Nebiyou Hailemariam

412-251-7954 | nebhailema@gmail.com | linkedin.com/in/nebhailemariam | nebhailemariam.github.io/

Education

Carnegie Mellon University — CGPA 3.93 *Aug. 2022 – May 2024*
Masters in Information Technology, Applied Machine Learning Specialization

Addis Ababa University — CGPA 3.8 *Sep. 2016 – May. 2020*
Bachelor of Science, Computer Science

Toptal — Top 3% of Freelance Engineers
Among the top 3% of engineers worldwide after Toptal's rigorous screening

Experience

Software Engineer II, AI *Mar. 2025 - Present;*
Motive *Remote*

- Building and maintaining a **multi-tenant car dealership platform** used by 400+ dealerships across the U.S. and Canada, serving millions of users, leveraging **Python (LangGraph, FastAPI, FastMCP)**, **Ruby on Rails**, and **Next.js**.
- Developed an **agentic conversational system** using **LangChain**, **LangGraph**, **LangSmith**, **FastAPI**, and **Pydantic**, capable of creating and updating webpages, performing content analytics, and generating SEO-optimized blog posts, enabling dealership admins to scale content automation.
- Built a **Vehicle Trade-in Evaluator** by training a deep feedforward network on 2 million vehicle inventory records in **PyTorch**, encoding vehicle attributes with **Sentence Transformers** (**all-MiniLM-L6-v2** embeddings), achieving **9.02% MAPE** on trade-in value prediction.
- Fine-tuned **lightweight reranker models** using **PyTorch**, **Sentence Transformers**, **Hugging face**, and **Vertex Workbench** to train a **content-based filtering recommendation system** to improve search result relevance and boost conversion rates.
- Designed and deployed **MCP** servers for dealerships to utilize our platform tools and integrate them with **Claude**, **ChatGPT**, and **Cursor** AI assistants.
- Set up **training and deployment pipelines** on **GCP Vertex AI** (Cloud Storage, Artifact Registry, Vertex Training, Vertex Model, Vertex Endpoint) and used **Weights & Biases** for experiment tracking and monitoring.
- Utilized the **pytest** framework to test AI microservice functionality, structuring tests with the **Arrange–Act–Assert** pattern for clarity and maintainability.

Software Engineer I, AI *Aug. 2024 - March 2025;*
eezly *Remote*

- Worked on **eezly**, a grocery price comparison application used by over 30,000+ users, leveraging **ASP.NET Core Web API**, **Python (FastAPI)**, **PyTorch**, and cloud-based microservices to build scalable, AI-driven app.
- Built a **Recipe Recommendation System** using **LangChain**, **OpenAI**, and the **Recipe1M+** dataset, creating a **Retrieval-Augmented Generation (RAG)** system to suggest recipes based on the products users purchase. Incorporated the **Weaviate** vector database to enhance search and recommendation.
- Employed **PyTorch** and **Hugging Face** to train hierarchical machine-learning models for classifying retail products from various stores (e.g., Walmart) into **aisles, categories, and subcategories**.
- Integrated **Gorse**, a recommender system, and contributed to open-source recommender systems.
- Designed and implemented **RESTful APIs** for inventory management using n-tier architecture and developed a **single-page application** with **React.js**.
- Implemented **OAuth 2.0 client-credential flow** using **OpenIddict** for secure machine-to-machine communication, **Single Sign-On (Firebase, Cognito)**, and **ASP.NET Core Identity** for user management.
- Designed a messaging system using **Kafka** with **Golang** as the message producer.
- Employed **xUnit** and **pytest** to write unit and integration tests for microservices using a custom web application factory.

Research Assistant in Machine Learning *May 2021 – Jan 2022*
Empathic Computing Lab *Auckland, New Zealand*

- **Empathic Computing Laboratory (ECL)** is an academic research laboratory directed by **Prof. Mark Billinghurst** at the **University of South Australia** in Adelaide, Australia, and the **University of Auckland** in Auckland, New Zealand.

- Collaborated with **Ph.D. students** to refine methods for detecting **emotions from physiological signals**.
- Conducted extensive **literature reviews** and analyzed the performance of various **machine learning** and **deep learning models**, applying rigorous Hyperparameter tuning. Authored a **14-page paper (IUI - ACM)**.

