HUMZA IQBAL

8 Eagle Rock Road, Mill Valley, California 94941 | C: (415)686-5128 | humzaiqbal@berkeley.edu | W:humzaiqbal.github.io

Education

UC Berkeley B.A Computer Science, 2017

Related Coursework: CS 61B (Data Structures), CS 70 (Discrete Math and Probability), EE 126 (Probability and Random Processes), CS 170 (Algorithms), CS 189 (Machine Learning)

Research

Bajcsy group (**Sep 2015- Dec 2015**). I work with postdoc Qifei Wang on analyzing human motion data using machine learning in order to figure out when people are exercising using data from Microsoft Kinect. I have implemented classifiers to detect different kinds of exercises and actions from the data

Skills

Programming languages: Python, *Java, SQL, Matlab Scheme, C, MIPS*
Operating systems: UNIX
Version control*:* git

Experience

Elastica (Summer 2015)
. built classifiers to perform multi-class classification of large text documents using machine learning with over 90% accuracy on test data.
. built a pipeline to collect data and feed it into my classifier.
. built a dashboard to allow users to examine the classifiers performance and manually handle special cases

CalSol (Feb 2015 - Sep 2015) - Strategy team Lead and Electrical Team member for UC Berkeley Solar Car Team
. Analyzed data from car as well as track conditions to find optimal route to race the car
. Wrote microcontroller code ensuring that the car runs safely and efficiently

Hackathons

Lookout: As a part of the UC Berkeley hackathon (CalHacks), I implemented a tool to classify facebook messages as being for a different recipient than intended using the Project Oxford LUIS AP for Natural Language Processing in order to find the subjects that users would typically talk about, and detect if a message was not from those subjects

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.