Fall 2015:

Revisiting the uniqueness of simple demographics in US population [Java]

Analyzed US 2010 Census {Gender, Age, Location}, and Home/Work location pairing data to find anonymity value for US population. 63% population was uniquely identifiable; proving that release conditions of even simple demographics still require improvement.

Spring 2016:

Music Recommender System [Apache Spark, Python]

Created a music recommender system in Python, based on public song data from AudioScrobbler, using Collaborative Filtering techniques in Apache Spark.

Real Time Twitter Sentiment Analysis [Apache Spark, Python]

Processed live Twitter data stream using Apache Spark streaming service, with Kafka as a broker to handle data. Analyzed using Python and a word sentiment lexicon.

Predicting Bitcoin Price Variations [Python]

Developed a Python based predictor for variations in the price of the virtual cryptographic currency, using machine learning technique of Bayesian Regression.

Sentiment Analysis of IMDb Movie Reviews [Python]

Built a Python classifier for IMDb movie reviews using Logistic Regression (with NLP feature vectors) and Naïve Bayes (with Doc2Vec) methods.

Discovering Causality between Manufacturer-Retailer Price Channels [R]

Performed causality analysis of a Soft Drink system at a supermarket. Analysis done using R shows that it follows a vertical-integrated system rather than a Stackelberg leader model.

Market Segmentation of Social Network Data [Python, R]

Developed a Python application to identify communities in a college network using attributed graphs and igraph package. The communities are then self-verified by observing influence propagation over the network.

Optimal Adwords Placement [Python, R]

Replicated the solution for Adwords placement problem in Python, using Online Bipartite Graph Matching, to maximize revenue of the advertisers, and minimize number of unclaimed ad slots.

Anomaly Detection in Time Evolving Networks [Python]

Developed a Python application to detect anomalies in time evolving network graphs (e.g. p2p network), using the typical eigen-behavior of a feature set.

Bot for Agar.io Game [Processing, Java]

Developed an artificially intelligent bot for the multiplayer real time strategy game ‘*agar.io’*, using Java and Game AI techniques, such as movement behaviors, decision trees, and path-finding algorithms.

Predicting Optimality of Network Path [Python, R]

Explored a set of 15 chronological directed, dynamic, real-world graphs to predict the optimality of a given route in future graphs, using python-igraph, and R. Probabilistic prediction was used on models created using Random Forests, Linear Regression, Markov chains.

Fall 2016:

Anomaly Detection in Time Evolving Networks [Python]

Implemented NetSimile algorithm to detect anomalies on a collection of time-evolving graphs using Python. Calculated eigen-behavior over a sliding window to calculate moving average based on set of extracted features, such as degree of node, clustering coefficient, edges in ego-net.

Preventing network wide epidemic using Virus Propagation [Python]

Developed a Python program to simulate the propagation of a virus on a given static network, and analyze effective strength of virus against various immunization policies.

GitHub Bot for Predicting Bug Lifetime [node.js, R]

Designed a bot which tracked GitHub issues, and commented back a bug lifetime prediction, using node.js and GitHub API. Developed and trained the machine learning prediction model in R, using archived Bugzilla data and features such as bug severity, priority, no. of comments and contributors.

Spring 2017:

Course Management System [Java, PL/SQL]

Designed a relational database application while learning PL/SQL and Java, which manages data about students, staff, courses, and handles enrollment, billing, and transcripts.