
  + Step 4: Create Tuning dataset:
* Upload train dataset to Google Cloud Storage: “gs://daft/Cleaned\_Covid19\_Train-7.jsonl”
* Model Validation: upload validation dataset to Google Cloud Storage (optional)
  + Step 5: Start Tuning with Gemini 2.0 model. <https://console.cloud.google.com/vertex-ai/generative/language/locations/us-central1/tuning/tuningJob/1012173862948831232?project=181085238689>

1. **Screenshot or terminal output (required):**



1. **Secure Loading of Token in Code:**

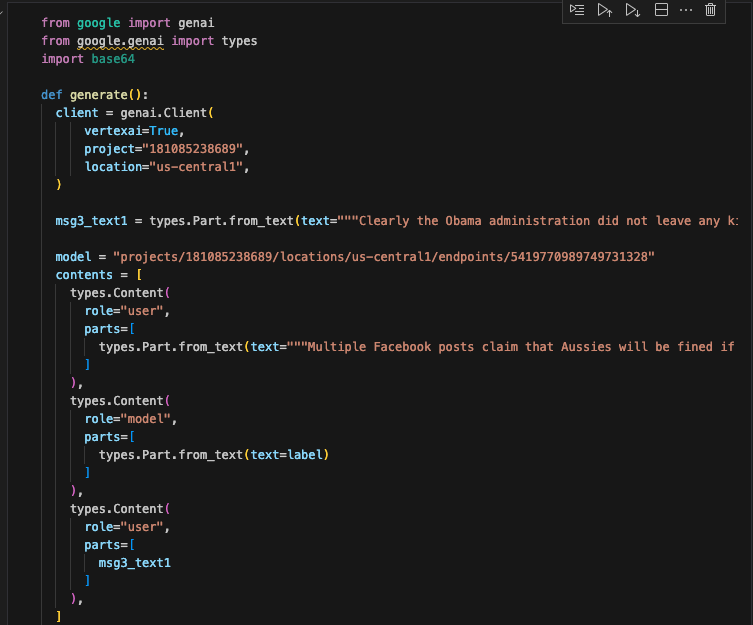
Using: vertextai.init() by importing:

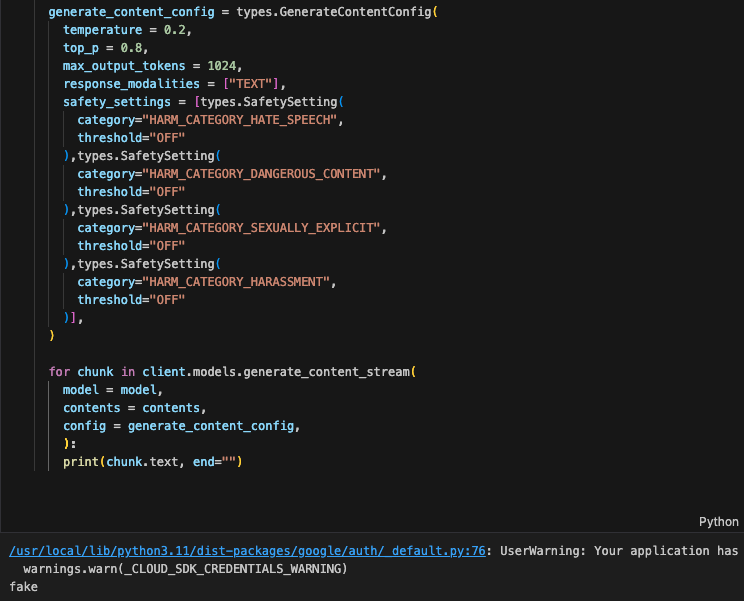
* from google.colab import auth as google\_auth google\_auth.authenticate\_user()
* import vertexai
* from vertexai.generative\_models import GenerativeModel
* from vertexai.preview.tuning import sft

Using genai.Client for testing prompt and get respond as label (True, False, or Misleading)

**Code: Load Token Securely**

****

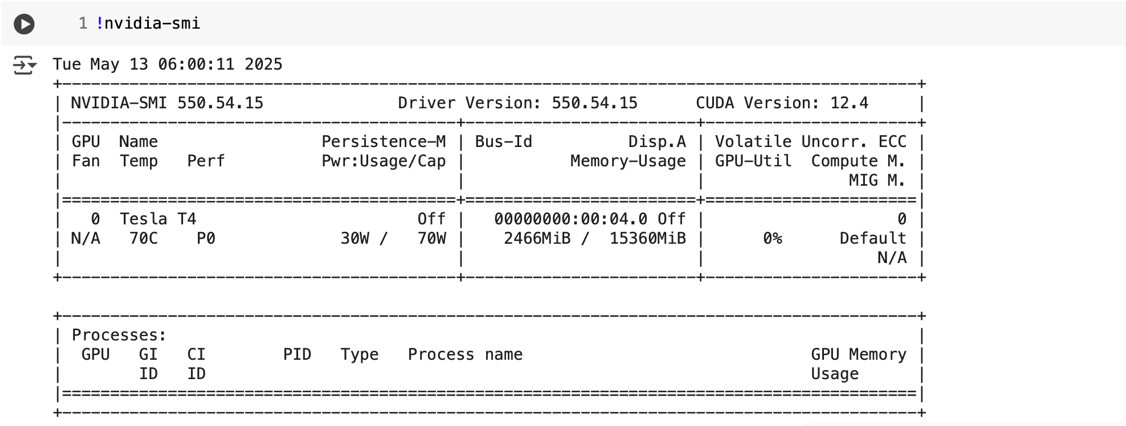
****

****

**4. Environment Setup**

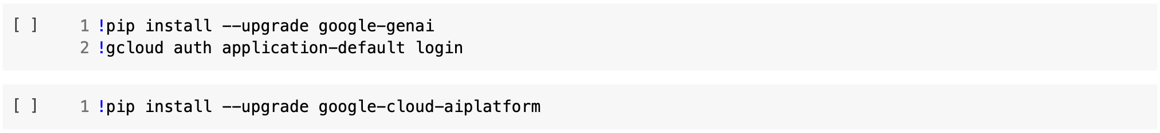
* **Development Platform:**
  + Google Colab
  + Local Machine (macOS)
  + GPU Available? [✓] Yes
  + GPU Type (if applicable): Local T4 GPU
* **Python Version:** Python 3.10
* **Other Tools Used:** VS Code, Google Colab, Google Vertex AI, Google Cloud.

