Jonathan Pan

(408) 831-8048 • San Francisco, CA • jonathanpan@berkeley.edu • linkedin.com/in/jonathan-pan/

EDUCATION

UC Berkeley | B.A. Computer Science & Data Science May 2021

- Relevant Coursework: Machine Learning, Operating Systems, Database Systems, Computer Security, Efficient Algorithms and Intractable Problems, Artificial Intelligence, Data Structures, Principles and Techniques of Data Science, Machine Structures and Computer Architecture, Discrete Mathematics and Probability

SKILLS

- Languages (in order of proficiency): Python, Java, Go, C#, Scala, SQL, C, C++
- Platforms/Tools: Spark, Splunk, Grafana, Artifactory, Akka, AWS, Kinesis, S3, Firehose, Airflow, Kubernetes, Terraform, Selenium, Gradle, Datadog, Flask, Django, scipy, TensorFlow, NumPy, Pandas, Pytest, JUnit, CUnit

EXPERIENCE

Software Engineer, Einstein & Service Cloud *(Java, Python, SQL, AWS)* Jun. 2021 - Oct. 2022

Salesforce | San Francisco, CA

- Backend engineer for customer support Article Recommendations product; responsible for serving knowledge article suggestions for customer questions
- Ownership over availability metrics for team's essential services. Graphed and published duration, error, request rate, availability, and saturation data on a unified index and dashboard projected to improve debugging speed by 30%
- Created database and endpoints to capture pre-chat behavior as metadata for improved model training and accuracy by 40%
- Built conversation mining team to pipeline data to other AI projects as well as provide customers with more transparency
- Delivered article fields and support agent conversations as training data for suggestion model
- Collect customer feedback and implemented customer specific models to increase suggestion accuracy on average

Software Engineering Intern *(Java, Scala)* May 2020 - Aug 2020

Salesforce | San Francisco, CA

- Designed and built from ground up a low latency scalable data persistence pipeline service for case classification team and other ML teams on Einstein Platform to improve model training speed and accuracy
- Prototyped for Spark, Firehose, and JVM to weigh pros/cons for design
- Designed and implemented state diagram to ensure pipeline health in all possible cases of container failure
- Generalized pipelines to consume and perform parallelized map reduce while maintaining timestamp information for writes to S3 data store and MLLake

Software Engineering Intern *(Python, Airflow)* May 2019 - Aug 2019

Claritas Rx | South San Francisco, CA

- Used Apache Airflow to create DAGs that schedule tasks that poll and parallel pull files off SFTP servers for S3 storage
- Developed pytest quality assurance foundation that validated schemas and data quality
- Created automation testing that parsed hundreds of files for SQL models to be tested against the database
- Used Selenium to run web app tests and data scraping for quality assurance checks

Project Lead *(Python, BigData)* Jan. 2019 - May 2020

Berkeley UAVs | Berkeley, CA

- Lead Auto-nav sub team that focuses on shortest path waypointing and object detection/avoidance for drones
- Implemented several research papers to generate 3D flight paths for fixed winged drones
- Time based competition AUVSI-SUAS for avoiding physical and virtual obstacles that were received with python scripts

Back End Developer *(Python, GraphQL, Typescript)* Aug. 2020 - Jul 2021

Berkeleytime | Berkeley, CA

- Architected and implemented our API graphQL for new scheduler feature on berkeleytime.com
- Mapping system for courses that held different names depending on fall or spring semester that they are offered
- Automated email alerts for grade distribution updates and new semester course schedule releases

Spotify Classifier *(Python)* Dec. 2019 - Feb. 2020

Berkeley, CA

- Developed an app that would cluster related songs based on features instead of artists or genre
- Used Spotify API to get song metadata information such as danceability, speechiness, pitch, and many other features
- Implemented k-means and k-nearest neighbors algorithms to cluster songs into possible playlists