Michael Pham

ktm-p.github.io
Email: ktmpham@berkeley.edu
Mobile: (916)-968-0563

Education

- **River City High School** West Sacramento, CA
  High School Diploma *Mar 2019 – Jun 2022*
  - GPA: 4.00
  - Graduated Salutatorian
- **University of California, Berkeley** Berkeley, CA
  B.A. in Computer Science and Mathematics *Aug 2022 – Present*
  Minor in Data Science
  - GPA: 3.87
  - Member of Upsilon Pi Epsilon Honor Society

Projects

- **A Secure File Sharing System** | Golang
  - Designed and implemented a secure file sharing system using cryptographic library functions.
  - Implemented file creation, appending, sharing, and deletion among multiple users. Users could also sign on from multiple devices and edits would be reflected across all accounts.
  - Utilized symmetric encryption, HMACs, and digital signatures to ensure security.
  - Extensively tested implementation, writing over three thousand lines of test code. Utilized fuzzing as well.
- **Berkeley Admissions Visualization** | Python, Matplotlib, NumPy, Pandas, Plotly, RegEx, Seaborn
  - Compiled data on Berkeley's Californian public school admissions, and created visualizations for it.
  - Filtered, regularized, and merged data from various sources with Pandas and RegEx.
  - Visualized data using scattermaps, choropleth maps, and other charts using Matplotlib, Seaborn, and Plotly.
- **Machine Learning** | Python, PyTorch
  - Used PyTorch on a variety of machine learning problems. Approximated a sinusoidal curve. Additionally, implemented language detection and handwriting recognition.
  - Utilized a two-layer Recurrent Neural Network for language recognition of words of differing lengths. Achieved an accuracy of over 80%.
  - Implemented a two-layer Linear Neural Network with ReLU activation function and Cross-Entropy Loss for handwriting recognition. Filtered the data using convolution, and then flattened it to enhance model performance. Achieved an accuracy of over 98%.
- **Reinforcement Learning** | Python, PyTorch
  - Utilized Reinforcement Learning to train Pac-Man agent to win. Achieved a win rate of over 90%.
  - Implemented value iteration, Q-Learning, Approximate Q-Leaning, and Deep Q-Learning using PyTorch.
  - Used Multi-Layered Linear Neural Network with ReLU activation and Mean Square Error Loss in Deep Q-Learning. Fine-tuned hyperparameters such as learning rate, hidden layer sizes, and number of training episodes.
- **Spam Classifier** | Python, Matplotlib, NumPy, Pandas, RegEx, scikit-learn, Seaborn
  - Created a spam email filter using a Logistic Regression model. Achieved an accuracy of 99.2% on given test data.
  - Cleaned and visualized data using Pandas, RegEx, Matplotlib, and Seaborn.
  - Fine-tuned hyperparameters by cross-validation with GridSearchCV.

Technical Skills

- **Programming Languages**: C, CSS, Golang, HTML, Java, Javascript, MATLAB, Python, R, RISC-V, Scheme, SQL
- **Frameworks/Libraries**: Matplotlib, Numpy, OpenMP, OpenMPI, Pandas, Plotly, Processing, PyTorch, scikit-learn, Seaborn, TensorFlow
- **Tools**: Docker, gdb, git, Logism, LaTeX, Valgrind
- **Mathematics**: Abstract Algebra, Discrete Mathematics, Linear Algebra, Linear Programming, Logic, Numerical Analysis, Real Analysis