**Code: Environment & GPU Check**

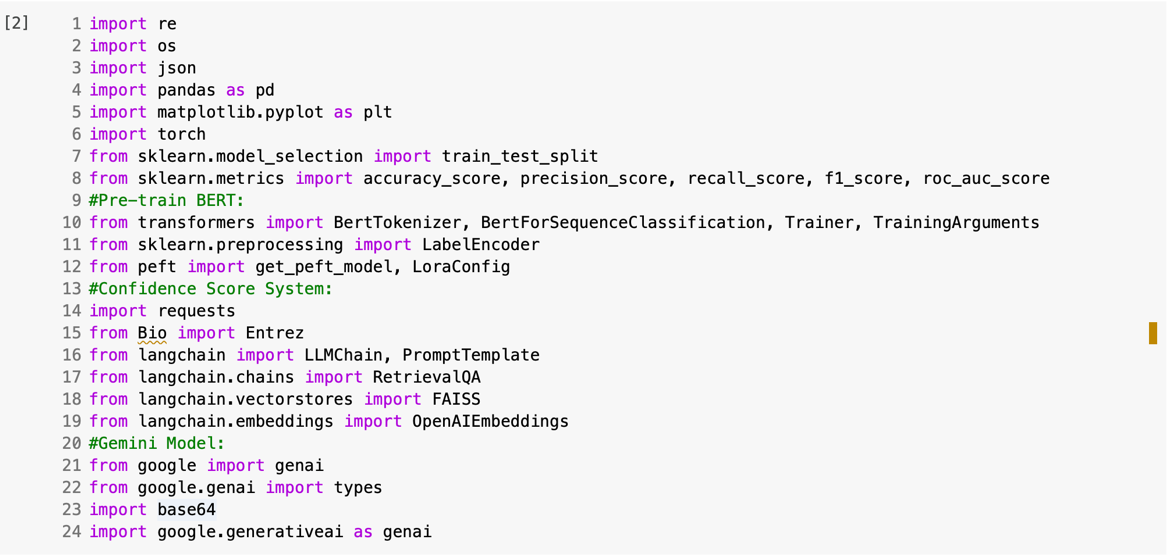
****

**5. LLM Setup**

* **Model Name:** Gemini 2.0 Flash (Experimental) and BERT method
* **Provider (OpenAI, Hugging Face):** Google Vertex AI (Gemini Model), Hugging Face (BERT)
* **Key Libraries & Dependencies (with versions)**
* **Libraries and Dependencies Required:**  
  ****



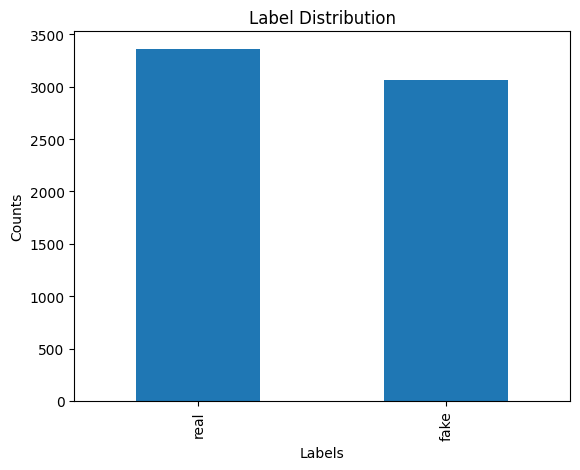




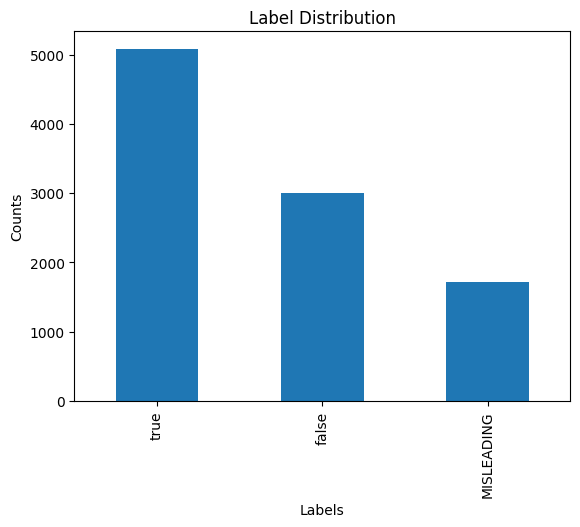
**6. Dataset Description**

* **Dataset Name & Source:** 
  + HealthFact: General misinformation
  + SciFact: Scientific claim fact-checking
  + COVID-19 Fake News: Pandemic-related misinformation
* **Access Link (if public):**
* Covid Fake New: <https://github.com/diptamath/covid_fake_news?tab=readme-ov-file>
* Health Fact (Medical Misinformation Detection) dataset: <https://github.com/neemakot/Health-Fact-Checking/blob/master/data/DATASHEET.md>
* Scifact dataset: <https://scifact.s3-us-west-2.amazonaws.com/release/latest/data.tar.gz>
* **Feature Dictionary / Variable Description:**
  + Claim: Textual medical claim
  + Label: true, false, and misleading
  + Evidence: Text used for verification (in two datasets)

