Putting a price tag on football players

Bjørn Skeel-Gjørling, Chritian Lund Sørensen, Giullaume Slizewicz & Amer Skaljic

26 August 2016

1 Introduction

The purpose of this papers is to predict the transfervalue of football players in the five major European leagues.[1] We use a variety of methods from the field of data science to gather, descripe and lastly predict footballers' transfer value. The paper is structured in three sections: Data gathering, data description using different visualizations and prediction modelling.[2] In the first section we briefly describe how we scraped the data from different webpages, what type of data emerged and what ethical challenges we should consider. In the second section the dataset is described using vizualization in order to get an overview of the data and the variables which seems to affect the transfer values. In the final setion, 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. The paper concludes that the random forest model is slightly better at predicting the footballers' transfer value in our test sample compared to rest of the presented models in the paper.

2 Data Gathering

We have to gather data in order to do our analysis. We gathered the relevant data from the websites Transfermarkt.co.uk and Wikipedia.org. Transfermarkt.co.uk contains information on football transfers, player information and performance statistics. We expect all this

[1] Premier League (England), Bundesliga (Germany), La Liga (Spain), Serie A (Italy) and Ligue 1 (France).

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

information to be highly relevant in order to predict the transfer value of a player. We used Wikipedia.org in order to find data about the final table ranking for the five leagues in the season before the transfers (season 14/15). To extract the relevant data we build three web scrapers. One for Transfermarkt, another for Google.com and a third for Wikipedia.org. The scraper for Transfermarkt.co.uk both collected all transfer information from the transfer windows "Summer 15" and "Winter 16" in the five leagues and the individual performance statistics from each transfered player. We then used the Google-scraper to gahter information on the number of google hits when searching for the name of each transfered player followed by the word "footballer". Information on transfer, performance statistics and google hits were hereafter merged into one data frame consisting of player data.

The scraper for Wikipedia ran through the different webpages with the table ranking for the season 14/15 in the five leagues. The table rankings were hereafter merged into one data frame containing club data. The two different datasets were at first cleaned and then merged into one big dataset containing only the variables we find relevant in predicting the transfer values of the players. We do not find any ethical issues scrapping the data from Google and Wikipedia because all the information is publicly available and not private in any way. On the other hand it can be argued that the performance and transfer data is the core of Transfermarkt's business model and they therefore has the right to privacy. We would therefore link to stress that we only use the data for scientific purpose and not a commercial purpose.

Figure 1 below illustrated the data gathering process.

2.1 Data Cleaning

We had to clean both the data frame with player data and the data frame club before merging them into one final data set.

