Engineering and Applied Science Programs for Professionals
Johns Hopkins University, Whiting School of Engineering
685.621 Algorithms for Data Science
Programming Assignment 2
Assigned with Module 8, Due at the end of Module 14

Total Points 100/100

In this programming assignment students are to apply Machine Learning using the features generated from the MNIST data set and build AI's (search algorithms) for the game of Tic-Tac-Toe using Game Theory search techniques. This assignment is to be implemented by the individual student while not collaborative you are allowed to discuss on the PA2 discussion area. Please follow the requirements provided in the Programming Assignment Guideline.

> **Info:** You will see the term "Built-ins are not allowed" throughout this document. This means that if there is a function out there that computes the answer in one line for the provided task, it is not allowed. For example, we ask you to build a Bayes Classifier. You cannot do the following for that:
>
> ```
> from sklearn.datasets import load_iris
> from sklearn.model_selection import train_test_split
> from sklearn.naive_bayes import GaussianNB
> X, y = load_iris(return_X_y=True)
> X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)
> gnb = GaussianNB()
> y_pred = gnb.fit(X_train, y_train).predict(X_test)
> ```
>
> This would be considered using a built-in and would receive no points. Summarily, if the library function solves the main question of the problem it is not allowed, any other intermediate functions to get to the final algorithmic solution can be used. If you have specific questions please send a note to the professors.

Problem 1 - Feature Ranking Methods (FLDR, Decision Tree Classification)

30 Points Total

1. (5 points) Using the MNIST dataset, rank the features using Fisher's Linear Discriminant Ratio as the criteria (you may utilize your implementation from the previous assignment).
2. (25 points total) For part 2, you will create and fit a decision tree classifier (using the scikit learn built in function) offering a new avenue for feature ranking.
   (a) (5 points) Create an algorithm that will accept your dataset and fit a decision tree classifier to it.
   (b) (5 points) Using your algorithm in part 2.1, run a 5 fold cross validation and score the model based on its ability to classify accurately.
   (c) (2.5 points) From the model you fit, extract the feature importances.
   (d) (2.5 points) Add unit testing for the feature ranking in both methods. (Hint: You have already done a smaller dataset in PA1 that should allow you to test your implementation)

(e) (5 points) Plot your decision tree graph and examine the key parameters displayed. Offer an analysis for how the decision tree runs its predict method along with an analysis of the runtime complexity.

(f) (5 points) Compare the features importances from the decision tree classifier with the feature ranking using FLDR. Consider what sets the two methods apart and offer a discussion on the benefits of each approach.

Problem 2 - Machine Learning

45 Points Total

In this problem the features generated from HW2 for the numerical data set are to be used. This is the starting point for this problem. A minimum of 5,000 observations need to be used in the problem. A data set called ”trainFeatures42k.xls” will also be provided if needed. The updated data is provide as an Excel file with 42,000 observations and 60 features, 20 from each direction. In this assignment data processing and machine learning techniques need to be combined, the ”best” combination is determined by the best classification accuracy:

1. [5 points] Use a minimum of one of following data preprocessing methods (If more than one method, the processing order is up to you. Built-ins are not allowed.):
   (a) Data Normalization
   (b) Outlier Removal
   (c) Feature Ranking and Selection
   (d) Dimensionality Reduction
2. [5 points] Even if you did not choose Outlier Removal as a preprocessing method, you are going to use an outlier removal algorithm and unit test it using the iris dataset. You will determine if the data contains an outlier by plotting each class individually, the key is to plot two features at a time, *n* different combinations, e.g., feature 1 vs feature 2, feature 1 vs feature 3, etc. (**Required:** Use the code discussed in Module 13)
3. [30 points total] Use the following Machine Learning (ML) techniques (built-ins are not allowed):
   (a) [10 points] Bayes Classifier (built-ins not allowed)
   (b) [10 points] Parzen Window (Gaussian kernel)
   (c) [10 points] Support Vector Machine using your implementation of optimization to identify the support vectors (built-ins are not allowed)
4. [5 points total] Provide an analysis of your results:
   (a) [2.5 points] What combination from the above methods gave the best results? The ”best results” is considered the highest classification accuracy for the 10 digits from the 5-fold cross validation results.
   (b) [2.5 points] Was there any part of the combination of the techniques used computationally expensive and why?

```
function Min-max-Search(game, state) returns an action
  player ← game.To-Move(state)
  value, move ← Max-Value(game, state)
  return move
```

```
function Max-Value(game, state) returns a (utility, move) pair
  if game.Is-Terminal(state) then return game.Utility(state, player), null
  $v \leftarrow -\infty$
  for each a in game.Actions(state) do
    v2, a2 ← Min-Value(game, game.Result(state, a))
    if $v2 > v$ then
      v, move ← v2, a
  return v, move
```

```
function Min-Value(game, state) returns returns a(utility, move) pair
  if game.Is-Terminal(state) then return game.Utility(state, player), null
  $v \leftarrow +\infty$
  for each a in game.Actions(state) do
    v2, a2 ← Max-Value(game, game.Max-Value(state, a))
    if $v2 < v$ then
      v, move ← v2, a
  return v, move
```

Figure 1: An algorithm for calculating the optimal move using MiniMax - the move that leads to a terminal state with maximum utility, under the assumption that the opponent plays to minimize utility. The functions Max-Value and Min-Value go through the whole game tree, all the way to the leaves, to determine the backed-up value of a state and the move to get there.

Problem 3 - Game Theory (Search Algorithms)

25 Points Total

In the tic-tac-toe code provided add the following method to allow an unbeatable AI in your game. Implement either MiniMax or Alpha Beta to play against and allow the play to choose the skill level.

1. Random Move - Skill Level Easy
2. Utility Based Agent and Goal Based Agent (PA1) - Skill Level Medium
3. [25 points] Skill Level Hard - Implemented the MiniMax and Alpha Beta algorithm from the Game Theory document for the tic-tac-toe game. You will need to alter the provided pseudo code to input the game board.