Zhihao (Johnson) **Du**

zhihao617@berkeley.edu | +1 (510) 833-4417

Links

Github
github.com/JohnsonJDDJ

Personal Website
zhihao.myxd.place
(for info on all projects)

Skills

Languages
HTML+CSS (since 2017)
Python (since 2017)
SQL (since 2019)
Java (since 2020)
R (since 2020)
C (since 2021)

Coursework
Natural Language Processing
Neural Networks
Statistical Learning
Service Operations Design
Database Systems
Reproducible Data Science
General Linear Models
Data Structures
Machine Structures
Linear Algebra
Probability Theory

Tools/Frameworks
Python Frameworks:
- Numpy/Pandas (since 2020)
- Matplotlib (since 2021)
- Scikitlearn (since 2021)
- Pytorch/Jax (since 2022)
- Flask (since 2023)

Git (since 2020)
DBeaver (Summer 2021)
MS Azure (since 2022)
MySQL Database (since 2021)
MongoDB (since 2023)
Unix System
Microsoft Office

Education

Applied for master's program with expected graduation date: 05/2024.

University of California, Berkeley Expected graduation date: 05/2023
B.A. Statistics, B.A. Computer Science | GPA: 3.8/4

Professional Experience

ETL Engineer Intern DataCVG Co Ltd | Shanghai, China | 05/2021 - 08/2021
Successfully optimized the database system of the client FosunPharma as part of the database services team. Engineered directly on the client's pharmaceutical database system by designing and implementing extract-transform-load (ETL) pipelines on semi-confidential relational data:

- Collectively designed target relational database architecture containing 100+ distinctive tables with ER diagram;
- Independently submitted >50% of the queries programmed to merge data from two source databases for the pipeline constructed with DBeaver;
- Debugged and overcame architecture failures through long diagnostic process and extensive communication with PM and client representatives

Academic Experience

Tutoring Deep Neural Networks | 01/2023 - Present
- Revised, improved and consolidated interactive demos on BERT and Encoders;
- Developed and peer-reviewed new content material on CNN concepts and applications;
- Led, facilitated and supported students on weekly discussion sections and homework parties

Research Assitant Real-time audio emotion classification | 01/2022 - Present
Advised by Prof. Dacher Keltner, assembled a police aggression discernment and early warning system powered by a parallel CNN Transformer neural network using pytorch, librosa, and pyaudio. The system is capable to classify emotions from streaming real-life audio speech data:

- Experimented, trained, tested, and finetuned the parallel neural system using emotional databases (RAVDESS, SAVEE) through MS Azure cloud platform;
- Spearheaded training data preprocessing with robust data augmentation techniques including Gaussian white noise, simulated room impulse response, and randomly sampled background noise, boosting performance at evaluation time to 71%;
- Programmed and installed real-time audio streaming and continuous model evaluation on a Raspberry Pi 4 device

Technical Projects

Howamidoing Full stack web developer | 01/2023 - Present
- Designed and developed college level course grade tracker and class standing estimator using the Flask framework, HTML, CSS, and JavaScript;
- Implemented and optimized JSONizable user data objects and stored in NoSQL database with connection to MongoDB, locally and through MongoDB Atlas

Domain and language translation on PINNs 11/2022 - 12/2022
- Composed a comprehensive and self-contained homework assignment with solution on hand-crafting a Physics-Informed Neural Networks (PINNs);
- Demonstrated key merits of PINNs using minimal compute and memory requirement, achieved outstanding distributivity and reproducibility.

Zilean Package for data mining and engineering pipelines | 05/2022 - 08/2022
Advised by Prof. Fernando Pérez, developed python package "zilean" that bridges the Riot Games API with traditional python data science APIs (scikitlearn, pandas) to produce data pipelines for multidimensional data ready for downstream ML or DL tasks;

- Programmed, tested and refined data mining/engineering algorithms for large semi-structured with rate limiting API request algorithms;
- Promoted and published as open source project with immediate collaborators after established CI/CD pipelines using Github Actions and Readthedocs documentation