Pong Option B

**Libraries used**

* Pygame
  + The pygame package is used to create the objects and surface on which the game takes place.
* Pandas
  + Pandas is used to insert data into dataframes, which can be manipulated in the desired forms. Pandas also integrates well with sklearn.
* Sklearn
  + Sklearn is used to perform the linear regression which is required to predict the position of the ball, and how the agent plays.

**Model Architecture**

**Global variables specification**

The script starts with global variable specification. Parameters such as window size and colour schemes are set. The user can alter these to his or her preference.

**Preliminary Functions- Building the Arena and Objects**

This section consists of various functions which are called in the main function. The main function is the one which ultimately initiates and loops through the respective game events.

The Arena and surface is defined at outset.

Next, we create the objects by drawing them with pygame.draw(). The objects required are the ball and the two paddles.

Collision Contingencies are defined i.e how the ball will react upon touching the game boundaries and paddles.

Next, the scoring mechanism is defined and scripted to display.

The next part is to define the way the paddles behave. Each paddle is assigned an AI agent, contingent on ball position and direction.

**Main Function**

The Main Function utilizes the preliminary functions defined beforehand.

We initiate the pygame package at outset. Specific parameters such as the starting position of each object and counting process starting values are set.

Now we create the loop for game events. We insert certain variables into a training data frame, which will populate as the game events loop. We also make sure that the game can quit. The linear model is parameterized and used to predict where the ball will be.

Finally, we perform miscellaneous if statements that define certain contingencies i.e such as when the game should automatically end.

**Training Process**

The training data table is populated with the following data:

* The position of paddle1
* The vertical distance of paddle1 and the ball
* The vertical and horizontal direction of the ball
* The horizontal position of the ball

As the game events loop, the training set populates. The horizontal position of the ball (ak.ka ballx) is linearly regressed by the respective variables through sklearn’s linear\_model package. Ballx is predicted for the current game event.

The AI of paddle2 is a function of the predicted ballx values. Paddle2 will follow the predicted ballx values.

The regressed variables are independent and therefore linear regression is statistically sound, with unbiased estimators. Further statistical analysis is required to perform residual diagnostics i.e white noise testing.

**User Input**

The main function is a function of player1 AI element {0,1}. Main(AI=1) assigns an AI to paddle1, which plays the perfect game of ping pong. Main(AI=0) assigns control over paddle1 to the user, which controls the paddle with a mouse.

Other user parameters (defined at the start of the script) include:

* Max Score before game ends
* The speed of the game. Please note that game speed must be an element of Mod5, since collision contingencies are restrained by pixel size, which are in units of 5, ceterus paribus.

Appendix A

1. K- means clustering can be used. Ballx centroids can be made, which can then be used to direct paddle2. Neural networking is also an alternative option.
2. Supervised learning uses pre-trained datasets to formulate a fixed mathematical pattern which the response variable will follow, irrespective of new information/data. I.e the coefficients will never change their value. Unsupervised learning is when the model adapts to new information and alters its coefficients to minimize the sum of squares error continuously.
3. Classification machine learning yields output as singletons, and can provide strings. Regression produces output which is in the form of a range. An example of classification is the language problem.
4. Class imbalance can lead to a large bias. This is often due to a high level of statistical dependency between the “independent” variables.
5. Variance transformation and scaling techniques are popular to mitigate the effects of class imbalances. Proportional models is also viable to deal with this.
6. Deep learning takes data representation into account whereas non-deep learning does not.