Raj Pulapakura

Machine Learning Engineer

raj.pulapakura@gmail.com / 0469-766-823

github.com/raj-pulapakura | linkedin.com/in/rajpulapakura

www.rajpulapakura.com | medium.com/@raj.pulapakura | youtube.com/@rajpulapakura9119

Technical Skills

Machine Learning and Data Science	Python, TensorFlow, TensorFlow JS, TensorFlow Lite, Keras, PyTorch, Scikit-learn, OpenCV, Hugging Face, NumPy, Pandas, Matplotlib, Plotly, Seaborn, ChromaDB, Langchain, OpenAI API
Database	SQL, MySQL, PostgreSQL, MongoDB, AWS DynamoDB, Firebase Firestore, Firebase Realtime Database
Cloud and Infrastructure	AWS (ECR, EKS, EC2, S3, API Gateway, IAM, DynamoDB, Lambda, Athena, Glue), GCP (Vertex AI), BentoML, Terraform, Firebase, Git, GitHub, Docker, Kubernetes, Shell
Frontend + Backend Development	TypeScript, React.js, Next.js, Vercel, JavaScript, HTML, CSS, Tailwind, Material UI, Redux, Zustand, Node.js, REST APIs, Express.js, GraphQL, ApolloGraphQL, Redis, TypeORM
Mobile App Development	Flutter, Dart

Certifications

- **Database and SQL for Data Science with Python**, IBM — Dec 2023
- **Deep Learning Specialization**, DeepLearning.AI — Dec 2023
- **Advanced Machine Learning on Google Cloud**, Google Cloud — Nov 2023
- **IBM Professional Machine Learning Certificate**, IBM — Sep 2023
- **TensorFlow Developer Certificate**, TensorFlow — Aug 2023

Work Experience

A.I.GORITHM, Melbourne   Jan 2024 – Feb 2024

Software Engineer

- Developed and deployed a Property Report Generation Tool which enables farmers to get detailed agricultural reports pertaining to their geographical AOI (area of interest).
- Pooled and organized geospatial data from several APIs and data sources to provide agricultural analytics to farmers.
- Employed a microservices architecture to develop APIs/services and a user-facing website.
- Automated deployment via GitHub Actions CI/CD pipeline and Terraform IaC (Infrastructure as Code), to provision AWS resources.
- Software Technologies: *Next.JS, TypeScript, Tailwind CSS, Python, NumPy, Pandas, GeoPandas*
- Infrastructure: *Terraform, Docker, Microservices Architecture, AWS ECR, AWS ECS, AWS EC2, Fargate*
- Data Management: *AWS S3, AWS Athena, AWS Glue, AWS IAM, SendGrid*

Project Work

Real-Time Object Detection: Chrome Dino Game, [code](#), [video](#)

Project Overview: Developed a replica of the fun Chrome dinosaur game with Python, where the jump action is controlled through hand gestures captured through the webcam in real-time.

- Employed OpenCV and NumPy for real-time image processing
- Utilized my webcam and TensorFlow to collate a custom image dataset of hand images.
- Developed a real-time object detection model using TensorFlow, reaching precision of up to 97%.
- Improved inference time by 70% of baseline through iterative model optimization, while balancing detection accuracy and FPS through periodic detections, to ensure a smooth playing experience.
- Technologies: *Python, TensorFlow, TensorFlow Model Zoo, OpenCV, NumPy, Git, GitHub*

Computer Vision-Powered Search Engine, [code](#), [web](#), [video](#)

Project Overview: Engineered a full-stack website enabling users to drag and drop an image and find similar images using a Siamese neural network.

- Used a pre-trained MobileNetV3 neural network in PyTorch to compute image embeddings.
- Deployed the model through a REST API backend built with BentoML and hosted using automated Terraform infrastructure through AWS API Gateway and AWS Lambda.
- Developed a web interface using Next.js, TypeScript and Vercel, with drag-and-drop functionality and a beautiful grid layout.
- Technologies: *Python, PyTorch, REST API, BentoML, TerraForm, AWS API Gateway, AWS Lambda, Next.JS, TypeScript, Vercel, Tailwind CSS, Git, GitHub*

Temperature & Power Consumption Time Series Forecasting Model, [model](#)

Project Overview: Developed univariate and multivariate time series models for temperature forecasting.

- Achieved 30% better performance than baseline (metric was Mean Absolute Error) by employing advanced machine learning techniques including 1-dimensional CNNs, extensive hyperparameter tuning, Adam optimizer, and exponentially decaying learning rate.
- Conducted data manipulation, cleaning, preprocessing, train/validation split, analysis, and visualization.
- Technologies: *Python, TensorFlow, Scikit-learn, NumPy, Pandas, Matplotlib, Pyplot, Kaggle*

Natural Language Processing (NLP) Text Toxicity Checker, [code](#), [web](#), [model](#)

Project Overview: Developed a neural network and web interface for users to check their text for toxicity across 6 toxicity categories (toxic, severe toxic, obscene, threat, insult, identity hate).

- Employed TensorFlow to train a text classification model using LSTMs, on 60 MB of textual data.
- Quantized the model using TensorFlow Lite and TensorFlow JS for efficient edge-device inference and stored the model shards in a GitHub repo.
- Developed a web interface using Next.js (deployed with Vercel), TypeScript and Tailwind CSS, which enables users to enter their text and get a toxicity rating.
- Technologies: *Python, TensorFlow, TensorFlow Lite, TensorFlow JS, Kaggle, Next.js, TypeScript, Vercel, Tailwind CSS, Git, GitHub*

Other

- Contributed to the open-source packages TensorFlow and Scikit-learn on GitHub.
- Write a technical blog on [Medium](#), sharing content on machine learning, deep learning, computer vision, and software development.
- Post videos on my [YouTube](#) channel, demonstrating my projects and sharing knowledge on machine learning, data science, and web/mobile development.