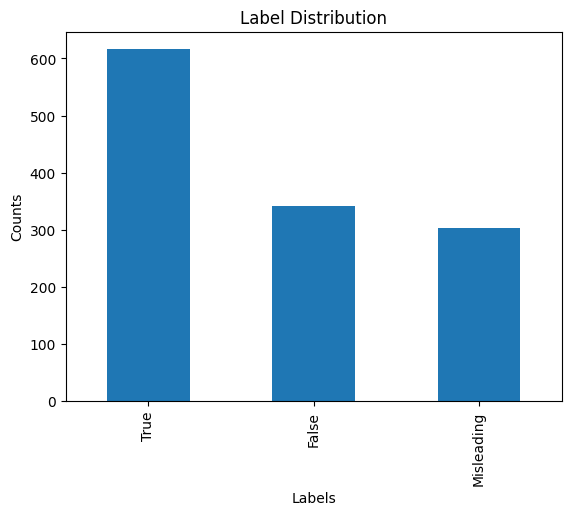
1. Covid19 Fake New dataset:



1. Healthfact Train Data:



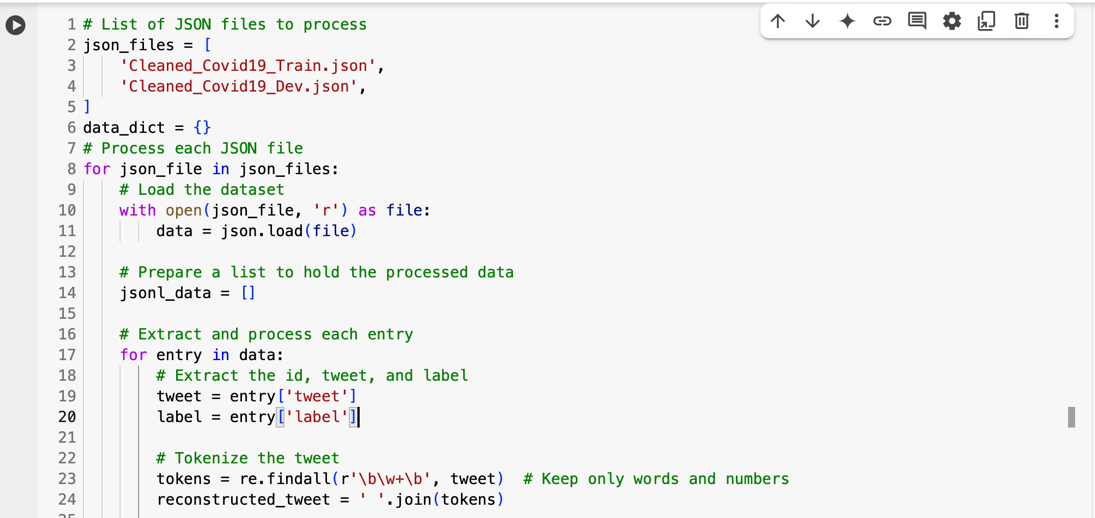
1. Scifact Train Data:



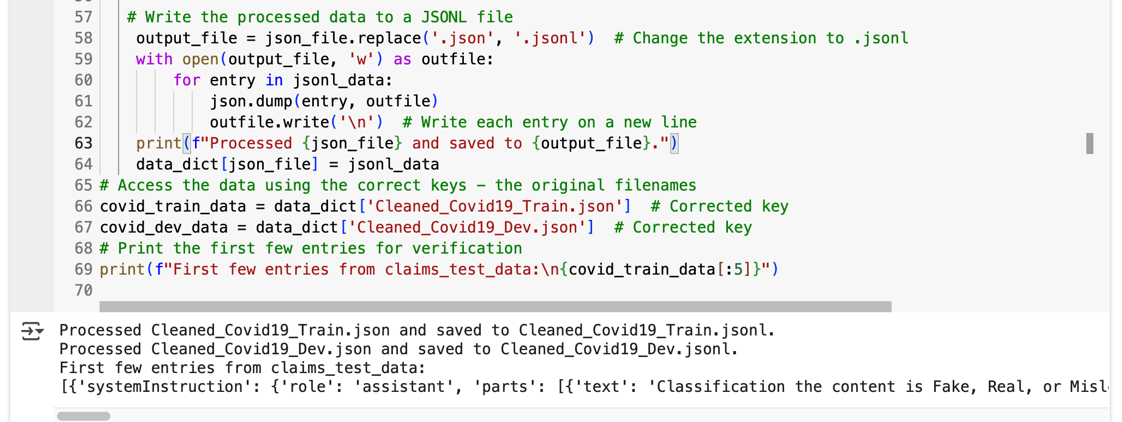
* **Was preprocessing done? If yes, describe:**
  + Pre-processing and cleaning for 3 datasets, and slits each dataset into three sub-set for train, test, and validate.
  + Standardized all datasets into 3-class classification: true, false, misleading.
  + Converted JSON into “.jsonl” for Vertex AI ingestion with the correct format.
  + Tokenized using Hugging Face tokenizer.
  + Balanced label distribution.

