Zaheeda Duduzile Chauke

Data Scientist / AI - ML Engineer

Pretoria and Cape Town, South Africa

081 371 2822
zaheedatshankie@gmail.com

Technical writing: https://medium.com/@zaheedachauke

GitHub: https://github.com/ZaheedaT

SKILLS & TECHNOLOGIES

Programming:
Python, SQL, PostgreSQL, Pandas, Scikit-learn, PySpark, C++, R, Git, Linux/Ubuntu, Conda, Jupyter

MLOps & ML Engineering: End-to-End ML Pipelines, Model Deployment & Serving, Experiment Tracking, Model Registry, Feature Stores, Model Monitoring, Data/Model Drift Detection, Automated Retraining, CI/CD for ML, ML Observability.
Tools: MLflow, Feast, BentoML, Evidently, Apache Airflow, GitHub Actions,Kubeflow, TFX, ML Metadata (MLMD), DagsHub, Hopsworks.

Cloud, Containers & Infrastructure: Docker, Kubernetes, YAML, Terraform, AWS (S3, SageMaker, EC2), GCP (Vertex AI, Compute Engine, Dialogflow), Azure OpenAI

Generative AI: LLMs, LangChain, RAG (Retrieval-Augmented Generation), GraphRAG, Prompt Engineering, Human Feedback Human-in-the-Loop (RLHF).

Machine Learning & AI: Regression, Classification, Feature Engineering, Model Evaluation, Data Mining, NLP, Sentiment Analysis, Computer Vision, CNNs, RNNs, LSTMs, Transformers, Deep Learning, TensorFlow, Keras, PyTorch

Data & Storage: SQL, PostgreSQL, MongoDB, Neo4j

Statistics: Probability Theory, Distributions and Hypothesis Testing, Model Evaluation, Bayesian Models, A/B Testing

EDUCATION

UNISA: BSc - Computer Science and Applied Mathematics (*2024)*
Explore Data Science Academy - *1 year Onsite* Data Science Certificate (2018-NQF5)

SHORT COURSES & CERTIFICATIONS

1. **MLOps Specialisation** - DeepLearning.AI (6 Certificates)
2. **Deep Learning Specialisation** - DeepLearning.AI (6 Certificates)

TECHNICAL PROJECTS/PORTFOLIO

Written in detail with Github Repo Links on page 4

1. End-to-End MLOps Pipeline: Python, MLflow, Feast, BentoML, Docker, Airflow, Evidently.
2. LLM RAG Chatbot for Hospitals | Python, LangChain, Neo4j, FastAPI, Streamlit, Docker
3. CI/CD Automation for Machine Learning | GitHub Actions, CML, Scikit-learn, Hugging Face
4. Llama 3 Fine-Tuning & Local Deployment | Python, PyTorch, Hugging Face, PEFT (LoRA), TRL, llama.cpp
5. **Technical Articles:**
    - “Build a Computer Vision WebApp”
    - “Intro to Docker for Data Scientists”
    - “How to set up a Data Science Project
    - “Boost your Topic-Modeling Performance with Coreference Resolution

EXPERIENCE

Outlier.ai

Code *Expert - Freelancer* *Oct 2024 - current*

- Training and fine-tuning various types of LLM models by generating effective prompts using Reinforcement Learning from Human Feedback Human-in-the-Loop (RLHF) techniques:
- Response Generation (Text-Based RLHF) ,Image-Based RLHF (Vision-Language Alignment), Image-Independent Reasoning, Test Reasoning (Task-Specific Reasoning), Safety and Ethical Reasoning
- Working independently on Code Refactoring by improving the structure and readability of code; Code Reviewing, Bug Fixing (Debugging) and Performance Optimization to make code run efficiently for C++ and Python scripts.

Zelo Analytics

Data Scientist & ML Engineer - Independent Consulting *Dec 2021 - Dec 2023*

I worked independently with a focused client base while completing my studies, designing and developing end-to-end analytics, machine learning and AI solutions. I built models and prototypes into deployable, production-oriented applications, including APIs, cloud infrastructure, MLOps pipelines and monitoring systems.