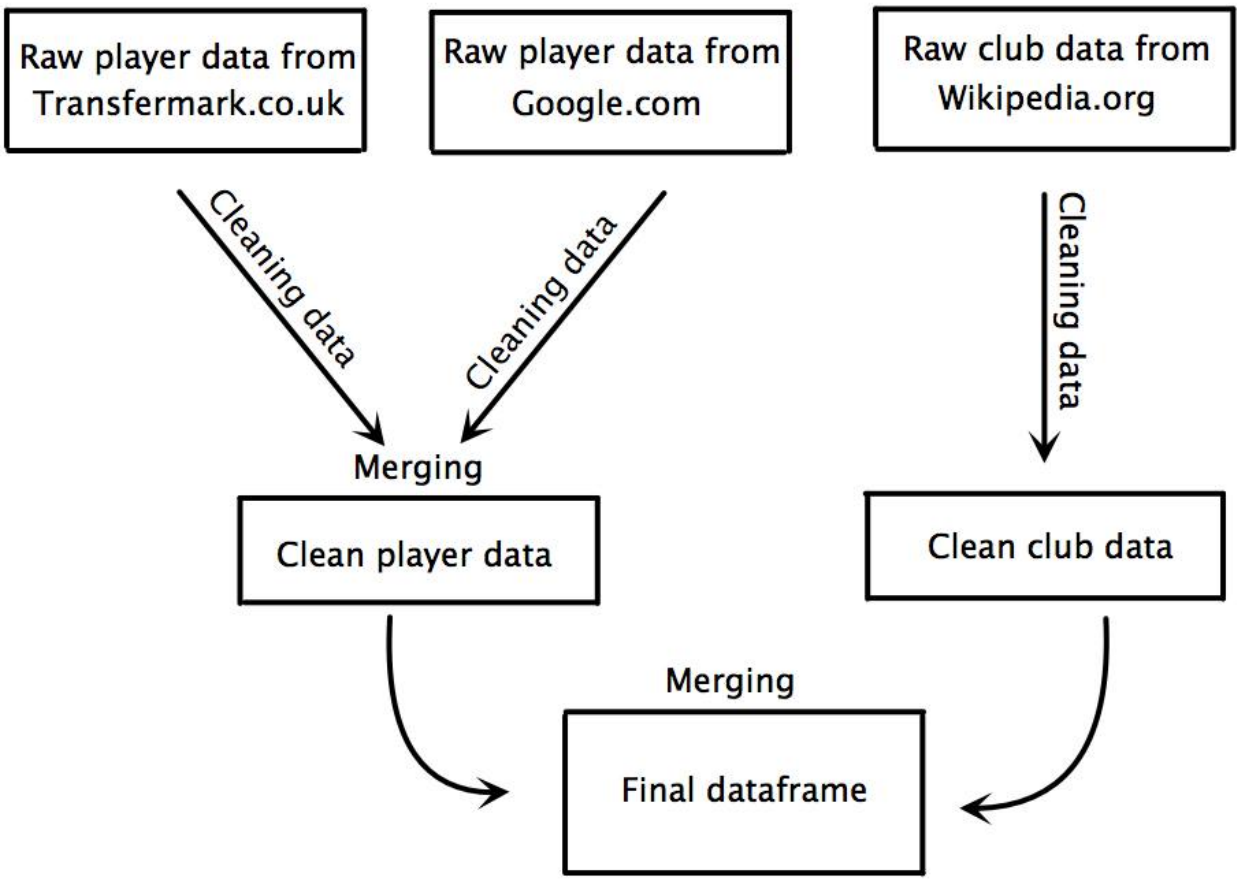


Figure 1: Data gathering process

2.1.1 Cleaning The Player Dataset

For the player data we first turn all the unavailable data into NA instead of different signs. Furthermore, we clean the transfer date, end date of contract and birth date and turned them into the correct date-format in R. The reason for during this was to be able to calculate the precise age (in years) and contract period left (in months) at the day of transfer. We afterwards removed all keepers, duplicated observations and observations for which the transfer fee is unknown. The reason for removing the keepers is that our scraped performance data is mostly offensive statistics, which obviously don't descripe the performance of goalkeepers and thereby affect their transfer value. At last we divided the players into three different categories: Defender, Midfield and Attacker. The reason for doing this categorization is to have are more simple position variable with fewer categorizes. The simplification enable more intuitive visualizations and predictions.

2.1.2 Cleaning The Club Dataset

For the club dataset we started by grouping the different clubs in 4 categories: Top Club (the five highest ranking clubs in the season 14/15), Middle Club (the following ten clubs ranking from six to sixteen), Bottom Club (the remaining clubs in the league that season) and Promoted Club (clubs that entered the league in season 15/16). Afterwards we renamed the clubs so they matched the player dataset, which enabled us to merge the player data set and the club data set using leftjoin. Lastly, we selected the variables that we find most interesting for the later prediction (club name, club status, league) and removed all other variables in our final cleaned club dataset.

2.1.3 The Final Dataset

After the cleaning process we merged the player data set and the club data set into one single data set. The final and tidy dataset contrains 696 observations and 25 variables. A list of the each variable with a short description is avaiable in the appendix. As stated, the final data set contains alot of variables. We will therefore in the following section use different

Table 1: Variables in the data set

Variables	Description
name	Name of player
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 occurred
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 in
substitutions_out	Total amount of matches where player gets substituted out
total.minutes.played	Total amount of minutes played by player
minutes.pr.goal	Amount of minutes played per goal scored by player
yellow cards	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	Months left of the transfered players' contract at transfer
club.to	Which club player is transferred to
club.from	Which club player is transferred from
league	The league that buying club is playing in
Status	Status of the buying club
transfer.data	Date of transfer
transfer.fee	The transfer fee measured in million £
searchresults	Number of search results when you google the player name and 'footballer'

visualizations to select the variables that seems most suited for predicting a football players' transfer values.

