Web Intelligence Project Documentation

Alexandra-Denisa Pert

February 2020

1 Introduction

The project was modeled to complete the following task: choose the best classifier for predicting the winner of the Australian Open 2019.

1.1 Components

The project has 3 major components:

- tennis folders: 2015-2019.xlsx for training our algorithm (with the 2019.xlsx having the data only until the Australia Open begins) and a partial 2019.xlsx containing only the Australian Open matches;
- main.ipynb: containing the prediction models;
- utils.py: containing the functions used for the data processing part;

1.2 General Idea

The general idea is to use data from all the tennis competitions that took place from 2015 to 2019 (until the Australian Open) and to fit them into 2 models:

- RandomForestClassifier;
- LogisticalRegression;

and choose the one with the best score for simulating the matches for Australian Open 2019.

2 Analysis of the code

In this section I will discuss the steps I followed for creating the program.

2.1 Data Handling

The first step is to prepare the data for the models by choosing the X and Y parts of the model. Initially the data set was organised under the winner/loser format, but I considered the P1/P2 format and an additional column(P1_won) that can signal if P1 or P2 won (True/False for the P1 column). The P1_won is going to be used as the Y for the models.

The X of the models will consists of some initial features found in the data set plus two custom features added by me. The already existing features used are:

- Tournament: the match tournament;
- Court: the match place(outdoor/indoor);
- Surface: the court surface(hard/clay/grass);
- Round: the hierarchical round;
- Best of: number of played sets;
- Series: the series of the match;
- P1Rank: the rank of P1 at the time of the match;
- P2Rank: the rank of P2 at the time of the match;
- P1Pts: the number of points for P1 at the time of the match;
- P2Pts: the number of points for P2 at the time of the match;
- AvgP1 betting score for P1;
- AvgP2 betting score for P2.

The custom features added are:

- P1_Experience;
- P2_Experience;
- P2_W/L;
- P2_W/L;

The experience is represented by the number of matches played by the player until the current match. The W/L ratio is represented by: $\frac{Nr\_Loses}{Nr\_Wins} * 100$ where Nr_Loses is the number of loses the player has until the match date and Nr_Wins the total number of wins until the match date.

Implementation: All the functions that are dealing with the organization of the data are found in utils.py. The main idea is that the Winner / Loser columns were changed in P1/P2 columns. After this, the P1 column contained all the time the winner, so a P1_won column was created with pseudo random values chosen from 1 and 0. The data frame was parsed and every time the P1_won was 0, the P1 and P2 were switched, so the data remained correct.Also all NaN values were replaced with the mean() value.

Factorization: Each non-numerical feature from the data set was factorized using the pandas function: factorize().

2.2 Models

For the prediction of data, 2 models were implemented using the sklearn lib and then compared :

- RandomForestClassifier;
- LogisticalRegression;

2.2.1 RandomForestClassifier

For the Random Forest, a comparison was made in regards with the criterion:

- Gini criterion;
- entropy(Informational Gain criterion);

The criterions were compared(Gini fit with a score of approx 0.82 and the entropy fit with aprox 0.81) and the one which gave the best score was used further.

2.2.2 LogisticalRegression

In regards with this model only one parameter was set custom: max_iter, because the model was reaching the maximum limit on a data frame this big.

2.2.3 Training/Test set

The training and test set were chosen with the skllearn function: $train\_test\_split()$ and the test set was set at 0.25 of the training test.

2.2.4 Comparison

The RandomForestClassifier seems to be the one with the highest score of aprox 0.82,compared to the LogisticalRegression, which is around: 0.65 on the test set.

2.3 The score of the models on the 2019 AO data

I observed that the Random Forest had a small boost in score of almost 0.02(using the Gini criterion) and the Logistical regression had a boost of: 0.05 (0.65 on test set, 0.70 on Australian Open, still being slower than the RandomForestClassifier).

3 Conclusion

In conclusion: the Random Forest has a better performance for a data set this big. The Random Forest score can be improved by adding new features and by analysing the importance of the current ones.