**Code: Load & Pre-process Dataset**

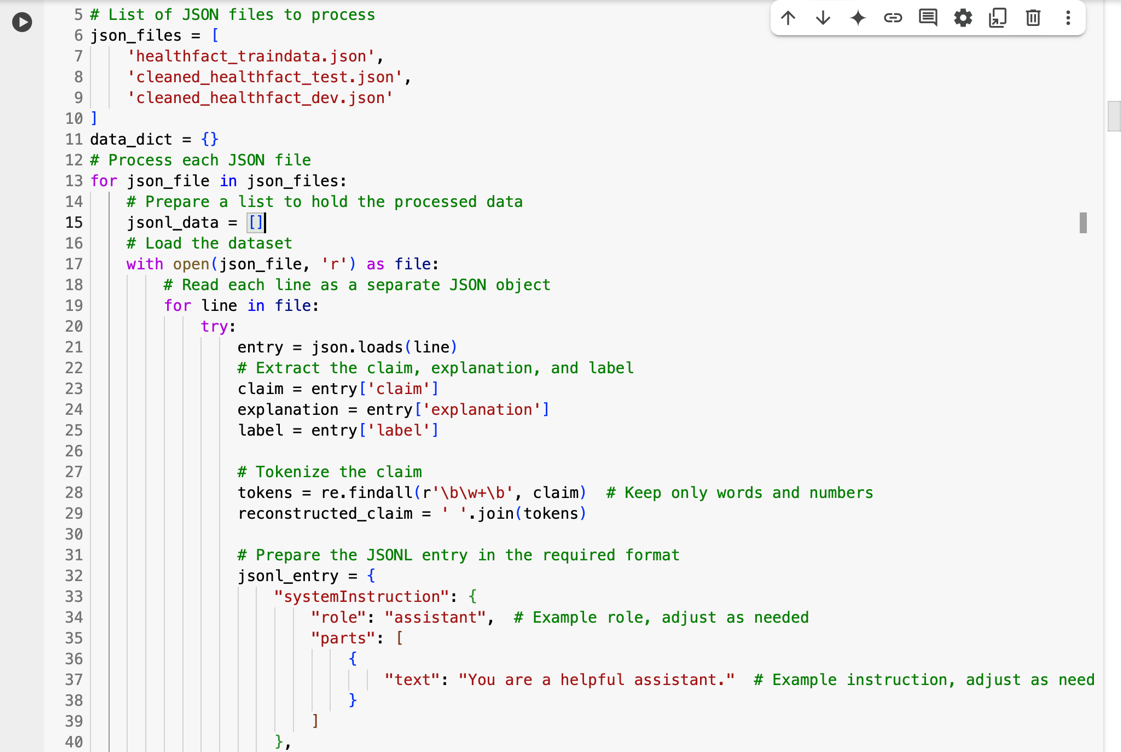
1. **Covid Fake News Dataset:**

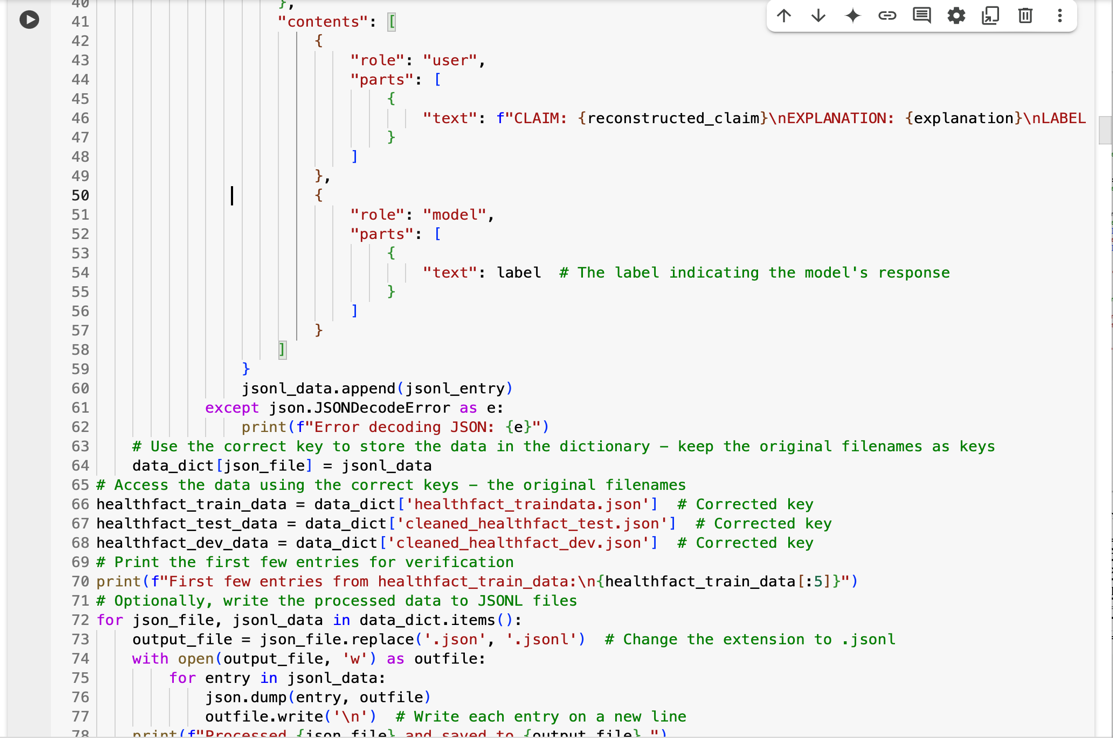
****

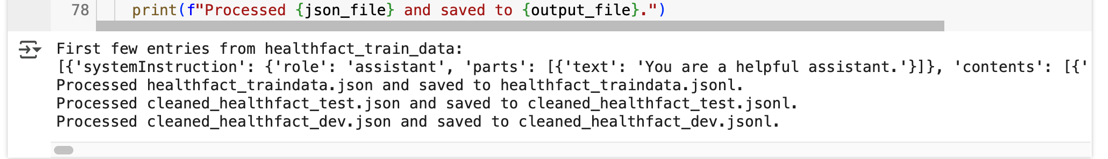
****

****

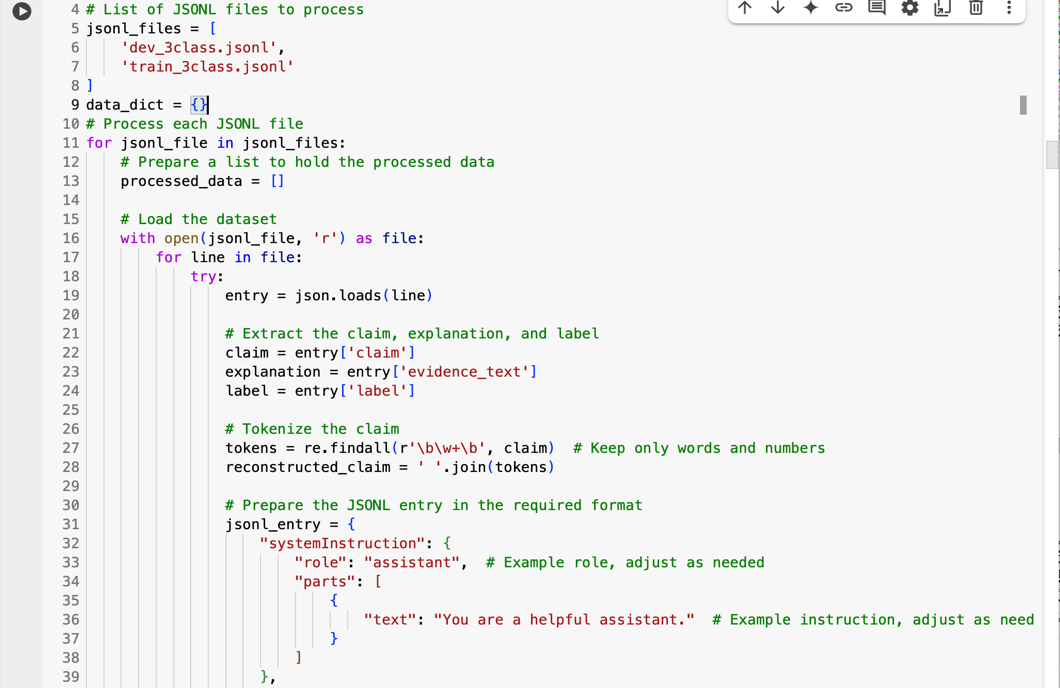
1. **Health Fact Dataset:**

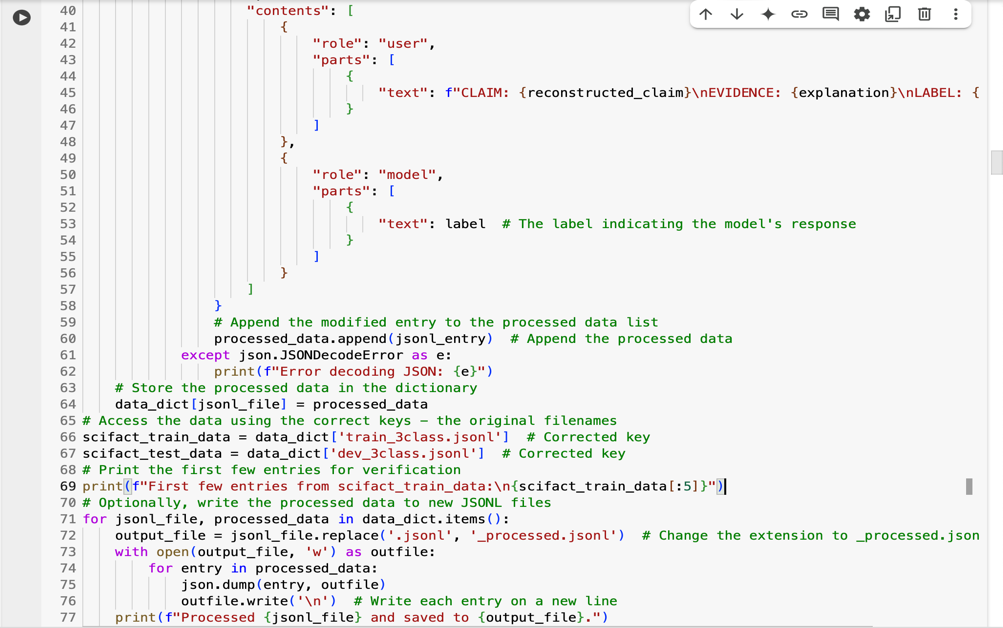
****

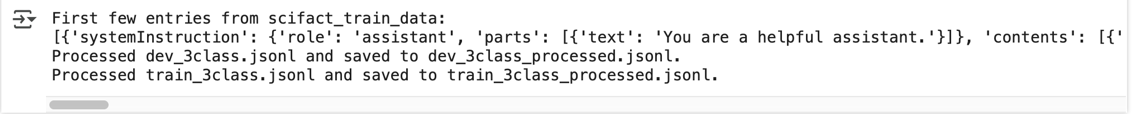
****

****

1. **Scifact Dataset:**

