Prediction of Footballers' Transfer Values

BjÃ¸rn Skeel-GjÃ¸rling, Chritian Lund SÃ¸rensen, Giullaume Slizewicz & Amer Skaljic

26 August 2016

Introduction

In this paper we want to gather data, describe it and lastly to predict footballers' transfer value by using a variety of methods from the field of data science.Naturally the paper is structured into three sections: Data gathering, data description using vizualization and moddeling. [1] In the first section we briefly describe how we scraped the data from Transfermarkt.co.uk, Wikipedia.org and Google.com what type of data emerged and what ethical challenges we should consider. In the second part the dataset is described using vizualization in order to get an overview of the data and the variables which affect the transfer values. In the final part, we use statistical learning models to predict the transfer values of footballers. The emphasis will be on testing different models and comparing their predictive power on a test set.

Data Gathering

In order to do the analysis, we have to gather data. We gathered the relevant data from the website Transfermarkt.co.uk and Wikipedia.org. Transfermarkt.co.uk contains information on football transfers, player's information and statistics. We find all this kind of information highly relevant in order to predict the transfer value of a player. We used Wikipedia.org in order to find data about the table rangings in the five leagues, which also was available on Transfermarkt.co.uk but we could not succeed in scraping it from the webpage. The data was extracted by two web scrapers, one for Transfermarkt.co.uk and the other for Wikipedia.org. The two scrapers was biuld with the purpose of scraping the relevant data on both webpages. The scraper for Transfermarkt.co.uk ran through all transfers in the transfer windows "Summer 15" and "Winter 16" in the five biggest European leagues: Premier League, Bundesliga, La Liga, Serie A and Ligue 1. It also ran through the transfered players' data. This means that we had two dataframes from Transfermarkt.co.uk at first, which then was merged into one dataframe consisting of players' data. The scraper for Wikipedia.org ran through the different webpages with the ranging tables of the five leagues. At first we had 5 different dataframes, which we merged into one dataframe. The two different datasets were at first cleaned and then merged into one big dataset containing only the variables we find relevant in predicting

[1] All calculations, graphics and writing was conducted using the programming language R and the IDE RStudio.

the transfer values of the players. We do not consider any ethical issues scrapping this data, because all the information is publicly available and not private in any way.

Data Cleaning

Cleaning The Player Dataset

As written earlier before merging the different dataframes into one final, we had to clean both the dataframe with the player data and the dataframe with the club data. Cleaning the player data, we first had to make all unavailable data into NA in stead of different signs. We also had to clean the transfer date and put it into the right format, which we also did with the variable containing remaining time under contract. Some of the data scraped from Transfermarkt.co.uk had some mistakes, which we had to fix by putting them to be equal to NA. We calculated the age of each player at the transferdate by first cleaning the birth date of the players and afterwards calculating the transferage in R by our selves. At last we divided the players into three different categories: Defender, Midfield and Attacker because we found it more appropriate in order to do our analysis. If we did not divide the players into these categories, we would have a dataset with a lot of different type of players, which could be categorized as we did. This categorization means that it is possible to do a reasonable vizualizaton considering that we "only" have 12 pages to write our project. We removed all duplicated observations, observations where transfer.fee was NA and the keepers. The reason for removing the keepers was because we scraped data on mostly offensive statistics, which obviously does not have an affect on the keepers' transfer values. Also the number of variables concerning the keepers' was different compared to the other three types of players, which meant that the observations did not fit into the right column for the keepers.

Cleaning The Club Dataset

Cleaning the club dataset we started with grouping the different clubs in 3 categories: Top Club, Middle Club and Bottom Club. Afterwards we renamed the clubs, so they matched the player datasetm which meant that we could merge the player dataset and the club dataset by using leftjoin. At last, before merging, we selected the variables that we found interesting and removed all other variables in our final cleaned club dataset.

