Humza Iqbal

2700 Hearst Avenue – Berkeley, CA
☎ +1-(415)-686-5128 • ✉ humzaiqbal@berkeley.edu
humzaiqbal.github.io

Education

University of California, Berkeley **Berkeley**
BA. Computer Science *2014-2017*
Courses Taken: Data Structures, Discrete Math and Probability, Probability and Random Processes, Algorithms, Machine Learning

Experience

Elastica **San Jose, CA**
Machine Learning Intern, *May 2015 - Aug 2015*

- Built classification tools to perform multi-class classification among five classes: Business, Legal, Computing, Health, Finance, with over 90% accuracy on a test set as defined by custom metrics using state of the art machine learning techniques
- Implemented GUI to track classification performance, and allow users to manually change results

Bajcsy Group **Berkeley, CA**
Undergraduate Researcher, *Sep 2015 - Present*

- Analyzed human motion data using machine learning in order to figure out when people are exercising. Implemented classifiers to detect different kinds of exercises from the data

CalSol **Berkeley, CA**
Strategy Lead and Electrical Member, *Feb 2015 - Aug 2015*

- Analyzed data from car as well as track conditions in conjunction with machine learning to find the optimal route to race the car
- Wrote microcontroller code in C++ ensuring that the car runs safely and efficiently

Skills

Programming Languages: Python, Django, Java, C, C++, Matlab, HTML, SQL, MIPS, TeX

Workflow tools: Git, Bootstrap, UNIX, Eclipse, LPCXpresso, MapReduce, SciPy

Projects

- Wordnet: Used Java to make a program which interacts with the Google Ngram Dataset to analyze the history of words over a given time period, going as far back as the 1400s.
- Maximum Acyclic Subgraph Approximator: Used an ensemble of three different algorithms to approximate the Maximum Acyclic Subgraph problem (an NP hard problem)
- Random Forrest: Built a Random Forest learning machine capable of getting approximately an 85% accuracy rate on the census data set
- Gitlet: Used Java to build a miniature version of Git which allowed the user to commit, checkout, branch, rebase, and merge as regular Git would.
- Neural Net: Built a 2 layer Neural Network and trained it on the MNIST data set to achieve a 5% error on Kaggle data.