Paul Hein

Software Engineer · Big Data Specialist

Redmond, WA | (+1) 520-289-9886 | paul.d.hein@gmail.com | github.com/pauldhein | linkedin.com/in/pauldhein

Education

University of Arizona

Aug. 2013 — June 2019

MS Computer Science — *Aug. 2017 — June 2019*
BS Computer Science / BA Mathematics — *Aug. 2013 — May 2017*

Thesis: Assembling Executable Scientific Models from Source Code and Free Text

Experience

Rocket Mortgage

Sept. 2021 — Aug. 2023

Senior Machine Learning Engineer — *June 2023 — Aug. 2023*

- Created a data quality monitoring system with **AWS Lambda**, **SQS**, and **CloudWatch** for detecting and reporting anomalies from a lead mining service.
- Automated ETL pipeline validation by creating a dataset synthesizer RESTful service using Synthetic Data Vault, **FastAPI**, **Docker**, and **Kubernetes**.
- Led a compute cost reduction of up to 95% for several pipelines by leveraging **Apache Spark** and **SQL** optimization to improve data pipeline efficiency.

Machine Learning Engineer — *Sept. 2021 — May 2023*

- Improved the throughput of a marketing attribution model using **Apache Spark**, **SQL**, and **AWS EMR** enabling daily processing of terabyte-scale data.
- Improved the technical maturity of a junior engineer through **mentorship** and **pair-programming** which lead to them receiving a promotion.
- Translated a proof-of-concept bayesian model for paid search optimization from **R** into **Python** using **Pandas**, **awswrangler**, **NumPy**, and **PyMC3**.
- Deployed a paid search optimization model to an **AWS SageMaker** endpoint capable auto-tuning Google Ads keyword bids with real-time inference.
- Created a development + deployment environment using **Bash**, **CircleCI**, and **Terraform** that reduced ML model rollout time from days to hours.

ML4AI Laboratory

June 2016 — Sept. 2021

Research Software Engineer — *June 2019 — Sept. 2021*

- Implemented a **Naïve Bayes** model, a **Bi-LSTM** network, and a deep **CNN** using **PyTorch** for classifying biological taxonomy from DNA sequences.
- Designed an **encoder-decoder model** for generating Python code from assembly code that led to the lab being awarded a DARPA research grant.
- Utilized the **PyTorch** DataParallel module and **Slurm** to achieve a 6x training acceleration for a sequence translation network on a GPU cluster.
- Provided graduate students with technical guidance on using scientific Python libraries with Docker to conduct **reproducible ML** experiments.

Graduate / Undregraduate Research Assistant — *June 2016 — June 2019*

- Implemented **feature selection** and **class imbalance** correction routines for a relation extraction model leading to a 45% improvement in precision.
- Designed a parallel hyperparameter grid search program using **MPI4Py** capable of tuning any **Scikit-learn** classifier on a distributed computing cluster.
- Created a corpora of musical patterns from jazz solos using a **spatial pattern discovery algorithm** for training an ML jazz solo generation model.
- Created a web application with **Python**, **Flask**, and **D3.js** capable of allowing an AI jazz generation model to record duets with a human musician.

Lunar Planetary Laboratory

April 2015 — June 2016

Student Programmer

- Assisted in developing a web application using **Node.js** that enabled scientists across the globe to view, create, and catalog spacecraft telemetry data.
- Assisted in designing a **database ERD** and implementing a **SQL schema** for pedigree tracking of data products originating from telemetry data.

Projects

BTD purchase predictor

July 2021 — Sept. 2021

- Successfully tuned an **SVM**, **random forest**, and a **neural network** classifier to determine if a bank client would purchase a bank term deposits (BTD).
- Created a training pipeline with **Python**, **Pandas**, **NumPy**, **Scikit-learn**, class imbalance correction, and grid search to achieve an 89% AUC-ROC score.

Source code summarization

Jan. 2019 — May 2019

- Developed an **encoder-decoder neural network** using **dyNet** and **NumPy** to generate natural language summaries for Python function source code.
- Created a corpora of python functions and docstrings from the Python package index using **NLTK**, **gensim** and regex for tokenization and encoding.

Skills

Engineering	Algorithm analysis • MLOps / DevOps • Object oriented programming • Test driven development • Agile software development
Data / ML	Supervised learning • Sequence modeling • Deep learning • Feature engineering • ETL • Data modeling • Data visualization
Languages	Python (expert) • Java (proficient) • R (familiar) • C++ (familiar) • JavaScript / Node.js (familiar) • Terraform • SQL • Bash
Packages	Apache Spark • Pandas • NumPy • SciPy • PyMC3 • Scikit-learn • PyTorch • NLTK • gensim • FastAPI • PyDantic • D3.js
Technologies	Git • Docker • Kubernetes • AWS • CircleCI • Slurm • Jira • GDB • Linux • Jupyter Notebooks • Microsoft Office suite
Soft skills	Technical leadership • Mentorship • Problem solving • Communication • Teamwork • Time management • Adaptability