End-Term Project

Data Set-jungle_chess_2pcs_endgame_rat_elephant

This dataset is part of a collection datasets based on the game "Jungle Chess" (a.k.a. Dou Shou Qi). Dou Shou Qi is a game in which two players control a number of pieces, each of them aiming to move one of their pieces onto a given square. He and his team implemented an engine for analysing the game. Moreover, they created a series of endgame table bases containing all configurations with up to four pieces. These table bases are the first steps towards theoretically solving the game. Finally, they constructed decision trees based on the endgame table bases. In this note they report on some interesting patterns.

This data set has 47 features and 5880 instances, out of which 20features are object type variable.

```
white_piece0_strength                          5880 non-null int64
white_piece0_file                              5880 non-null int64
white_piece0_rank                              5880 non-null int64
white_piece0_advanced                          5880 non-null object
white_piece0_distanceto_white_den              5880 non-null int64
white_piece0_distanceto_black_den              5880 non-null int64
white_piece0_unopposedto_black_den_length      5880 non-null int64
white_piece0_unopposedto_black_den_shortest    5880 non-null object
white_piece0_movesto_white_den                 5880 non-null int64
white_piece0_movesto_black_den                 5880 non-null int64
white_piece0_in_trap                           5880 non-null object
white_piece0_in_water                          5880 non-null object
white_piece0_can_cross                         5880 non-null object
white_piece0_can_cross_shortest                5880 non-null object
white_piece0_unopposed_to_bank                 5880 non-null object
white_piece0_distanceto_black_piece0           5880 non-null object
white_piece0_distanceto_black_piece0_parity    5880 non-null object
white_piece0_nextto_black_piece0               5880 non-null object
black_piece0_strength                          5880 non-null int64
black_piece0_file                              5880 non-null int64
black_piece0_rank                              5880 non-null int64
black_piece0_advanced                          5880 non-null object
black_piece0_distanceto_white_den              5880 non-null int64
black_piece0_distanceto_black_den              5880 non-null int64
black_piece0_movesto_white_den                 5880 non-null int64
black_piece0_movesto_black_den                 5880 non-null int64
black_piece0_unopposedto_white_den_length      5880 non-null int64
black_piece0_unopposedto_white_den_shortest    5880 non-null object
black_piece0_in_trap                           5880 non-null object
black_piece0_in_water                          5880 non-null object
black_piece0_can_cross                         5880 non-null object
black_piece0_can_cross_shortest                5880 non-null object
black_piece0_unopposed_to_bank                 5880 non-null object
```

```
black_piece0_at_d7                             5880 non-null object
black_piece0_distanceto_white_piece0           5880 non-null object
black_piece0_distanceto_white_piece0_parity    5880 non-null object
black_piece0_nextto_white_piece0               5880 non-null object
highest_strength                               5880 non-null object
closest_to_den                                 5880 non-null object
closest_to_den_diff                            5880 non-null object
fastest_to_den                                 5880 non-null object
fastest_to_den_diff                            5880 non-null object
white_unopposed_to_den                         5880 non-null object
black_unopposed_to_den                         5880 non-null object
white_unopposed_to_den_quick_detour            5880 non-null object
black_unopposed_to_den_quick_detour            5880 non-null object
class                                          5880 non-null object
```

The focus is on object data type variable. at the same time, we have to check the variance of the numerical variable.

```
white_piece0_strength                        12.250000
white_piece0_file                             4.327551
white_piece0_rank                             7.046259
white_piece0_distanceto_white_den             8.122780
white_piece0_distanceto_black_den             8.122780
white_piece0_unopposedto_black_den_length     7.713681
white_piece0_movesto_white_den                8.122780
white_piece0_movesto_black_den                8.122780
black_piece0_strength                        12.250000
black_piece0_file                             4.327551
black_piece0_rank                             7.046259
black_piece0_distanceto_white_den             8.122780
black_piece0_distanceto_black_den             8.122780
black_piece0_movesto_white_den                8.122780
black_piece0_movesto_black_den                8.122780
black_piece0_unopposedto_white_den_length     6.276588
```

What is Variance? Why we check for Variance?

Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before.

But the main concern of this data set is that the variables are shown as object type but values in side the data side are numerical. So, we are going to do Label encoder to change the data type in to int and also, we are doing One-hot Encoding to convert instances in to dummy variable.

Label encoding

To convert categorical text data into model-understandable numerical data we used Label Encoder.

One-Hot Encoding

Depending on the data we have, we might run into situations where, after label encoding, we might confuse our model into thinking that a column has data with some kind of order or hierarchy, when we clearly don’t have it. To avoid this, we used one hot encoding.

After doing Label encoding and one-hot encoding we covert all object type features to integer type. Now to check the accuracy of the model I used Logistic regression and Neural network.

By using Logistic regression, I got an accuracy of 92%. To improve the model again I used Neural Network and got an accuracy of 99%.