HAO LUO

Woodland, CA 95776 | royluo@ucdavis.edu | 415-806-9033

EDUCATION

University of California, Davis | B.S in Statistics

JUN 2016 - DEC 2019 | Davis, CA

Calculus [A] | Computing Data with R [A] | Python Programming [B+] | Econometric [B+] | Statistical Data Technology [B]

De Anza College

JAN 2012 - SEP 2014 | Cupertino, CA

TECHNICAL SKILLS AND LANGUAGES

Programming Languages:	R, Python, C/ C++, SQL, Java
Platforms and Tools:	Linux Commands, Jupyter Notebook, Git, Statistical analysis
Language:	FLuent in English and Chinese (Mandarin and Cantonese)

PROJECT EXPERIENCE

Categorical Analysis for Disease prediction **UC Davis | 2019**

- Investigated relationships between disease on the one hand and other variables based on 5,419 random samples
- Performed stepwise model selection and interaction term independent test
- Fit Logistic regression model for the disease prediction
- Calculated the probability of the disease with given situations of a subject
- Specificity is 89.2823% and the error rate is 11.4987% from the model
- For more details, please click here for the source code

Data Visualizations on Housing Data **UC Davis | 2018**

- Analyzed a given random sample of 20, 000 housing sales from the San Francisco Bay Area
- Applied R Packages: stringr, lubridate, MASS, treemap, maps, and ggplot2
- Cleaned and stored data into proper type for analysis
- Removed outliers and conduct Linear Regression and Hypothesis test
- Developed a heatmap by utilizing longitude and latitude, to stack number of sales made on a block
- For more details, please click here for the report and source code

Machine Learning **UC Davis | 2018**

- Developed a machine learning algorithm to recognize test images (total 10 objects)
- Applied to different objects: airplanes, automobiles, birds, cats, deer, dogs, frog, horse, ship, and truck
- Given data source: 5,000 training images and 1,000 testing images, image is 32X32 RGB (600 images per object)
- Reduced cross validation runtime by calculating distance at the beginning of the program
  - In particular, calculated the distance matrix instead of subsetting one-fold at a time and calculated one-fold and the rest
- Applied Manhattan and Euclidean methods
  - The statistics indicate that Manhattan has a better result than Euclidean
  - Euclidean was supposed to have a better result if sample size was larger
- Achieved an overall accuracy rate of 30% due to complexity of the objects and the size of data set
- For more detail, please click here for the report and source code