Nicholas Belev

| nicholas.belev@gmail.com | (617) - 893 - 6652 | Boston, MA | GitHub | LinkedIn | Portfolio |

Analytical university senior in CS & Finance seeking full-time roles in finance and technology. Skilled in Python, SQL, data modeling, and applied AI.

Education

Bachelor of Science in Computer Science and AI | McGill University | *Montreal, QC* (Aug 2022 – Present)
▪ **Business Minor in Finance** ▪ CGPA: *3.94* ▪ Graduation: May, 2026

Key Courses: ▪ Data Structures & Algorithms ▪ AI / ML ▪ Data Science ▪ Statistics ▪ Software Design ▪ Robotics ▪ NLP ▪ Computer Vision ▪ OS ▪ Finance ▪ Blockchain & Cryptocurrency ▪ Financial Accounting

Certifications:
- Cybersecurity Foundations, *Google Cloud*
- Stock Market and Investment Research, *Bentley University*
- Data Visualization in JavaScript, *UMass Amherst*
- AI Fluency, *Adava University*

Technical Skills

Programming – Python (pandas, NumPy), Java, C / C++, JavaScript, OCaml, MIPS
Data & Analytics – Power BI, Jupyter Notebook, RStudio
Databases / SQL – PostgreSQL, MS SQL Server, SQLite
Cloud / DevOps – AWS EC2, Docker, FastAPI, Git, GitHub, Linux/Unix
Productivity – Excel (advanced formulas, VBA), PowerPoint, Word

Professional Experience

Production Software Developer Intern | **Northfield Information Services** | *Boston, MA* (May 2024 – Sep 2024)
- Automated migration of 16,000 legacy risk-model files into PostgreSQL with Python, delivers security data in < 100 ms.
- Benchmarked SQLite, PostgreSQL, and MS SQL Server on 1 TB of risk data query-retrieval; SQLite delivered 1.5x faster.
- Developed and Docker-deployed a FastAPI service with PyODBC, facilitating direct client data accessibility, for queries and responses; avg. response time of < 220 ms.

Quant Analyst - Developer Intern | **Northfield Information Services** | *Boston, MA* (May 2023 – Sep 2023)
- Implemented a "Householding" algorithm using Python and Northfield's Optimizer API to efficiently optimize the risk and return of conglomerate investment portfolio holdings while respecting individual portfolio preferences and risk tolerances.
- Utilized Python and SQL to identify time-series and cross-section trends in different dimensions of risk model data.
- Monitored shifts in security and risk factor volatility, identifying areas of concern for client portfolios.
- Designed and implemented data structure in credit rating estimation solution, enhancing reliability of risk analysis.

Treasurer | **McGill University Sailing Team** | *Montreal, QC* (Nov 2023 – Present)
- Manage $100K in assets, prepare annual budget, report financial status to McGill Athletics for audits and tax filings; oversee financial deliverables: balance sheet, cash flows, event-expense tracking; enables performance-to-cost analysis, enhances team's financial strategy.
- Forecast expenses with 96% accuracy by applying statistical methods, enabling long-term cost planning aligned with sustainability goals.
- Co-led fundraising initiatives; increased team assets by $20K in 1 year.

Highlighted Projects

Stock Evaluation Utility | Dec 2022 – Apr 2023
- Developed a Python application to evaluate potential stock over- or under- valuation and forecast future performance.
- Parsed financial statements and stock data from Yahoo Finance using Pandas, Requests, and Selenium Libraries.
- Calculated fundamental analysis ratios and derived a multivariable regression with autoregressive terms to capture momentum and reversals in stock behavior.

Gaze Tracking Software & Gaze Pattern Analysis (on GitHub) | Jan 2024 – May 2024
- Developed a real-time gaze-tracking system with OpenCV and MediaPipe to analyze visual focus on magazine covers.
- Mapped gaze data using perspective transforms; generated heatmaps and trajectory plots to visualize user attention patterns.
- Informs data-driven cover optimization (e.g. color scheme) through identified links between design choices and gaze behavior.

AI Player for Othello (on GitHub) | Oct 2024 – Dec 2024
- Designed an Othello game-playing AI, combining Alpha-Beta pruning, iterative deepening Minimax, and Zobrist hash state caching.
- Developed a game-state evaluation heuristic assessing stability, mobility, and position metrics, leveraging move-order techniques to calculate an optimal move under 2-second time and 500 MB memory constraints.
- Achieves 100% win rate vs. Greedy agents; 99% win rate vs. Stochastic agents; 80% win-rate vs. top peers' Minimax agents.