****

****

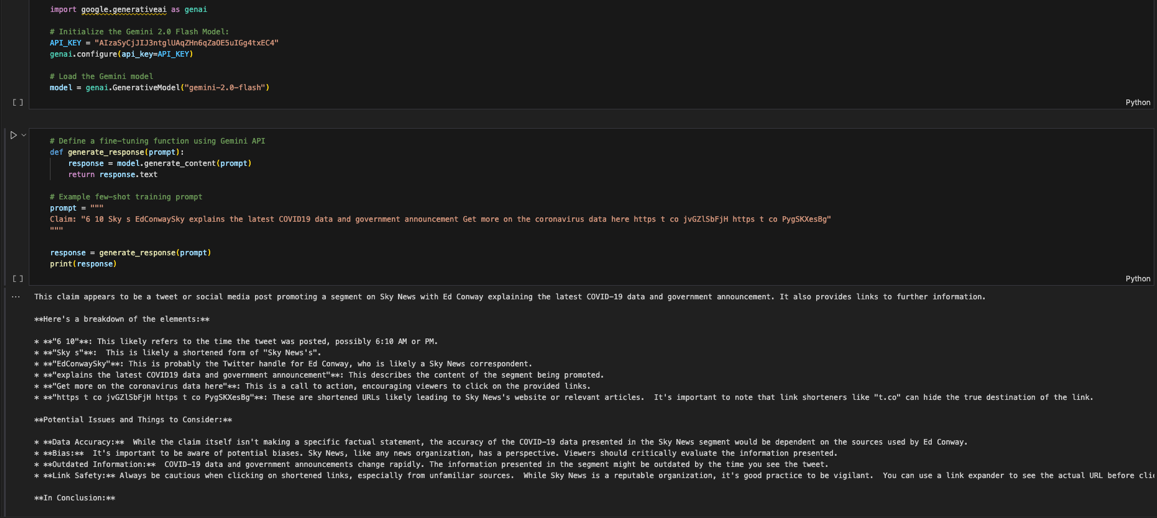
****

**7. Improving LLM Performance**

| **Step #** | **Method** | **Description** | **Result Metric (Accuracy)** |
| --- | --- | --- | --- |
| 1 | Zero-shot Prompt | No Training, Direct Response. | 58% |
| 2 | Few-shot Prompt | No Training, Testing 1 Prompt, and No label Response, Unclear Response | 58% |