Projects — Full project list at nebhailemariam.github.io/projects

NebTorch — a minimal Autograd engine built from scratch using NumPy

- NebTorch is a minimal **Autograd engine** built from scratch using **NumPy**, inspired by PyTorch's automatic differentiation system.
- After completing and serving as a TA for **11-785: Introduction to Deep Learning** at CMU taught by Prof. Bhiksha Raj, I was inspired to build my own Autograd engine from scratch.
- Building NebTorch has been very rewarding—I've solidified my understanding of **Deep Learning** and **Automatic Differentiation**, and gained appreciation for frameworks such as **PyTorch** and **TensorFlow**.

DeepSeek-Mini — DeepSeek Model Components Implementation

- Implemented DeepSeek's major components in **PyTorch** for learning and experimentation: **Mixture of Experts (MOE)** with shared expert networks, **Multi-head Latent Attention (MLA)** for optimized inference with KV-caching, and **Rotary Positional Encoding (RoPE)**.
- Building these components from scratch solidified my understanding of MLA KV-caching and MOE architecture for reducing inference latency in transformer models.

Gorse Recommender System — github.com/gorse-io/gorse

- Contributed to **Gorse**, an open-source machine learning recommendation engine written in **Go**, enhancing the **.NET** library to make the Gorse recommender system more accessible to .NET developers.

XLM-R Based Extractive Amharic Question Answering with AmaSQuAD

- Completed my thesis project at Carnegie Mellon University in multilingual question-answering research (NLP).
- Developed a novel framework for translating the SQuAD 2.0 dataset into Amharic, resulting in the creation of an open-source dataset called AmaSQuAD.
- Implemented a translation-based data generation framework valuable for extractive Question Answering (QA) systems, contributing to the advancement of natural language processing (NLP) for low-resource languages.
- Leveraged XLM-R, a pre-trained language model, and fine-tuned it specifically for Amharic QA tasks, achieving 7% F1 improvements in baseline performance.
- The dataset is publicly available on Hugging Face.

Machine Learning Based Rain Gauge Using Acoustic Data

- Addressed the challenges of conventional weather stations such as high setup costs, instrument fragility, and measurement errors by exploring alternative sound sources for making rainfall predictions.
- Used a Convolutional Neural Network (CNN) regression model using PyTorch and TensorFlow to estimate rainfall intensities from MFCCs extracted from acoustic recordings. Employed CNN model and achieved a Mean Absolute Percentage Error (MAPE) of 35.20% and a Mean Squared Error (MSE) of 0.66, outperforming a baseline Support Vector Regression (SVR) model with 152.55% MAPE and 1.73 MSE.

Technical Skills

Languages: Python, C#, Go, JavaScript, Ruby
Frameworks: PyTorch, TensorFlow, Scikit-learn, Hugging Face, Pandas, NumPy, LangGraph, LangChain, FastMCP, FastAPI, Flask, Django, ASP.NET Core Web API, Entity Framework, OpenIddict, Gin, Express.js, Node.js
Dev Tools: Git, Postman, Grafana, Prometheus, Grafana-Loki, Jaeger
Web & Frontend Technologies: TypeScript, React, Next.js, CSS, Bootstrap, Redux
Cloud Services & Platforms: AWS (EC2, EKS, RDS, SES, SQS, MQTT), Google Cloud (Compute Engine, Cloud Storage, Colab), Azure, Firebase
Security: OAuth 2.0, JWT, ASP.NET Core Identity, Passport.js
Databases & Data Storage: Relational (MySQL, PostgreSQL), NoSQL (MongoDB, DynamoDB, Cosmos DB, Redis), Vector (Weaviate, Pinecone), ORM (SQLAlchemy)
Testing & Quality Assurance: Unit test, Integration test, xUnit, WebApplicationFactory, Bogus, pytest, Mocha, Jest
DevOps & CI/CD: Docker, Kubernetes, Jenkins, Containerization
Machine Learning & Deep Learning Skills: Linear Regression, Support Vector Machines, Bagging and Boosting, Neural Networks, CNNs, RNNs, LSTMs, NLP, Graph Neural Networks (GNNs), clustering, Graph Attention Networks (GATs), Transfer Learning, XGBoost, Hugging Face Transformers, Weights & Biases, Computer Vision
Data Science: Data Visualization (matplotlib), Data Analysis, NLTK
Architectures: Event-Driven, Microservices, Serverless, Monolithic Architecture, Test-Driven Development (TDD), Design patterns, Agile, REST API