Xinyun Yu
2870165
EECS 731
Project 2

To Be Or Not To Be

The goal of this project is to apply classification models on existing data to help with predicting future ones. The one I am using for this project is the decision tree.

Before applying the decision tree on the data, I dropped a few features so that what remained in the data set are those that would be helpful for prediction. More specifically, I deleted the data that contains “NAN” in it, which indicate it is not a valid tuple, as well as “DataLine” and “PlayerLine”. The reason that “PlayerLine” is dropped from the table is because it is possible to predict which player it belongs to by only looking at one single line, which makes it too sparse to be trained.

Noted also that the data entry in the table is not in numerical type, which is required for applying a decision tree with python. The conversion of a data from a label into a numeric form involves 2 possible algorithms in python, one hot encoding and label encoding. Compared to label encoding which labels every feature with a number, one hot encoding splits the column into several columns and marks it with a ‘1’ or ‘0’, depending on what value it holds. This process, which would only increase the depth of the tree as a result, is very similar to how decision tree proceeds - A question with binary answer is asked every time the decision needs to make a decision. Hence, one hot encoding does not help very much when applying a decision tree on the data. Therefore, I chose to only apply label encoding for this project.

But I am aware that the label encoding could bring some problems too. When labeling the features with numbers, it is possible that the decision tree is making decisions based on its numerical properties. For example, it is making a decision based on which one of the two has a greater value while such property does not work on what numbers are representing(i.e. the Act Scene Line.) This could bring some errors into the prediction.

After preprocessing the data, I could apply the decision tree model which involves training and testing. The training percentage was set to be 20%.