Selected ML & AI Engineering Solutions:

- Customer Analytics: Developed customer segmentation, churn propensity and product recommendation solutions to support targeted marketing, customer retention and revenue growth.
- LLM & GraphRAG Chatbot: Built and deployed an LLM-powered chatbot using LangChain, FastAPI, Streamlit, Docker and Neo4j, integrating a knowledge graph for contextual responses.
- Cloud & MLOps: Designed CI/CD and ML pipelines using GitHub Actions and MLflow; deployed models on AWS SageMaker and GCP Vertex AI, with infrastructure managed through Terraform and Kubernetes.
- Computer Vision: Developed a real-time object detection application using Flask and MongoDB, designed for live camera-stream processing and cloud deployment.

Omdena - AI for Good

Remote

Machine Learning Engineer - Community Collaborator *Dec 2021 - Nov 2022*

- Credit Scoring : Appointed as a Team Lead on a finance project to develop a credit scoring model. I had direct interaction with the Business Client to clarify the problem and present the solution to. Unsupervised modelling techniques were used to complete the task.

- Natural Language Processing (NLP) Environmental Conflict Tracker:

This project scrapes news media articles to identify environmental conflict events. I was appointed a Team Leader where we boosted the model's Accuracy by 10% using

- Appointed as a Task Manager for NLP Coreference Resolution which was performed on annotated data. We cross collaborated with and handed over data to the Topic Modelling team.

Direct communication was had with the Business Client.

The Gradient Boost ***Remote***

Data Science Mentor *25 July 2020 - Dec 2021*

Headhunted online to be a mentor for aspiring Data Scientists at beginner level.

- I taught Introduction to Python, SQL Fundamentals, Pandas, Descriptive Statistics, Data Visualisation, Introduction to Machine Learning and presentation/communication skills.
- Some big achievements while taking on this role are effective communication within groups, patience when guiding and teaching, interpersonal skills as well as empathy and understanding for those I mentor.

Telkom Insights ***Cape Town***

Data Scientist - Artificial Intelligence and Insights *(2 years) Jan 2018 - Dec 2019*

- Churn Propensity Model: Lead the development of a model which forecasts which customers or services are likely to churn. This is aimed at Customer Retention. The model and automated framework is used by multiple Business Units within Telkom to reduce revenue loss.
- Customer Segmentation: Part of a group that developed customer profiles using segmentation techniques. The project was concluded with a technical report and a Git Repository was handed over to the client for model Deployment.
- ChatBot Application: I focused on building aspects of the application backend and the frontend. I collaborated in building models on Google's DialogFlow.
- Speech-to-text Application: Collaborated to build an Automated Speech Recognition system for a ChatBot by running baseline models and gathering appropriate annotated voice data. (Models used SpeechRecognition, DeepSpeech, Wav2Vec 2.0 (Transformers) andTorchaudio together) HuggingFace-Speech2Text
- Telkom COVID-19 Fundamental Insights: I have written SQL scripts which aggregate important user movement data. I also supported drawing insights from relevant datasets and dashboards. The insights were then presented to the Telkom CEO and respective South African Provincial Ministers by our Insights Manager.
- Object Oriented Programming (OOP): Built multiple python modules and pipelines which were used in project Development and Staging processes before Deployment.

Team Research:

- Computer Vision as a weekly research topic. I took on a few Computer Vision and NLP text projects to upskill and enrich future team projects. Each week I presented my findings to our team, received feedback and improved the models by exploring techniques in this regard.

Explore - Software ***Cape Town***

Data Science Intern - One month internship *7 January 2019 - 31 January 2019*

- Assist in updating online software that will be used by 2019 students of Explore Data Science Academy
- Updating/fixing MS SQL Databases to be used by students
- Writing Deep Learning Jupyter Notebook and Google Colab tutorials for students to learn from.

Business Connexion (BCX Insights) ***Cape Town***

Data Science Intern *October 2018- December 2018*

- **Object Oriented Programming:** I dealt with automating the Data Science Process for BCX by developing Visualisation Modules using functions and classes in Python

- **Data Analysis:** Used Data Quality Reports (DQR) and Exploratory Data Analysis (EDA) to gain key insights from visualisations for Analysis. Findings and insights were presented to the supervisor and feedback was integrated to improve deliverables.
- **Team Management**: I was tasked with being the team manager to make sure deliverables were submitted on time and also assist the team in problem-solving.

