Arjun Aggarwal

arjunaggarwal173@gmail.com | linkedin.com/in/arjunaggarwal1/ | github.com/arjunaggarwal03 | arjunaggarwal.dev

Education

University of Maryland Expected: May 2025

B.S. Computer Science (Honors) & Applied Mathematics; Minor in Data Science *College Park, MD*

- **Honors**: Dean's List (all semesters), CS Departmental Honors Program, OMSE Academic Excellence Award (Spring 2022)
- **Relevant Courses**: Algorithms I & II, Data Structures I & II, Discrete Structures, Parallel Computing*, Database Design*, Intro to Compilers, Intro to ML, Computer Systems, Computational Methods, Linear Algebra, Intro to Probability Theory *(* are current)*

Experience

Amazon Web Services May 2024 – August 2024

Software Development Engineer Intern *Seattle, WA*

- Designed an auditing tool for AWS financial events in a ledger reporting system (handles **10M** monthly events), resolving the issue of untracked dropped events, ensuring **100% reporting completeness**, and **automating multi-hour** on-call response.
- Implemented **real-time** event tracking across services using **SNS-SQS messaging**, providing detailed monitoring.
- Wrote **Lambda** CRUD functions for new **DynamoDB** storing event status data, improving event visibility and reliability.
- Automated logging of incomplete events to **S3** using daily **EventBridge cron jobs** and configured **CloudWatch** alarms for proactive team alerts, preempting customer-reported issues and enhancing customer satisfaction.
- Streamlined deployment by configuring SNS, SQS, DynamoDB, and S3 attributes using **AWS CloudFormation** stacks (IaC).
- Built a **React** frontend application integrating a Lambda and the DynamoDB to display event statuses and incomplete events.

Bank of America June 2023 – August 2023

Software Engineering Intern *Jersey City, NJ*

- Completed 3 projects as part of a POC aiming to transition BofA batch risk testing to stream processing using **Apache Kafka**.
- Automated risk data testing with **Python** and **SQL**, replacing an older Alteryx workflow and reducing run time by roughly **85%**.
- Integrated Bitbucket API with workflow tools, reducing manual **30+ minute** data check-in time to **seconds** for **750+ analysts**.
- Designed a test info microservice using **Java** and **Spring Boot**, containerized with **Docker**, replacing inefficient legacy scripts.

Capital One January 2023 – April 2023

Machine Learning Engineering Intern *College Park, MD*

- Applied **Spark**'s optimized distributed querying to the Card transaction graph (**900M edges**), enabling faster node info retrieval.
- Utilized Spark GraphFrames and **motif queries (DSL)** for filtered node searches, leading to median **6x faster** graph querying.
- Conducted **80 cloud-based trials** with varying RAM/storage metrics to validate results; presented metrics to stakeholders.

Projects

CryptoArb Engine (link) | *Python, FastAPI, Kafka, Spark, Redis, Cassandra*

- Engineered a real-time cryptocurrency arbitrage detection system using **Apache Spark** Structured Streaming and **Kafka**, processing live price data from exchanges (Coinbase, Binance, and Kraken) to identify profitable trading opportunities.
- Designed a distributed system architecture using **Cassandra** for storage and **Redis** for caching, enabling sub-millisecond access.
- Exposed the database via **FastAPI** for trading metrics and historical data, with monitoring endpoints for system reliability.

Hermes | *Backend: Python, FastAPI, MongoDB, BERT, Pinecone, AWS EC2* | *Frontend: TypeScript, ReactJS*

- Designed a CLI tool allowing developers to message code snippets and communicate via the terminal, expediting development.
- Implemented user messaging via **FastAPI WebSockets** with **MongoDB** to store chat data; deployed API to **AWS EC2** instance.
- Stored **BERT embeddings** of messages in a **Pinecone vector database**, providing users with semantic search for chat logs.
- Utilized **$1K** award in credits from AWS Activate to host API and website built using **React** and **TypeScript**.

YOLOv3-based Vehicle Parking Pass Detector | *Backend: Python, Flask, YOLOv3, Google OCR* | *Frontend: HTML/CSS, JavaScript*

- Achieved **96% accuracy** in detecting vehicle parking passes with custom-labeled training data via **YOLO** real-time detection.
- Integrated **Google Cloud AI** optical character recognition to detect pass identification numbers, automating record-keeping.

Skills

Languages: Python, Java, JavaScript/TypeScript, C/C++, Ruby, OCaml, SQL, HTML/CSS
Other: Git, Django, Flask, FastAPI, Linux/Unix, AWS Tools (S3, EC2, DynamoDB, Lambda, SNS/SQS, EventBridge, CloudFormation), Apache Spark, MongoDB, Pinecone