| 3 | Temperature Tuning | 10 epochs on Vertext AI, Tuned Temperature from 58% to 80% | 80% |
| 4 | Fine-tuning | 50 epochs on Vertext AI, Testing prompt with 98-100% correct response with labels. | 98% |

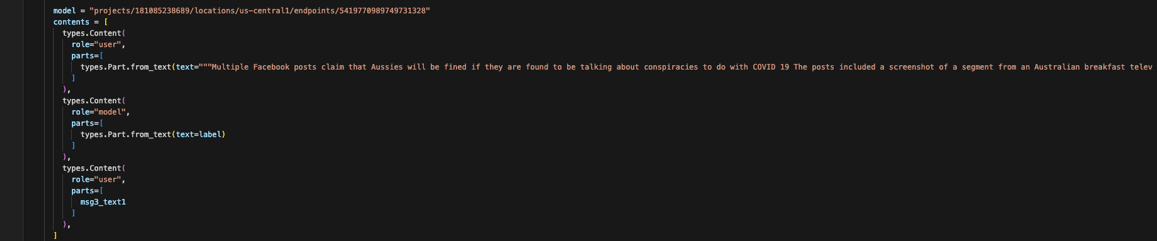
**Code Snippets for Each Step**

Before Training and Fine-tuning with few-shot training prompt:



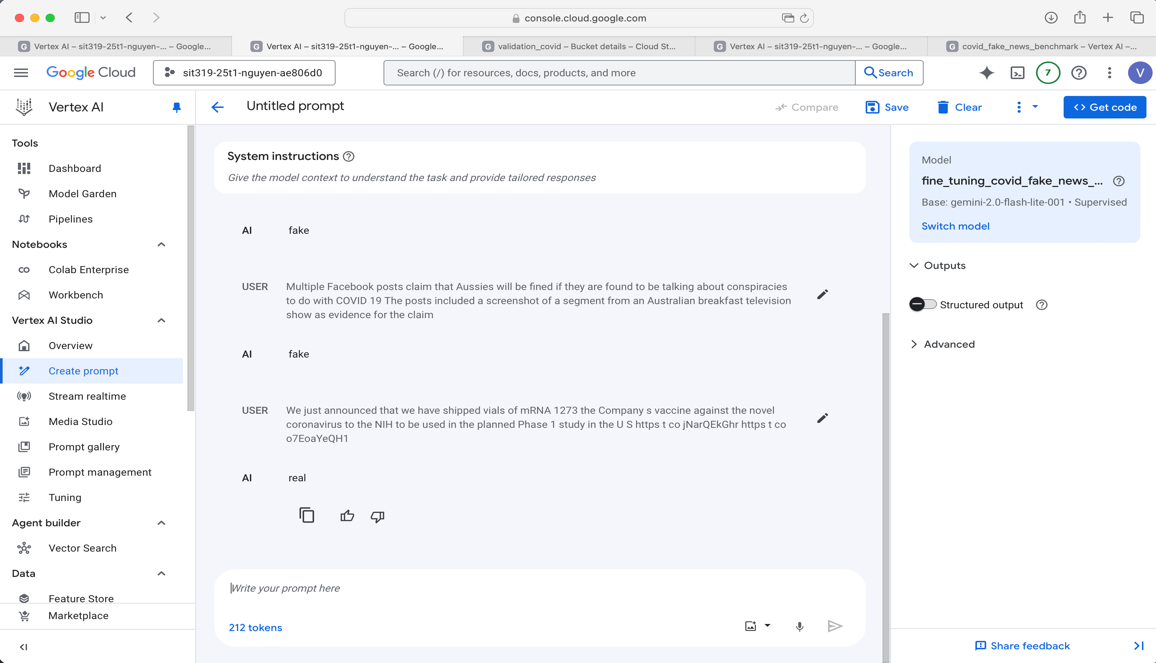
After Pre-train and Fine-tuning:

* + 1. One Shot Training Prompt





* + 1. Fine-Tuning google Vertext AI tool for prompt testing:



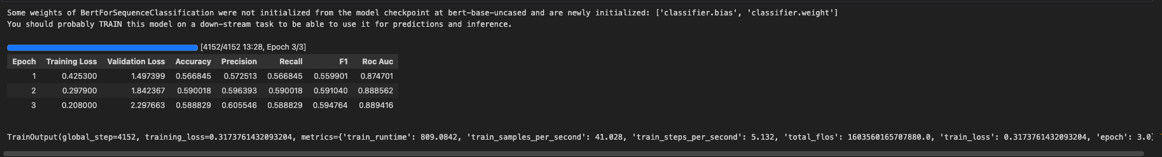
**8. Benchmarking & Evaluation**

**Required Components:**

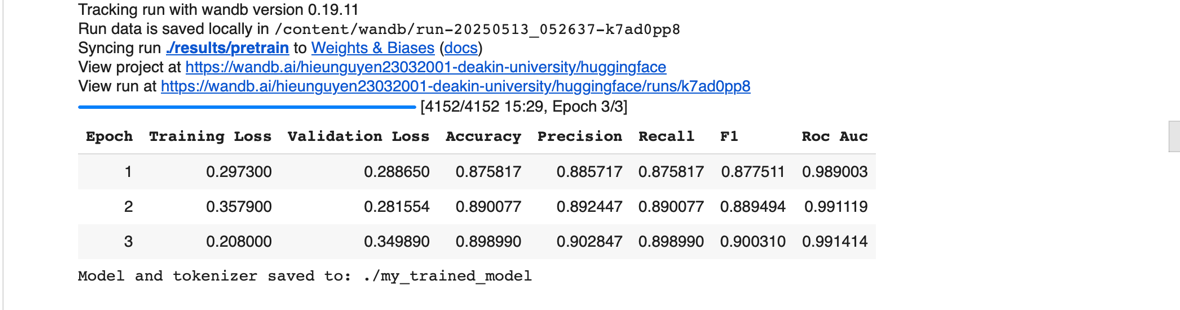
* **Metrics Used:**
  + Accuracy
  + Precision, Recall, F1 Score.
  + Confusion Matrix
* **Why those metrics?**

These metrics capture both correctness and type of error (especially important in medical misinformation detection). Moreover, using those metrics for comparing which evaluate as before and after applying fine-tuning model.

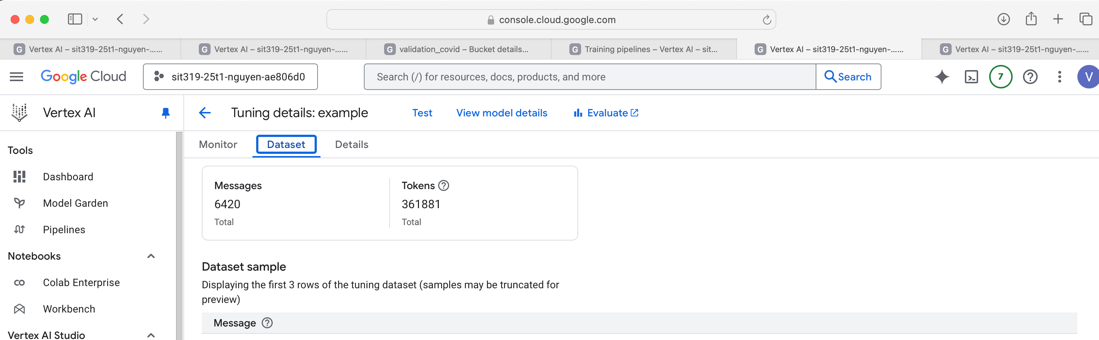
* **Benchmark Dataset & Sample Size:**
  + 1. Pre-train using BERT method for Covid Dataset:



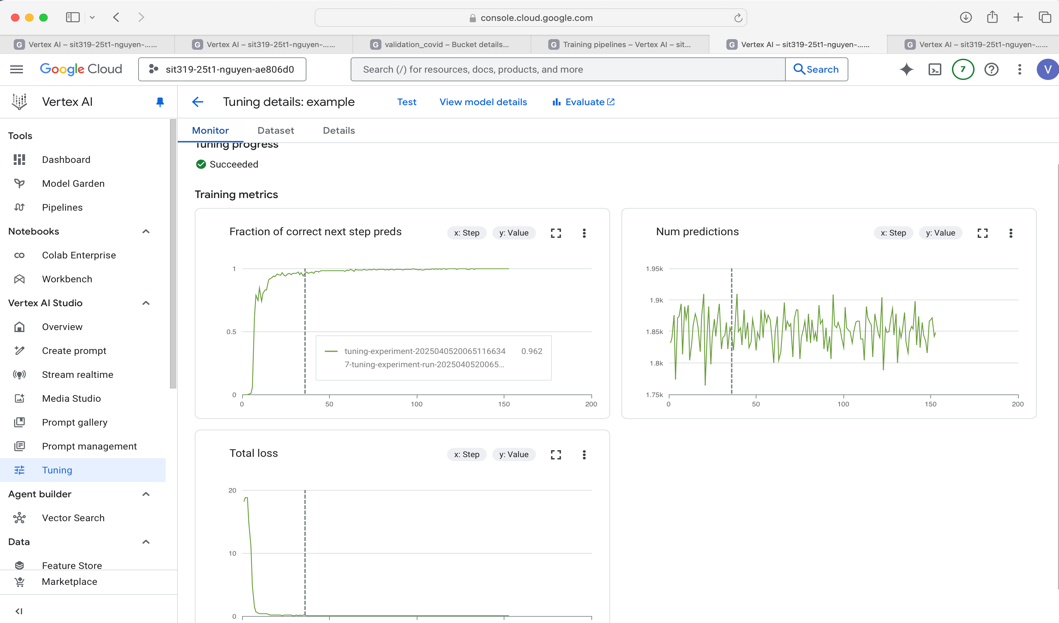
* + 1. Pre-train using BERT method for Combination Dataset:



