Conner Rose

linkedin.com/in/ConnerRose • github.com/ConnerRose • conner.n.rose@gmail.com • (517) 648-1359

Education

University of Michigan, *Ann Arbor, MI* **Expected May 2026**

B.S.E. and M.S.E. in Computer Science, Completing Requirements for B.S. in Honors Mathematics *GPA: 3.80/4.0*

- **CS Coursework**: Object-Oriented Programming, Data Structures and Algorithms, Discrete Math, Machine Learning, Computer Organization, Algorithm and Computation Theory, Web Systems, Operating Systems, Formal Verification of Systems Software
- **Mathematics Coursework**: Calculus I-IV, Linear Algebra, Combinatorics and Graph Theory, Probability, Real Analysis, Graduate Probability Theory, Advanced Linear Algebra, Discrete State Stochastic Processes

Experience

Bloomberg L.P., *New York, NY* **June – August 2024**

Software Engineering Intern – Enterprise Data, Index Core Data

- Implement dependency resolution and conflict handling system in **Python**, resulting in reduced turnaround times when updating index calculation fields, reducing use of unnecessary dependencies, improving quality of calculation pipelines

Traders At Michigan, *Ann Arbor, MI* **September 2023 – Present**

Head of Software Engineering

- Lead development of ETF trading game, played at UMich Trading Competition by ∼100 competitors simultaneously
- Design and deliver advanced SWE curriculum to club members, supporting their SWE interview and career preparation

Bloomberg L.P., *New York, NY* **May – August 2023**

CTO Office Intern – Compute Architecture and OSPO

- Designed automated access revocation system using **Python** and **LDAP**, deployed to **Docker**-containerized **Jenkins Pipeline**, ensuring appropriate removal of inactive accounts from Bloomberg's open-source GitHub repositories
- Developed GitHub crawler using **Python** to scan all projects contributed to by Bloomberg employees over 10 years, automating contribution cataloging and open-source license compliance verification, increasing audited projects by **3x**

Projects

Zinger's (ETF Trading Game) **March – July 2024**

C++20, JavaScript, React, Websockets, Multithreading

- Architected ETF Trading Game, utilizing **websocket**-based infrastructure, enabling real-time data and order matching
- Developed **multithreaded** matching engine in **C++**, capable of handling **4,000,000 orders** in < **4 seconds** across 4 assets
- Implemented server using **μWebSockets** and **Glaze** for efficient JSON decoding in **C++**, capable of < **0.1ms 99% latency**

IMC Prosperity 2 (Global Trading Competition) **April 2024**

- Utilized ETF arbitrage, pairs trading, game theory simulations, and other strategies to place 53rd of 9,140 (top 0.58%)

Historical Landmark Image Classifier **October – November 2023**

Python, PyTorch, Pandas, NumPy, Matplotlib, Computer Vision

- Designed and implemented **convolutional neural networks** for multiclass image classification of historical landmarks
- Researched **model architecture** and **data augmentation**, employing subsampling and noise generation to improve accuracy and mitigate overfitting while training model with 5 convolutional layers and **+2,000,000** parameters

Movie Review Prediction System **September – October 2023**

Python, Scikit-learn, Pandas, NumPy, Gensim, Matplotlib, Natural Language Processing

- Trained **support vector machines** capable of classifying positive and negative movie reviews achieving **92%** accuracy through **sentiment analysis** techniques, including learned **word association** and **negation handling**
- Leveraged **word embedding association test**, identifying gender bias within dataset and resulting learned SVMs

MST/TSP Solution Generator, *C++* **April 2023**

- Utilized **arbitrary insertion** heuristic approach to generate approximate solutions for the **traveling salesperson problem** with quadratic time complexity, allowing for computation for **+10,000-order** complete graphs in seconds
- Developed **branch and bound** algorithm to guarantee optimal solutions to the traveling salesperson problem and optimized via **solution tree pruning**, using MST-derived upper bound, reducing runtime by **90%**

Technical Skills

Languages: Python, C++, Java, JavaScript/TypeScript, HTML/CSS, SQL (SQLite), LaTeX

Tools: Git, Docker, Jenkins, Django, Flask, React, Jupyter Notebook, MongoDB, Pandas, NumPy, Scikit-learn