Arjun Rao

mailarjunrao@gmail.com | arjun.fyi | arjun-rao | arjunra0 | Google Scholar

Summary

Software engineer specializing in building systems at the intersection of applied ML, research and AI infrastructure with industry experience working on large-scale recommender systems at Netflix & Microsoft.

Industry Experience

Netflix *Los Gatos, CA, US*

Software Engineer, Machine Learning, Core Recommendations *March. 2024 - Present*

Building & optimizing end-to-end ML training infrastructure for core ranking & page construction to personalize the Netflix Homepage. As an engineering partner for applied researchers, I collaborate with downstream serving and platform teams to enable new capabilities, improve experimentation agility and optimize resource utilization. I'm also involved in the hiring process, conducting system design and technical interviews for various roles.

- Tech lead for re-designing our existing training infrastructure for our ranking models to a new managed ML experimentation platform that aims to improve research productivity and enable new ML capabilities like Multi-Task Learning, larger scale models, and optimize compute utilization. I contribute to cross-cutting architecture design and own our system integration effort. The platform aims to abstract various ML Lifecycle activities into re-usable components that can be dynamically chained to produce ML artifacts using a custom configuration layer.
- Integrating a flexible dynamic catalog access system with our training infrastructure to enable access to accurate, point-in-time catalog availability at model training time. This improves our personalization capabilities to support new content types and subscription tiers.
- Redesigned our real-time signal events pipeline to a flexible framework that allows schema changes independent of binary deployments, mitigating the need to backfill data, leading to faster turn-around time for new user signal adoption from weeks to days.
- Technologies Used: Java, Scala, Python, Tensorflow, Pytorch, S3, Iceberg, Spinnaker, Bazel, Monorepo

Microsoft *Redmond, WA, US*

Software Engineer II, Feeds Ranking Infrastructure *June. 2021 - Feb 2024*

Tech lead for the online serving infrastructure for Microsoft's personalized feed product (MSN/Microsoft Start) surfaced across various canvases like Edge, Windows, Xbox and Skype (500M+ MAU). My role involved designing and leading various infrastructure initiatives that significantly improved system performance, capability, user engagement, and researcher productivity across the ranking stack. I also led various initiatives to improve org culture, documentation discovery, conducted several interviews, on-boarded team members, and was an intern mentor.

- Primarily owned the serving stack for a real-time feature store used for serving aggregated user & item features for ranking models. The service was implemented using an internal implementation of Flink, a high-throughput key-value store, and custom serving infrastructure. I was responsible for designing support for various aggregation capabilities including second-level sliding-windows, life-long time-decayed aggregation and long-term batch aggregation. The serving system had an end-to-end avg. latency of less than 100ms (p95) and served an average 2M+ QPS from traffic across geographies. I was responsible for the end-to-end serving architecture & implementation, aimed at low-latency serving of fresh features, optimizing resource usage and supporting rapid experimentation agility. The system served 50% of the most important features in the L2 Ranker, and contributed to 11% incremental revenue gains, aggregated 15% gain in engagement metrics. My direct impact included an aggregated 50% reduction in CPU & memory footprint over several projects across 2 years.
- Led a project to create dynamic configuration for the feature store to allow self-servable feature configuration and reduce time from idea to feature from days to hours.
- Served as a tech lead for the debuggability workstream for FY24, working with a cross-functional team of engineers distributed across Egypt, China and Redmond. Worked on road-mapping and strategies for reducing the time to debug issues across the recommendation stack.
- Experimented with using GPT4 for User Interest Summarization and LLM based Feed Ranking as an internal prototype.
- Technologies Used: C++, C#, .Net, Python, Pytorch

Education

M.S. in Computer Science, University of Colorado, Boulder, USA Aug. 2019 - May. 2021

B.E. in Information Science, Ramaiah Institute of Technology, Bangalore, India Aug. 2014 - Jun. 2018

Industry Experience (Contd..)

Stride.ai Inc. *Bangalore, India*

NLP Engineer *Jul'18 - May '19*

Stride.AI is an AI startup that builds solutions for document intensive process automation. I was one of the early full-time engineers at the startup.

- Designed and built re-usable end-to-end components to build custom Intelligent Process Automation solutions that leverage AI models to support document intensive processes for knowledge extraction & compliance across industry verticals like financial regulation, banking and investments. Reduced the turn around time to build a new Proof-of-Concept from over 1 month to 1 week. The components I built included all parts of the stack from UI, Backend, Storage, Model Training, Inference, Feature Parsing and Serving.
- Built a custom PDF Viewer using PDF.js that allowed automatic ML based annotations and additional capabilities like support for multiple monitors, side-by-side scrolling of multiple documents, cross-browser support, lazy-loading documents. The viewer is was used in 10+ Projects and reduced latency of loading 100 page pdfs from 5+ seconds to < 1 second.
- Technologies Used: Python, Django, Keras, Tensorflow, D3.js, Angular, Typescript, PDF.js, Internal tools

Academic Research Experience

University of Colorado, Boulder *Boulder, US*

Emotive Computing Lab, Institute of Cognitive Science *Jan. 2020 - May 2021 (1.5 Years)*

- Worked with Prof. Sidney D'Mello as part of his Emotive Computing Group. Research focus was on discourse and language modeling for team-based collaborative problem solving.
- Developed a re-usable pytorch based framework for training and experimenting with transformer based language models and multi-modal temporal sequential learning models using LSTMs.
- Co-authored 5 Peer-reviewed publications submitted to top-tier conferences and journals.
- Technologies Used: **Python**, **PyTorch**, **GPU Training**, **AWS**

Select Publications

Using artificial intelligence to assess personal qualities in college admissions *Science Advances 2023*

B. Lira, M. Gardner, A. Quirk, C. Stone, **Arjun Rao**, S. K. D'Mello, et.al. *[PDF]*

Worked on training and validation of machine learning models to automattically assess college admission essays.

CPSCoach: The Design and Implementation of Intelligent CPS Feedback *AIED 2023*

A.E.B. Stewart, **Arjun Rao**, S. K. D'Mello, et.al. *[PDF]*

Designed a fully-automated system that assesses and provides feedback on collaborative problem solving (CPS) competencies during remote collaborations

Do Speech-Based Collaboration Analytics Generalize Across Task Contexts? *LAK 2022*

S. L. Pugh, **Arjun Rao**, S. K. D'Mello, et.al (best paper nominee) *[PDF]*

Investigated the generalizability of language-models across two collaborative problem solving tasks, among 95 student teams who collaborated over video conferencing.

Say What? Automatic Modeling of CPS Skills from Student Speech in the Wild *EDM 2021*

S. L. Pugh, S. K. Subburaj, **Arjun Rao**, S. K. D'Mello, et. al. *[PDF]*

Investigated the feasibility of using automatic speech recognition (ASR) and transformer models to classify collaborative problem solving (CPS) skills from recorded speech in noisy environments.

Multimodal, Multiparty Modeling of Collaborative Problem Solving Performance *ICMI 2020*

S. K. Subburaj, A.E.B. Stewart, **Arjun Ramesh Rao**, S. K. D'Mello *[PDF]*

Investigated the accuracy of multi-modal ML models trained on facial expressions, acoustic-prosodics, eye gaze, and task context information, computed one-minute prior to the end of a game level, at predicting success at solving that level.