* + 1. Fine Tuning for All dataset:
  + Sample Size dataset for fine-tuning:



* + Benchmark using Google Vertext AI:

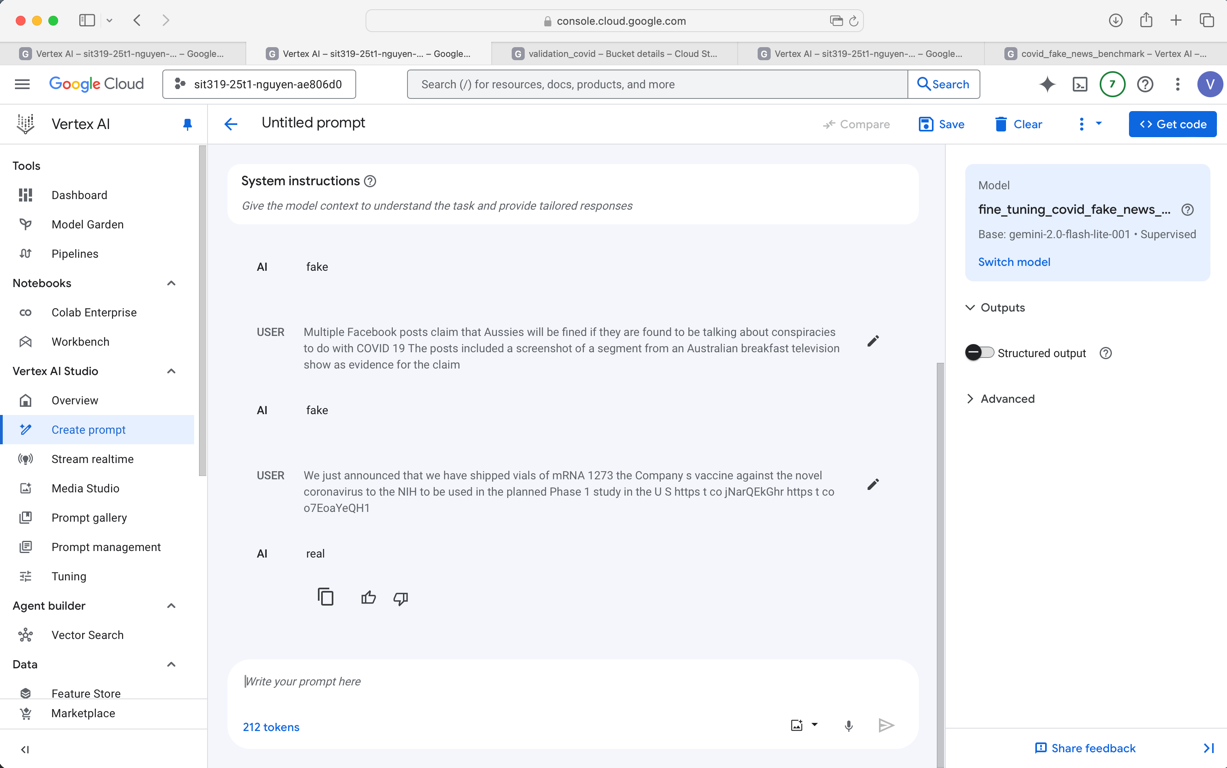


**Interpretation:**

The model was outstanding improve after fine tuning using Gemini 2.0 Flash model, it achieved almost 98 percent accuracy, compared with 54 to 58 percent using BERT method for pre-training model. This shows that fine-tuning using Gemini 2.0 Flash model makes the model more accuracy and more correct for the response.

**9. UI Integration**

* **Tool Used:** Google AI Cloud, Vertex AI
* **Key Features of the Interface:**
  + Accepts a medical question as input, and response with label.
  + Displays classification label (True, False, Misleading).
  + Displays confidence score.
* **Include Screenshots of Working UI:** using Google Cloud and Vertext AI for deployment chatbots in Vertex AI Studio



**10. References**

[1]. Parthasarathy B.V, Zafar A, Khan A, & Shahid A (August 2024), “The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities (Version 1.0)”, Arxiv, CeADAR Connect Group. <https://arxiv.org/html/2408.13296v1#Ch4.S1>

[2]. Huizenga E (13 December 2024), “Developer’s guide to getting started with Gemini 2.0 Flash on Vertex AI”, Medium.com, <https://medium.com/google-cloud/developers-guide-to-getting-started-with-gemini-2-0-flash-on-vertex-ai-6b4fe3c6899f>

[3]. Youtube (17 February 2024), “Fine Tuning a Model in Gemini and Vertext AI | Steps to make a LLM”. <https://www.youtube.com/watch?v=ej_ZUcyKpoc&t=218s>

[4]. Google Cloud (29 April 2025), “About Supervised Fine-Tuning for Gemini models”. <https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini-supervised-tuning>

[5]. Huizenga E (19 November 2024), “A Step-by-Step Guide to Fine-Tuning Gemini for Question Answering”. <https://medium.com/google-cloud/a-step-by-step-guide-to-fine-tuning-gemini-for-question-answering-8b3fb117dbbf>

[6]. Huizenga E (10 February 2025), “Fine-tuning Gemini: Best Practices for Data, Hyperparameters, and Evaluation”. <https://medium.com/google-cloud/fine-tuning-gemini-best-practices-for-data-hyperparameters-and-evaluation-65f7c7b6b15f>