John Choi 최정혁

fast learner / smart builder

+82 10 3474 2178
isnbh0@gmail.com
isnbh0
isnbh0

I'm an analytical and resourceful software engineer with leadership experience and a track record of learning things quickly and using them to build robust solutions based on good ideas. I'm also a good technical communicator—I believe that the full potential of any technology can only be unlocked when the tech is effectively communicated, both among its developers and to its users.

Experience

2022.01.– **Senior Software Engineer**, *ailys*, Seoul

Led a cross-functional team on development of a web-based no-code ETL application called SSDP[1].

- Designed and led modernization of backend system behind a FastAPI gateway allowing for secure remote use of Spark interactive shell for stateful operations. Reduced processing time by as much as 92% while simplifying API for improved usability.
- Communicated closely with business team about client needs in order to assess requirements and prioritize development tasks.

2021.02.– **Software Engineer**, *ailys*, Seoul

2021.12. Led the conception, design, and development of SSDP from scratch, expanding ETL capabilities of our machine learning product by 15x.

- Designed the DAG-based core entity model and implemented it as a DSL in JSON Schema.
- Designed core processing logic using functional principles, implemented features on Spark, and exposed a REST API via Apache Livy.
- Led cross-functional discussions for designing UI/UX, focusing on intuitiveness of transitions between UI states, drawing inspiration from imagining how physical objects would work in similar situations.

2018.06.– **Machine Learning Scientist**, *ailys*, Seoul

2021.01. Developed DAVinCI LABS, a web-based no-code machine learning platform.

- Conducted research on supervised/unsupervised anomaly detection systems as well as manifold learning for data visualization and exploration. Built backend unsupervised learning module based on clean architecture principles using Dask/scikit-learn/TensorFlow. Contributed heavily to UI/UX design, focusing on consistency of user workflow.
- Developed weighted supervised modeling and researched use-case in accident rate prediction for auto insurance, improving predictive performance by as much as 17% over unweighted modeling.
- Using above modules, conducted several data analysis/R&D client projects worth up to ₩300M in revenue. *Clients include: DB Insurance (KR); AEON Financial Services (JP); KEB Hana Bank (KR); Korea Institute of Nuclear Safety (KR)*

2013.06.– **Private Instructor**, *Sehan Academy/Accel Education/Paul Academy/freelance*, Seoul

Teaching high school students math, physics, statistics, computer science, and data science, in classes of up to 20 students for up to 40 hours a week.

Technical Background

Skills

Languages	**Python, SQL**
Tools	**FastAPI, Pydantic, Spark, Dask**
Concepts	**MLOps, Clean Architecture, OOP, Functional Programming, Machine Learning**
Environments	**Docker, Linux/Shell, Git, GitHub CI/CD**

[1] Self-Service Data Preparation

Patents

2021.10. **Model generating method and apparatus for easy analysis, and data classifying method and apparatus using the model**, *Korean Patent No. 1023148480000*, Co-inventor

2021.10. **Optimal model seeking method and apparatus**, *Korean Patent No. 1023148470000*, Co-inventor

2021.06. **Method for applying user intension[*sic*] to unsupervised learning and apparatus therefor**, *Korean Patent No. 1022738680000*, Co-inventor

2021.06. **Method and apparatus for generating supervised learning model based on unsupervised learning, and method and apparatus for analyzing unsupervised learning model using the supervised learning model**, *Korean Patent No. 1022738670000*, Co-inventor

Personal Projects

2020.11.–2021.06. **From Nand to Tetris: Building a Modern Computer from First Principles**, *online course*, Python, Jack, Hack assembly, HDL

Completed all 12 projects on building up a working computer system (named "Hack") from just logic gates.

- Used an HDL[2] to simulate designing an ALU and a CPU from NAND gates.
- Wrote a VM translator in Python for translating stack-based VM bytecode into assembly language of the Hack system.
- Wrote a compiler in Python for converting the system's Jack high-level language into VM bytecode.
- Implemented a small operating system in Jack for the Jack platform on the Hack computer system.
- Implemented a version of 2048[3] in Jack and ran it on my OS.

Education

2008.08.–2012.05. **B.S. in Mathematics with Distinction**, *Harvey Mudd College*, Claremont, CA, USA

Coursework: Data Structures (CS), Computability and Logic (CS), Stochastic Processes, Operations Research

References

Available upon request.

[2] Hardware Description Language

[3] A game: https://play2048.co/