KING'S COLLEGE LONDON

MASTER OF SCIENCE – INTELLIGENT SYSTEMS – 2018-19

PRACTICUM (UNMARKED)

7CCSMAIN - ARTIFICIAL INTELLIGENCE

LANGUAGE: PYTHON

I. GETTING STARTED WITH PACMAN
   a. Developed **GoWestAgent** which made Pacman go west on the grid whenever possible.
   b. Implemented **HungryAgent** which used information about the location of the food to move Pacman towards the nearest food.
   c. Developed **SurvivalAgent** which used the location of Pacman and the ghosts (and any other information that might be helpful) to stay alive as long as possible.

II. PACMAN DEALS WITH LIMITED VISIBILITY
   a. Developed **CornerSeekingAgent** which could sweep up all the food in an environment with partial observability and no ghosts.
   b. Created an agent which saved state information to prioritize seeking food and staying away from ghosts.
   c. Developed **MapBuildingAgent** which created a map data structure and used it to track every location that the agent had visited.

III. PACMAN STARTS TO PULL THINGS TOGETHER
   a. Developed Pacman agent which could cope with ghosts while clearing the food and win the games.

IV. PACMAN GETS GREEDY
   a. Implemented **MyGreedyAgent** which could move around the **non-deterministic** world by making its maximum expected utility (MEU) choice with or without ghosts or capsules.

V. PACMAN MEETS BELLMAN
   a. Developed more thoughtful **MDP-solver** which computed utility values of each space using value iteration and then let Pacman decide what to do in the non-deterministic world. Built similar agent using policy iteration.

VI. PACMAN IS RUNNING SCARED
   a. Created **MDPAgent** which could deal with ghosts by running value iteration at each step.

7CCSMML1 - MACHINE LEARNING

LANGUAGE: PYTHON

I. MACHINE LEARNING METRICS
   a. Explored the effectiveness of different machine learning techniques using **performance metrics** by training Decision Tree and k-NN classifier models with the popular **Iris** and **Wisconsin Breast Cancer** datasets.

II. REGRESSION AND GRADIENT DESCENT
   a. Implemented **univariate** and **multivariate** gradient descent procedures (stochastic and batch) w/o third party packages to develop regression models with self-generated linear datasets.
   b. Trained multivariate regression model on the **Boston Housing** data.

III. K-MEANS
   a. Implemented K-Means algorithm for the popular **Iris** dataset and compared the performance against **scikit-learn** K-Means implementation using **Rand index** (adjusted rand score).

IV. NAÏVE BAYES AND K-MEANS
   a. Developed Naïve Bayes classifier for the **Heart Disease** dataset from the UCI Machine Learning Repository and compared against the scikit-learn Naïve Bayes implementation.
   b. Experimented K-Means implementation on the **Stone Flakes** dataset from the UCI Machine Learning Repository and measured performance using the **Silhouette Coefficient** and the **Calinski-Harabaz** index.

V. SUPPORT VECTOR MACHINES
   a. Leveraged scikit-learn to carry out support vector classification and support vector regression with various **kernel functions** for the **banknote authentication dataset** from the UCI Machine Learning Repository and the **Wine Quality** dataset respectively.

    b. Developed **cubic**, **polynomial** and **Gaussian** kernels from scratch (without using scikit-learn) and examined performance with the SVM classifier.

VI. NEURAL NETWORKS

a. Experimented with different **neural network architectures** using scikit-learn against the popular **Iris** dataset and **seeds** dataset from the UCI Machine Learning Repository.
b. Developed a **single perceptron** and **single-layer multi-unit** neural network to classify the **Iris** dataset using the **error correction** procedure, **generalized** and **non-generalized delta** rules.
c. Developed **multilayer** neural network with backpropagation implementation for the **Iris** dataset.
d. Implemented neural network for the **seeds** dataset.

VII. EVOLUTIONARY ALGORITHMS

a. Implemented **genetic algorithm** to develop a control mechanism using **exploitation**, **mutation** and **crossover** methods that could help an agent decide what to do in the environment.

VIII. REINFORCEMENT LEARNING

a. Developed **bandit** learners using the **ϵ-greedy** strategy to let an agent move around the environment while learning the value of each state or the value of each action in each state.