Pranav Varanasi

Cupertino, CA | 408-988-0099 | varanasipranav@gmail.com| linkedin.com/in/pranav-varanasi | github.com/vpranav5

EDUCATION

The University of Texas at Austin Expected December 2021
B.S, Computer Science Cumulative GPA: 3.59
Certificate in Applied Statistical Modeling (18 semester hours)
Certificate in Scientific Computation and Data Sciences (18 semester hours)
Relevant Coursework: Artificial Intelligence, Natural Language Processing,**Computational Biology**, Algorithms, Virtualization, Computer Vision, Operating Systems, Software Engineering, Parallel Computing, Computer Architecture, Data Structures

SKILLS

Programming Languages: Fluent: Java, Experienced: **Python**, **R**, C, C++, x86 Assembly, Exposure: JavaScript, Swift, ROS, MATLAB
Web/Mobile Technologies: Flask, React, Postman, XCode, Android Studio, Appium, **R Studio**
Big Data/Cloud: PySpark, Pandas, Azure Databricks, Azure, GCP, Terraform, PostgreSQL
Tools: Git, Jira, Microsoft PowerBI

WORK EXPERIENCE

VMware Palo Alto, CA May 2021 – August 2021
Incoming Analytics Engineer Intern

NCR Dallas, TX
Software Developer, Part-Time August 2020 – September 2020

- Developed **Terraform** scripts to automate creation of Microsoft Azure resources for existing Data Factory pipeline

Software Engineering Intern June 2020 – August 2020

- Developed machine learning tool to predict customer sales based on trend analysis of point of sale financial data
- Implemented "forecasting engine" for time-series trend analysis in **Azure Databricks** environment using **Facebook Prophet** sales forecasting package along with **PySpark** API and **Pandas** data analysis library
- Created a new end-to-end production pipeline to continuously read in historical data from SQL, pass data to forecasting engine to generate predictions, and render PowerBI report with forecast plots
- Reduced prediction error from 10,000% in pre-existing model to below 20% with initial Prophet-based approach

UT Austin AI Lab Austin, TX January 2019 – March 2020
Undergraduate Researcher

- Co-developed machine learning algorithm in **ROS, Python, and C++** to enable a robot to follow a person through a crowd
- Authored project research paper discussing how integration of DeepSORT trajectory tracker with Triplet Loss function resulted in better accuracy in "person following" algorithm
- Received UT CNS Award for Excellence in Computer Science for above project at 2020 Undergraduate Research Forum

NCR Dallas, TX
Software Engineering Intern May 2019 – August 2019

- Developed "Mobile Order by Vendor" iOS (**Swift**) & Android (**Java**) mobile app and enabled successful production deployment across 400 customer locations in hospitality vertical
- Implemented protocols to send real-time purchase orders between restaurants and their suppliers, prevent data loss, pull active vendor lists from customer databases, and automate scheduling of deliveries
- Created cross-platform UI Automation testing suites with Appium to verify app's user authentication, data service connections, icon layout, and delivery of purchase orders

PROJECTS

Quarantine Fighter (gitlab.com/vpranav5/quarantinefighter)

- Developed COVID-19 relief website that allows users to search for nearby hospitals and drugstores in a geographic area
- Hosted website and PostgreSQL database on GCP and created internal RESTful API to return drugstore/hospital info
- Implemented front-end UI with React and deployed website using Flask web server

Face Detector

- Developed multi-class image classifier in MATLAB designed to detect faces in a picture
- Used Adaptive Boosting machine learning techniques to implement Viola-Jones face detector
- Achieved less than 15% error rate by training classifier to identify 2000 "Haar-like" facial features

SysHandler

- Created system call handler for custom, interruptible OS built on QEMU-x86 emulator
- Implemented system call functionality to fork threads, replace process executable, and perform file I/O operations (open, read, write, seek)
- Built system resources (threads, locks, file system) using C, C++, and x86 Assembly