Explore Data Science Academy

Cape Town

Learnership

21 *January 2018 - 6 October 2018*

- Completed an intensive Data Science learnership applying Python, SQL, AWS, machine learning and statistical analysis to real-world projects.
- Developed solutions involving regression, NLP, sentiment analysis, hypothesis testing, web scraping, APIs and data visualisation, presenting findings through dashboards and stakeholder presentations.
- Applied Agile methodology, delivering projects through iterative 2–3 week sprints.

AWS Cloud Computing was used for all projects*

TECHNICAL PROJECTS

---

1. **End-to-End MLOps Pipeline: Python, MLflow, Feast, BentoML, Docker, Airflow, Evidently.**

   (Repo: https://github.com/ZaheedaT/MLOps-Pipeline-OpenSource)

   Designed and implemented an end-to-end MLOps pipeline to automate the machine learning lifecycle from data ingestion and feature engineering through model training, registry, production serving, monitoring and retraining.

   - Built a feature management pipeline using Feast, enabling consistent feature storage and retrieval for model training and inference.
   - Implemented ML experiment tracking and model versioning with MLflow, enabling reproducibility and controlled model promotion.
   - Developed a production model serving API with BentoML and containerized the inference service using Docker.
   - Implemented data and model monitoring using Evidently, including detection of data drift, concept drift and model performance degradation.
   - Orchestrated ML workflows using Apache Airflow, including data processing, training, validation and monitoring workflows.
   - Implemented continuous integration and automated data/model validation, ensuring changes to source code and data are validated before deployment.
   - Developed a continuous monitoring and retraining feedback loop to trigger model retraining when production data/model behaviour deteriorates.

2. **LLM RAG Chatbot for Hospitals | Python, LangChain, Neo4j, FastAPI, Streamlit, Docker**

   (*Repo: https://github.com/ZaheedaT/LLM-RAG-Chatbot)*

   Developed an agentic RAG chatbot for querying synthetic hospital-system data across patients, visits, physicians, hospitals, payers and patient reviews.

   - Implemented a LangChain Agent to classify user intent and dynamically select the appropriate data retrieval tool, including Neo4j Cypher queries, vector search over patient reviews, and a hospital wait-time function.
   - Built a GraphRAG architecture using Neo4j AuraDB, leveraging graph relationships for structured healthcare queries and vector retrieval for unstructured patient-review data.
   - Developed and exposed the LLM agent as a REST API using FastAPI, with a Streamlit frontend for interactive user

access.
- Containerised the application using Docker and orchestrated the multi-service deployment with Docker Compose, enabling reproducible deployment across environments.

3. **CI/CD Automation for Machine Learning | GitHub Actions, CML, Scikit-learn, Hugging Face**

   (Repo: https://github.com/ZaheedaT/CI-CD_Pipelines)

   Built an automated CI/CD pipeline for an end-to-end machine learning workflow using GitHub Actions.

   - Automated model training, evaluation and reporting for a Scikit-learn Random Forest classification model using CML (Continuous Machine Learning).
   - Configured automated deployment of the ML web application to Hugging Face Hub, enabling reproducible delivery from code changes through deployment.

4. **Llama 3 Fine-Tuning & Local Deployment | Python, PyTorch, Hugging Face, PEFT (LoRA), TRL, llama.cpp**

   (Repo: https://github.com/ZaheedaT/Llama3-Finetuning)

   Fine-tuned Meta Llama 3 on a medical dialogue dataset to adapt the model for patient–doctor conversational interactions.

   - Implemented parameter-efficient fine-tuning (PEFT) using LoRA, reducing the computational requirements of adapting the base LLM.
   - Merged the trained LoRA adapter with the base model and published the resulting model artifacts to the Hugging Face Hub.
   - Converted the fine-tuned model to GGUF format and quantized the model for efficient local inference using llama.cpp-compatible tooling.
   - Deployed and tested the quantized model locally using Jan, enabling inference without relying on a hosted LLM API.
   - Structured the model for potential web application deployment, demonstrating an end-to-end workflow from model adaptation through inference.