Francisco Sanz Estévez

causaLens

Data Scientist- Innovative Applications of Causal AI Challenge

The chosen use-case is the prediction of the English Premier League results.

I chose this case because I thought it was original, and football is one of my passions.

For the problem, I couldn't find a database ready for action, so I had to find my way into creating a consistent DB. For this, I searched through Kaggle until I found the files that I wanted:

1. From https://www.football-data.co.uk/ I gathered the results and betting odds from the most famous betting companies.
2. From https://www.kaggle.com/shubhmamp/english-premier-league-match-data I was able to find the lineups and match stats for every game in between 2014 to 2018.
3. From https://www.kaggle.com/stefanoleone992/fifa-20-complete-player-dataset I found the attributes of all the players from the game FIFA since 2014.

I then created a database containing all of this for the seasons 2014/15 to 2017/18.

The task to create the database was tedious because most of the key columns of each file weren't a straight match. I solved it by using the Levenshtein distance provided by the library FuzzyWuzzy to find the columns that were the best match.

For the database I used in the forecasting, I deleted all the in-game statistics that wouldn't be known prior the game (shots, fouls, cards etc. in game), and I added the overall rating of the starter 11 of each team from the FIFA database. (FIFA provides a weekly update after the game, I think using the weekly update would provide a much better result since the overall of the players changes depending on how well they played the last game).

As for the forecasting, I focused on SKLearn.

I used Randomforest, GradientBoostMachine and Elasticnet.

For the hyperparameter tuning, since we can't use a CV K-fold, I used sklearn TimeSeriesSplit to provide 5 splits in the database, and then I created my own Random Grid Search.

I used the random search instead of the traditional brute force grid search to speed up the process. It is very likely that a random search will land in a sweet spot way faster than a brute-force approach.