The Final Dataset

The clean and merged data set contains the following descibtive variables of the transfered players:

Variable Name	Describtion
name	Name of player

Variable Name	Describtion
nationality	Nationality of player
birth_place	Birth place of player
birth.date	The date at which the player was born
transferage	Age of player when transfer occured

The clean and merged data set contains the following statistial variables of players:

Variable Name	Describtion
positions	Position on field of player
total.goals	Total amount of goals scored by player
penaltygoals	Total amount of penalty goals scored by player
total.assists	Total amount of assists made by player
substitutions_in	Total amount of matches where player gets substituted into the match
substitutions_out	Total amount of matches where player gets substituted out of the match
total.minutes.played	Total amount of minutes played by player
minutes.pr.goal	Amount of minutes played per goal scored by player

Variable Name	Describtion
yellowcards	Total amount of yellow cards the player got
secondyellow	Total amount of second yellow cards the player got
redcards	Total amount of red cards the player got
contract.left.month	How many month left of the transfered players' contract when transfer occurs

All the statistics above are from the season before the transfer happend, which means the season 2014/2015.

The clean and merged data set contains the following descibtive variables of clubs and transfers:

Variable name	Describtion
club.to	Which club player is transfered to
club.from	Which club player is transfered from
league	The league at which the buying club is playing
status	Status of the buying club, which is divided in "Top Club", "Middle Club" and "Bottom Club"
transfer.date	Which date the transfer happened

Variable name	Describtion
transfer.fee	The transfer fee in million of pounds

All the variables above, we think, are relevant in order to make the best prediction possible of a players transfer value.

Description of Data

Prediction models for transfervalue

Prediction versus causality

The main purpose of this rapport is to find the best way of predicting the transfervalue of footballers. We therefore focus on finding the best way to estimate our dependent variable (Y) when we know the value of several different indendent variables (X's) - hereafter called predictors. The issue of prediction stand in contrast to the regular goal in social sciences where we want to estimate the causal effect of one particular independent variable on the dependent variable. The key difference is that we don't care that much about the individual effect of a predictor (effects of causes) but instead focus the interplay between several predictors and how this interplay can help us predict the value of Y most efficient. Due to this scientific purpose we use the predictors as signals of a specific transfervalue and don't care about whether they come before Y in time or whether there is omitted variable bias (?). A good example of the above mentioned is our use of google searches as a predictor for the players transfervalue. The amount of searches are recorded in real time and therefore after the transfer was made. The number of google searches can not be the reason why the club bought the particular player for a given price but a high number of searches can signal that the players has had a succesful career or was part of a news generating transfer.

Evaluating prediction models using RMSE

To find the best way of predicting a football players transfervalue we will in the following make a comparative analysis of five different models where we compare the prediction accuracy of each model. The models we will use to predict the transfervalues are 1) simple average, 2) ordinary least square (linear model), 3) lasso model, 4) decision tree and 5) random forest.

We measure the models' prediction accuracy by finding the *out of sample error* (the prediction error when applying the models on a dataset on which the model was not trained). The first step of our comparative analysis is therefore to randomly split our data into a *training sample* and a *test sample*. We choose to put 70 percent of the observations in the training sample because we want a sufficiently large amount of observations to train our models.

We evaluate and compare the models by using the root mean squared error (RMSE) which is stated in the equation below. The RMSE gives us a measure for how well the model predict the observations in the test sample. (We use the RMSE instead of the more basic mean square error (MSE) because RMSE reduces the influnce of outliers.)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$$

A basic trade-off to take into account when dealing with prediction is the *bias-variance trade-off*. The reason is that the prediction errors is the sum of errors due to bias and errors due to variance between the training and test sample. When we build our prediction models we do it on the availiable data in the training sample which also include standard noise. If we on the one hand build a model with only a few variables the model will not be able to comprehent the complexity of the real world. Due to this *underfitting* we therefore end up with a biased model whos estimates are quite far from the real world values. If we on the other hand increase the number of variables and thereby the models complexity alot then the models will adjust to much to the training sample. The result of this *overfitting* will be prediction errors due to the variance between the training sample and the real world. The take-away point is therefore that the most accurate models have found a balance between over- and underfitting so that the increase in bias is equivalent to the decrease in variance. The trade-off is illustrated below:

Find graph here: http://scott.fortmann-roe.com/docs/BiasVariance.html

Simple Average