Maryam F Syeda

maryamsyeda.work@gmail.com | (209) 831-2991 | Dublin, CA | https://www.linkedin.com/in/maryam-syeda-52ab956/

EXPERIENCE

Senior Machine Learning Engineer Feb 2021 - Present

Wells Fargo, San Francisco, CA

- Developed and deployed an AI assistant using a Retrieval-Augmented Generation (RAG) framework integrated with vector embeddings, fine-tuned LLMs, and scalable deployment on AWS Bedrock.
- Implemented RAG using HuggingFace Transformers and LangChain and stored them in ChromaDB and pgvector for efficient storage, retrieval, and similarity search across domain-specific documents.
- Developed an agent-chaining mechanism using AGENTIC AI for tasks such as text-to-SQL query generation, query validation, and execution, empowering the AI assistant to retrieve accurate, real-time answers from structured databases.
- Created a feedback loop to re-integrate database-derived information into the language model, refining responses with precise, contextual data.
- Fine-tuned an LLM with domain-specific datasets, optimizing query understanding, contextual accuracy, and response relevance for regulatory compliance and knowledge-driven responses.
- Configured AWS RDS with Similarity Search to set up PostgreSQL with pgvector to store embeddings and enable efficient similarity search, enhancing the AI assistant's ability to retrieve contextually relevant information.
- Used AWS Bedrock with Docker containers on Amazon EKS, building scalable cloud applications that interact with AI models through API calls.
- Evaluated the retrieval and response performance of the AI assistant by implementing precision, recall, and F1-score metrics for information retrieval, BLEU and ROUGE scores for response quality. Ensured continuous improvement through performance tracking and iterative fine-tuning.
- Implemented A/B testing to evaluate model performance, comparing different model versions and configurations to optimize results and make data-driven decisions for continuous improvement.
- Monitored performance using AWS CloudWatch for monitoring and logging to ensure optimal performance and detect unusual responses, enabling proactive debugging and model improvements.
- Utilized LangChain for advanced conversational capabilities and workflow automation, enhancing the AI assistant's ability to manage complex dialogues and tasks.
- Collaborated closely with data scientists and software engineers to align on model requirements, data pipelines, and deployment strategies, fostering a cohesive and productive development environment.

Machine Learning Engineer July 2019 - Jan 2021

CIGNA Health Insurance, Irvine, CA

- Developed, deployed, and maintained machine learning models using PyTorch for real-time applications, leveraging advanced deep learning techniques to drive business outcomes.
- Conducted extensive data collection using web scraping tools like BeautifulSoup and Scrapy and integrated APIs via Requests and RESTful services, storing the data in MySQL and MongoDB databases, as well as data lakes like Amazon S3.
- Automated the end-to-end ML pipeline using MLFlow, leading to a streamlined workflow that significantly reduced manual effort and improved model versioning and deployment.
- Developed and fine-tuned deep learning models with PyTorch, employing experiment tracking tools like MLflow and TensorBoard, and optimized model performance using hyperparameter tuning frameworks such as Optuna and Ray Tune.
- Evaluated model performance using Scikit-learn to calculate key metrics and applied cross-validation. Conducted rigorous hyperparameter tuning with Optuna and Ray Tune to optimize model robustness and generalization.
- Deployed models through AWS SageMaker Endpoints and served predictions through FastAPI, allowing users to interact with models in real-time via APIs for efficient, scalable inference.
- Utilized Amazon S3 for storing processed logs, training datasets, and model artifacts. Integrated AWS CloudWatch for real-time monitoring and logging of system performance metrics.

Data Engineer June 2017 - Jun 2019

Sallie Mae, Newark, DE

- Designed and implemented a real-time data pipeline using Apache Kafka to ingest high-frequency sensor data from IoT devices, ensuring scalable and low-latency message brokering for real-time processing.
- Developed stream processing workflows with Apache Flink/Spark Streaming to process and analyze sensor data in real-time, including anomaly detection and complex event processing.

- Integrated real-time data analytics by designing scalable, fault-tolerant systems, reducing time to insights and enabling real-time decision-making for stakeholders.
- Leveraged InfluxDB to store and efficiently retrieve time-series sensor data, supporting real-time querying and monitoring with enhanced performance.
- Set up monitoring and alerting systems within InfluxDB to track sensor metrics, trigger alerts, and ensure timely detection of anomalies based on predefined thresholds.
- Optimized data processing pipeline for high availability and low latency, leveraging horizontal scaling and data partitioning to handle large data volumes efficiently.
- Collaborated with cross-functional teams to integrate IoT sensor data with intelligence tools for proactive monitoring and continuous system improvements.
- Implemented end-to-end IoT data analysis solution, demonstrating expertise in data ingestion, processing, storage, and real-time visualization, while ensuring performance scalability and reliability.

Full Stack Developer Apr 2015 - May 2017

Whitebox-Learning, Dublin, CA

- Utilized Next.js to create a dynamic, responsive website with server-side rendering, ensuring fast load times and SEO optimization for improved user engagement.
- Built the backend using Node.js, implementing RESTful APIs to handle data retrieval and interactions between the frontend and SQL database.
- Designed and implemented an SQL database to manage and store company data securely, ensuring efficient data retrieval and scalability.
- Created an admin page for user management, allowing administrators to handle login access, user roles, and authentication securely via a custom-built dashboard.
- Deployed the application using Docker containers for easy environment management and scalability, orchestrated with Amazon EKS to ensure seamless, high-availability deployment across cloud infrastructure.

SKILLS

Languages	Python, JavaScript
Frameworks	TensorFlow, PyTorch, Keras, MLflow, Reactjs, Nodejs, Redux
Libraries	Langchain, Llamaindex, NLTK, HuggingFace Transformers, Scikit-learn, NumPy, Pandas
LLMs	LLaMA 2, Anthropic Claude, BERT, LLaMA 3.1, BART
Database	SQL, PostgreSQL, MongoDB
AWS	EKS, Bedrock, RDS, CloudWatch, Lambda, S3 bucket, EC2