Université Nice Côte d'Azur
Enrico FORMENTI

Année 2023/24
M1 Info
NNL

Project
Football match results predictor
Data preparation

Contents

1 The data sets

In this here project we are going to consider data sets containing information about national football teams and their results in international competitions. You should download the following data sets from the Kaggle's website:

- "International football results from 1872 to 2023" which can be downloaded from here;
- "FIFA World Ranking 1992-2023" which can be downloaded from here.

The first one is a collection of 45,000+ results of international football matches starting from the very first official match in 1872 up to 2023 accounted from different international football associations. The second data set containts the FIFA rankings for men's national teams from 1992 up to (July) 2023.

We will make reference to the file `results.csv` in the first data set as `results` and to the file `fifa_ranking-2023-07-20.csv` as `ranking`.

2 The first things to do

Open both data sets with your preferred spreadsheet or text editor and take a look at their structure and content. You will remark that there are columns with quite similar contents between the two files. For example `country` contained in `results` and `country_full` in `ranking`. First of all, have a close look at the country names. It is clear that some of them have changed over time. Propose a *cutting year ie.* a year after

which the country names should be approximatively the same as actual names and explain the method or reasoning you made to choose this cutting year.

Drop all data with a date prior to the cutting year you have chosen. Now, we should have data which is consistent with countries enlisted in current competitions.

2.1 Further fixes on the country list

Have another close look at the countries list in both data sets. Delete all repetitions and keep a unique name for each country. For example, you may find both "Cabo Verde" and "Cape Verde Islands" or "St Kitts and Nevis" and "St. Kitts and Nevis", *etc.*

Then, *homogenize* the country names among the `results` and `ranking` data set. For example in one there is "St. Kitts and Nevis" and in the other "Saint Kitts and Nevis"; we want to keep "Saint Kitts and Nevis" for both data sets.

It is probably a good idea to printout a symmetric difference between the country names contained in `results` and in `ranking` data sets to have an idea if you did a good job.

2.2 Resampling

Both `ranking` data and `results` shall be indexed according to the `date`. However, remark that the dates in the two data bases does not necessarily coincide since they were asynchronously made. For this reason, in order to be able to retrieve coherent data to be added to the `results` data base later, you should resample (upsample indeed) the ranking data on a daily basis. This operation is going to create lots of new rows and some of them have necessarily invalid data so you have to instruct `Pandas` on how to fill the missing data (have a look at `Pandas`' resampler here and consider the methods `first` and `fillna`).

For more details on the resampling process, its meaning and how it can be implemented in `Pandas`, you can read this article.

2.3 Merging data

In order to conveniently processing data contained in both data sets you are invited to build a new dataframe named `rera` which takes `results` and merges with a certain number of columns from `ranking`. More precisely, per each row in `result`, we want to add `total_points`, `previous_points`, `rank`, `rank_change` for both home team and away team (use the suffix `_home` and `_away` to distinguish them) from `ranking` data set.

3 Adding more features

Propose interesting features which can be useful to predict the final result of a match. For example, you may consider average number of goals made in the last 5 matches, average number of points made during last five matches, rank increase during last five matches, or average points made during last 5 direct matches against the same team, etc. Add all the proposed new features to `rera` in a consistent way.

Make a correlation analysis to see how features are correlated. Select those that you think are strongly correlated and use other feature selection criteria to choose if drop them or not.

4 Final tweaking

Perform a last check for dropping from `rera` all columns which are not essential or which contain redundant information. Drop all rows which still contain missing data.