**AI Project Part 2 – Street Fighter**

**Mahad Ahmed 20i – 0426, Section F**

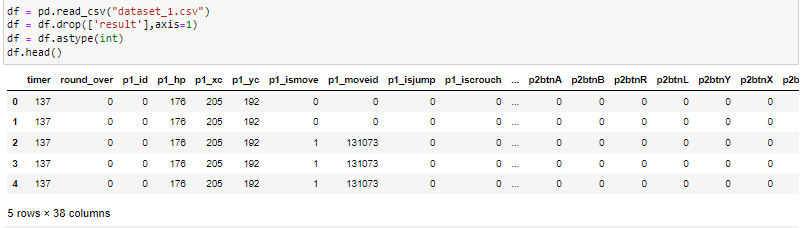
**Alishba Asif 20i – 0582, Section F**

**Purpose and Objective**

The purpose of this project was to train a street fighter bot using machine learning algorithms so that it could fight in the game on its own. We had to first collect the dataset, train the model, then change the code so our bot would fight depending on the model.

**Collecting the Dataset**

To collect the dataset, we used the state of the game and the parameters present. The parameters we used included the buttons, characteristics like ID, health, and movement status for both players. This would be done at a very frequent interval of 20-30 times a second. We played different characters against different characters to get a robust dataset which would be applied on different characters. The dataset looked like the following:



**Training the Model**

*Feature Engineering*

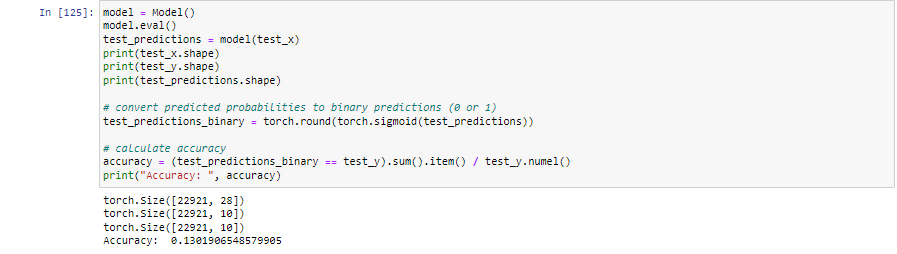
The model training required a few steps. Firstly, we indulged in feature selection. The p1btn rows (the data of the buttons we collected from the bot when training) were very sparse and so did not help our model so we removed it along with timer and result of the match as this did not help our model.

Secondly, we standardized our numeric features like health and coordinates, so that the model can comprehend them better as it adds context to the numeric value.

Finally, we converted our features to NumPy arrays and float values so that the datatype was constant throughout the data frame.

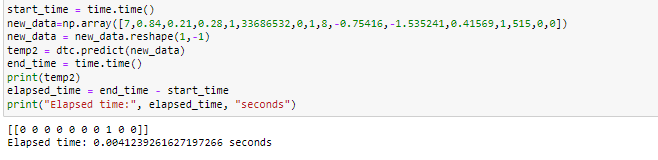
*Model Selection*

Firstly, A neural network model using pytorch with 3 layers was tried but it gave a very slow performance and gave a low accuracy.



Afterwards a multi–Output Classifier based on Random Forest Classifier was tried. It gave good accuracy but was very slow and when used in the game dropped the fps to 0.

Finally, the model we chose to go with was a decision tree classifier as this gave a good accuracy but was also very fast making predictions in only 0.002 seconds on average.



**Making the Bot Fight**

After loading our model through a pickle file into the initializer of the bot class we changed the bot fight function to now fight according to our model. Each time the fight function would be called, we would get the data from the parameters we would need to give to the model. The model would predict every 30 frames, each second. It would give the predictions of what buttons the bot should press in reply to the opponents moves and game status. We would then have the bot press the buttons and perform the moves our model would predict for it.

**Conclusions**

Through testing we see the bot fights when the opponent is far, but it also goes defensive when attacked and jumps back. However, the lack of different moves made could be possible to our limited dataset. If the dataset would be increased the model trained would be better.