Aditya Saxena

GitHub: asaxena1415 | Phone: 6124422221
Email: asaxena1415@yahoo.com | Web: asaxena1415.github.io

WORK EXPERIENCE

The Washington Post **Washington, D.C.**
Software Engineering Intern *May 2019 - Aug 2019*

- Led the development of a whole pipeline in Java (**10000+ lines of code**) in a team of four, for tiling, augmenting, validating, sorting, and visualizing training data for CNN models related to newsletter recommendations. Reduced the training time of models by up to **35%**, and increased the accuracy of internally trained YOLOv2 model by **14%** (as measured by a withhold image test set).
- Developed scripts in Java to analyze the predictions to compute accuracy, confusion matrix, and visualize bathtub curves. Performed hyper parameter tuning, and experimental study to fine tune model architectures decreasing type-2 error by **5%**.
- Organized a seminar with the rest of the intern team and senior engineers on using Functional Programming principles in Java for demonstrating its flexible and powerful scripting ability. Practiced Agile (Scrum) with 8 other developers in 2 week sprints. Learned to work with various technologies like AWS, Joyent Triton, JIRA Confluence, Google Cloud, Git, Fabric, Bloodhound in an efficient and time-critical manner.

University of Minnesota **Minneapolis, M.N.**
Undergraduate Research Intern *Jul 2018 – Sep 2018*

- Worked to identify and engineer ways of predicting user behavior in the realm of viral marketing by monitoring users' social activity and network structure.
- Developed web scraping scripts in Java to extract data from Pinterest, Twitter, and BuzzFeed (using API and AIDR). Implemented graph processing algorithms (TSM and Multiple Cascade) and other community detection algorithms to identify gullible users in a network.
- Developed a pipeline in Python to process user network data and compute various other influence metrics like Degree Centrality, Betweenness Centrality, Path Centrality, Hubs and Authorities, Rank, Trust Scores and a POC to identify **top-k** influential users in a given social network using the above metrics.
- Developed an experiment to implement a recurrent neural network to find the **top-k** influential users in a given network of more than **50,000** users.

College of Science and Engineering, University of Minnesota **Minneapolis, MN**
Undergraduate Teaching Assistant, (CSCI 2041) Advanced Programming Principles *Jan 2019 – May 2019*

- Taught core functional programming paradigms, answered students' questions, debugged student programs and suggested ways to optimize and write better formatted code in labs (**30** students) and office hours (up to **15** students). Debugged the automatic feedback and grading scripts written in Python by the Graduate Teaching Assistants for the course.
- Organized the student database by advanced DBMS tools online for improving the grading process. Updated Web-based DokuWiki for the Computer Science Department.

EDUCATION

UNIVERSITY OF MINNESOTA
B.S. Computer Science | College of Science and Engineering | Aug 2017 - Dec 2019 (Expected)

- Cumulative GPA**: 3.73/4.0;** Major GPA**: 3.81/4.0**
- Activities: Dean's List | Recipient of Computer Science & Engineering Scholarship ($**2500** for two years) | Co-Founder & President, App Developers Club of the University of Minnesota | Campus Representative for International Students (ISSS), Minnesota Student Association.
- Relevant Coursework: Algorithms and Data Structures (Java), Advanced Programming Principles (Clojure & OCaml), Operating Systems (C), Database Fundamentals (SQL), Program Design & Development (C++), Machine Architecture and Organization, Linear Algebra, Discrete Mathematics, Probability and Statistics.

PROJECTS

Skills: Proficient in Java, C, C++, HTML/CSS, Swift/iPhone, Python; familiar with AWS, Javascript, Unix/ Linux, Node.js, Ruby on Rails, MATLAB. (Source code available at: github.com/asaxena1415)

- **Tic-Tac-Toe**:
  A parent name for the game that allows you to play different variants of the simple Tic-Tac-Toe game.Variants include Simple, Ultimate, Crooked, 3D, Misere, Notakto, Quixo, and Wild. Each of these variants allow people to play in "2-Player Mode" or "v/s Computer Mode". Leveraged engineering algorithms to make a sophisticated computer player that can play on 3 levels against the human player. Project also involved making a website that provides an interactive GUI to play different variants. Built in, Python, Java, JavaScript, HTML & CSS, Unity 3D, and PHP. Currently working toward using Machine Learning and Artificial Intelligence to increase the efficiency of the computer bot so that it can learn from its mistakes and play in a better manner for some advanced variants of the game.
- **Robot Game Simulator**:
  A semester long project for the class CSCI 3081W - Program Designa and Development at the University of Minnesota. The project involved developing a game that allows a player to control a player entity in an arena full of other entities like SuperBot, Robot, Obstacles, Recharge Station and Home Base. The objective of the game was to freeze all the robots befor the player runs out of battery or before all robots turn to superbots. Project involved application of various design patterns like strategy and observer (with justification) along with programming in **C++ (3000+ lines of code)**, documenting using Doxygen, and refactoring the code periodically using Git.
- **Bus Line Simulation**:
  Project involved using a Priority Queue and a combination of other data structures in Java (**5000+ lines of code**) to implement a discrete simulation of busses running in the city of Minneapolis. Project also involved documentation and collection of analysis for the bus simulation which were presented in an official project document at the end, suggesting and justifying the use of particular type of busses to increase efficiency by analysing the statistics collected.