PYTHON FOR DATA ANALYSIS FINAL PROJECT

FOOTBALL PLAYERS PRICE PREDICTION

Etienne Fayolle & Pierre Archambault

SUBJECT & OBJECTIVE

Our initial dataset was not useable, so we were offered to work on the subject we like. Being football fans, we looked up on Kaggle for a dataset linked to it and found a very interesting one: “Football player price prediction”. Having our teacher’s backing, we decided to make this our final exam’s subject.

Here is the header of the dataset with some selected variables:

name	first_name	age	team	goals_sele...	position	price
Etheridge	Neil	28	Cardiff	0	Goalkeeper	4000000.0
LéoNatel		21	APOELNicosia	0	RightWinger	800000.0
Vidigal	André	20	APOELNicosia	0	RightWinger	650000.0
Antoniou	BaiAndrew	21	AlkiOroklini	0	SecondStriker	50000.0

This is a multivariate regression problem where the player’s price is the target variable and his performances, physical attributes, and more, are the explicative variables.

Player prices have become such an important stakeholder in today’s football world, with transfer windows often going crazy, and we find it very exciting to be able to make this kind of predications.

SUBJECT BOUNDARIES

After having a look around the other works that have been done with this dataset, it seems that the difficulty resides in the fact that leagues from all continents are considered. The result is an accuracy loss coming from the difference in terms of level and player pricing between leagues.

Therefore, we decided to consider only the players from the top 5 leagues:

- France: Ligue 1
- Germany: Bundesliga
- Italy: Serie A
- England: Premier league
- Spain: La liga

The prediction accuracy will surely be better, and we can focus on the leagues where the market windows are the most intense, which is more interesting.

VARIABLES OVERVIEW

name	first_name	age	nation	league	team	goals_selection	selections_nation	position	price	end_contract	goal_champ
Messi	Lionel	31	Argentina	LaLiga	FCBarcelona	65	128	RightWinger	160000000.0	3.0	410.0

assist_champ	own_goal_champ	sub_on_champ	sub_out_champ	yellow_card_champ	second_yellow_card_champ	red_card_champ	penalty_goal_champ
166.0	0.0	51.0	51.0	43.0	0.0	0.0	49.0

conceded_goal_champ	clean_sheet_champ	goal_cup	assist_cup	own_goal_cup	sub_on_cup	sub_out_cup	yellow_card_cup	second_yellow_card_cup
0.0	0	63.0	39.0	0.0	13.0	8.0	14.0	0.0

red_card_cup	penalty_goal_cup	conceded_goal_cup	clean_sheet_cup	goal_continent	assist_continent	own_goal_continent	sub_on_continent
0.0	6.0	0	0	106.0	28.0	0.0	12.0

sub_out_continent	yellow_card_continent	second_yellow_card_continent	red_card_continent	penalty_goal_continent	conceded_goal_continent	clean_sheet_conti
10.0	10.0	0.0	0.0	11.0	0	

KEPT THROWN CHANGED

VARIABLES KEPT

- Age
- Performance indices in league/national cup/continental cup/nation matches:
  - Goals
  - Assists
  - Own goals
  - Nation selections
- Contract valuation : number of remaining contract years

Each performance index, the number of goals for selection for example, is considering every goal of a player in professional matches from is early career to the 2019’s season included.

THROWN VARIABLES

We decided to separate variables related to field players and goalkeepers because they have barely nothing in common and price calculation is very different. Therefore, we will focus here on field players because they have more explicative variables, and we find it more interesting. We dropped:

- Goalkeeper performance indices in league/national cup/continental cup/nation matches:
  - Conceded goals
  - Clean sheets

Other variables we decided to drop:

- Number of subs in and out: we do not have the number of played match for each player so we cannot compute a ratio of subs/match. Therefore, these variables are not really interesting.
- Variables related to player identification:
  - Name
  - First name

CHANGED VARIABLES

The current team and league of a player is one of the explicative variables used to predict his price. Playing for a big club in a great league makes a player more expensive for example.

Since then, we had to find a way to give the team and league variables a valuation in order to apply the regression algorithm.

We decided to use the UEFA coefficients which ranks:

- Europeans teams depending on their recent performances in continental competitions.
- National leagues depending on how good the teams it gathers have performed in continental competitions.

Following this idea, we decided to apply it to nations by representing them with their FIFA ranking. it classifies nations depending on their performances in world cups and international matches.

Getting these UEFA rankings involved web scrapping on several websites using BeautifulSoup librairies.

CHANGED VARIABLES

The dataset contains the number of yellow cards, second yellow cards and red cards in league/cup/nations matches. This information gives an idea about the behavior of a football player during a match. However, there is not really a difference between being booked in a cup, league or nation match.

Therefore, we decided to sum all the received cards into 2 new variables : yellow cards and red cards. Second yellow cards being already included in red cards, we deleted them.

We also had to find a solution to transform the player position variable into a number. The best one was to use the get_dummies function which creates new columns in the dataset for each position. If the player position matches the column name, the value for this column will be 1, else it is 0. The original position column is then deleted.

After applying regression algorithms, we notice that it worked well with offensive positions having greater coefficients in order to predict the price: it is the case in real world, attackers tend to be the most expensive players as they are more decisive for their teams.

REGRESSION ALGORITHMS

We tried several regression algorithms which are all available on the Jupyter notebook. Our best results in terms of RMSE were obtained with the random forests algorithms applied to regression.

```
# With 4 trees and 6 of depth each
rf_4_6 = RandomForestRegressor(n_estimators=4, max_depth=6)
rf_4_6.fit(X_train, Y_train)
Y_pred_rf_4_6 = rf_4_6.predict(X_test)
display_rmse(Y_pred_rf_4_6)
compare_random(Y_pred_rf_4_6)
```

```
RMSE : 14896859.813532954
Example of prediction
Prediction : 25167752.24434036  True Value : 25000000.0
```

Here is an example of an impressive price prediction we made despite having an enormous RMSE with the algorithm. This RMSE magnitude is expected because we have not normalized the data.

We later made better scores in terms of RMSE with a random forest of 10 trees but struggled to get closer to reality predictions.

AREAS OF IMPROVEMENT

Globally, we are missing regularity in the prediction accuracy with sometime values that are very far from the reality.

Here is what could have been done to give more accurate answers. It is possible with some more time !

Concerning us:

- Learn how to use the grid search
- Discover other regression algorithm to give a better answer to the problem
- Take the time to deeply understand all the parameters than can be tuned

Concerning the database:

- Add the played matches in order to have performance ratios that can give information about a player’s form and his importance for the team
- Having more recent data

CONCLUSION

Working on this subject was enjoyable, interesting and gave us a lot of perspectives for the future.

Obviously, the chosen database was not most complete tool to answer the very complicated problem of predicting football player prices.

It lacked some very important information that we discovered lately such as the number of played matches for example. We also spent a lot of time pre-processing the data which was not usable originally.

However, we have the feeling of having done something nice out of this database. Now, we realize why making this kind of predictions is a full-time job !

If we have the opportunity, we would be glad to continue working on this project in order to refine the database and optimize the algorithms.