

# Michael Capriotti

in [LinkedIn](#) [Github](#) [Portfolio](#) ✉ [michaelcapriotti2028@u.northwestern.edu](mailto:michaelcapriotti2028@u.northwestern.edu)

## Education

<b>Northwestern University</b> - Bachelor's degree in <b>Computer Science &amp; Mathematics</b> <ul style="list-style-type: none"><li>◦ <b>Coursework:</b> MENU Linear Algebra &amp; Multivariable Calculus, Data Structures &amp; Algorithms, Human Computer Interaction.</li></ul>	<i>Evanston, IL</i> <i>2024 - 2028</i>
--	---

## Work Experience

<b>Northwestern Computer Science Department</b> <b>Teaching Assistant</b> <ul style="list-style-type: none"><li>◦ Held office hours helping students learn computer systems concepts, including bit/byte representation, assembly, virtual memory, cache, file systems, and concurrency.</li></ul>	<i>Evanston, IL</i> <i>Jan 2026 – Present</i>
<b>Northwestern Residential Services</b> <b>Mail &amp; Package Center Assistant</b> <ul style="list-style-type: none"><li>◦ Processed and organized incoming packages, ensuring accurate tracking and timely pickup for students and staff.</li></ul>	<i>Evanston, IL</i> <i>Sept 2025 – Present</i>

## Research Projects

<b>Los Alamos National Laboratory</b> <b>Quantum Research</b> — <i>Python, QAOA, NumPy, Scipy, Matplotlib, Qiskit, CVXPY, Jupyter Notebook</i> <ul style="list-style-type: none"><li>◦ Applied the Quantum Approximate Optimization Algorithm (QAOA) to solve combinatorial optimization problems (portfolio optimization, travelling salesman, and maximum independent set), by transforming QUBO problems into Max-Cut instances.</li><li>◦ Utilized Semidefinite Programming (SDP)-based “warm-start” strategies, leveraging classical optimization to accelerate and improve the initialization of QAOA.</li></ul>	
<b>Northwestern Kellogg School of Management</b> <b>Financial Research</b> — <i>Python, Pytesseract OCR, LLMs, Openpyxl</i> <ul style="list-style-type: none"><li>◦ Developed scalable automated obituary classification by transitioning from manual labeling (500 samples) to OCR-based data extraction and automatic labeling using large language models (LLMs).</li></ul>	

## Software Projects

<b>Stock Forecast</b> — <i>FastAPI, React, TypeScript, Python, NumPy, Pandas, TensorFlow, Sqlite3</i> <ul style="list-style-type: none"><li>◦ Engineered a machine learning web application predicting next-day stock closing prices for 400+ companies using individual LSTM models trained on 5 years of historical market data, achieving an average mean absolute error (MAE) of 2.76%.</li><li>◦ Built an end-to-end system with a Python/FastAPI backend to fetch, clean, and process stock data via Yahooquery, train per-stock LSTM models, store results in Sqlite3, and serve them to a React dashboard with Chart.js visualizations for interactive analysis.</li></ul>	
<b>Golf Swing Analyzer</b> — <i>Flask, Random Forest, Python, MediaPipe, OpenCV, NumPy</i> <ul style="list-style-type: none"><li>◦ Created and deployed a full-stack web app enabling users to upload, trim, and analyze golf swing videos, classifying them as Pro or Amateur with interactive annotated pose overlays while minimizing backend memory usage.</li><li>◦ Collected and processed a dataset of 100 golf swing videos, extracting and normalizing 3D coordinates, velocities, and joint angles for 13 key body landmarks using MediaPipe Pose, and trained a Random Forest Classifier achieving around an 80% accuracy.</li></ul>	
<b>Trading Simulation</b> — <i>C++, Python, Matplotlib, CSV</i> <ul style="list-style-type: none"><li>◦ Developed a single-ticker market simulator in C++ modeling an order book with price-sorted matching and market dynamics, featuring five distinct trading bot strategies (Value Investor, Trend Follower, Stop Loss, Market Maker, Noise).</li><li>◦ Engineered bot logic to submit orders each tick with price, quantity, and time-to-live, tracked portfolios across 2,000 ticks, and computed real-time metrics for best, worst, and average performance per bot type, visualized with Matplotlib.</li></ul>	
<b>Evolutionary Algorithm Flappy Bird</b> — <i>Unity, C#, Evolutionary Algorithm</i> <ul style="list-style-type: none"><li>◦ Developed a physics-based Flappy Bird clone in Unity with multiple gameplay modes, including training, watching, and playing against an AI agent, with interactive UI controls and real-time visualization of the training process.</li><li>◦ Implemented a custom NEAT-inspired neural network in C# with elitism, mutation, and multi-cycle fitness evaluation, enabling progressive AI improvement across generations and automatic saving of best models for replay and gameplay.</li></ul>	

## Technical Skills

<b>Languages:</b> Python, JavaScript, C++, TypeScript, C#, SQL, HTML/CSS.
<b>Frameworks/Platforms:</b> React, Node.js, Express, Flask, FastAPI, Unity, MongoDB, Supabase, Docker, GitHub Actions, AWS.
<b>Libraries:</b> Pandas, NumPy, TensorFlow, Scikit-learn, Matplotlib, Qiskit, Mediapipe, Pytesseract.

## Additional Information

<b>Programs:</b> MIT Introduction to Engineering, and Science (MITES).
<b>Activities:</b> Develop & Innovate for Social Change, Consultants Advising Student Enterprises, Table Tennis Club.