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

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

3 Description of Data

3.1 Age/Transfers Graph

In this graph we see that most of the observations actually are in between 0 and 20 million pounds accross all ages. But in the age spectre from 22-28 we see that there are more expensive transfers made. We for example see that the most expensive transfer is for a player that it just below 25 years old, and we also see that in the age spectre emphasized above we see that there are most transfers in the 20-40 million pounds layer. Not surprisingly the number of expensive transfers decreases as the age increases. We can also see that there is only 1 transfer in the 20-40 million pounds layer as the age is 30 and none of these transfers for players 30+ years old. We see the same for young players. Transfers between 20-40 million pounds at first occur as the players gets about 22 years old. The graph also shows that there are a couple younger players that are more expensive than most of the young players, but these players must be very big talents and therefore are something special.

3.2 Time left on contract/Transfer fees

In this graph there is a clear correlation between the time left on a players contract and the transfer fee payed. The longer time there is left of the contract at the time of the transfer, the more expensive the player will be. When the period left of the contract is below 20 months, then the transfer fee does not exceed 30 million pounds. But if we look at when the period left of the contract is above 20 months we see that the transfer fees increases to above 30 million pounds - some transfer fees even increases to above 40 million pounds, especially when the contract is above 40 months. We actually also see that the transfer fees again decreases as the period of the contract left is above 50 months. At last the graph shows that the number of transfers decreases when the period of the contract is above 50 months. This is because the transfer fee demanded by the club selling is to high, and therefore the transfer won't be completed.

3.3 Spending across national leagues

The chart shows the spending on transfers in the 5 major leagues in Europe. We see that the spendings on average are lowest in the French Ligue 1, while the German Bundesliga, the Spanish La Liga and the Italian Serie A all have bigger spendings on average than Ligue 1. We also see that the English Premier League dominates the chart with the highest amount of money spent on average. It exceeds the second most spending league, Serie A, by a bit less than three times as much, spending as much as 60 BILLION POUNDS?!? We imagine that the reason for this among others is the TV-rights agreement in Premier League, which is more lucrativ than in the other 4 leagues.

3.4 Spending by clubs across national leagues

Here we somehow see the same picture as in the former graph. Premier League clubs are spending most money on transfers on average compared to the clubs in the other leagues. But we see that the Spanish teams actually is closer to spending as much as the Premier League clubs compared to the former graph which showed that the difference between the two leagues is huge. This can be explained by the top teams in Spain: Real Madrid, FC Barcelona and Atlético Madrid. Those three clubs, especially the two first, spends a lot of money on transfers and tempts to have several transfer records.

3.5 Status of clubs/spending on transfers

Accross the five leagues we see that the top clubs are spending near double as much money on transfers compared to the middle clubs in the leagues. If we for example take a look at the Spanish top clubs Real Madrid and FC Barcelona, they are spending a lot of money every transfer window. Real Madrid even at the time of these transfers had the most expensive and second-most expensive player on their roster. This graph confirms that it is the case that the top clubs are spending most money. This is also correlated with their success. The more successful the club is in sport, the more money it earns and therefore has more money to spend on transfers. We also see that the promoted clubs actually are spending a bit more

money than the bottom clubs. This is perhaps expected because they come with the intention to improve their roster in order to be able to establish themselves in the major league. Some of the bottom clubs are relegated, which means that they loose money and therefore do not have as much to spend. They perhaps also think that their roster is good enough to compare in the second best league and therefore has no big need for spending on transfers.

4 Prediction models for transfervalue

4.1 Prediction versus causality

The main purpose of this section is to find the best way of predicting transfervalues. 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 hits as a predictor for the players transfervalue. The amount of hits are recorded in real time and therefore after the transfer was made. Due to the time order the number of google hits can not be the reason why the club bought the particular player for a given price. On the other hand a high number of google hits can signal that the players has had a succesful career or was part of a news generating transfer.
The predictors used in this section for estimate a players' transfer value are listed below:

4.2 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 created a function in R to make the RMSE calculation for us. We use the RMSE instead of the more basic mean square error (MSE) because RMSE is more robust to outliers.

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

The *bias-variance trade-off* is important to keep in mind when dealing with prediction. The reason is that the prediction errors are 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 who's estimates are quite far from the real world values. If we on the other hand increase the number of variables and thereby the model's complexity alot then the model 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:

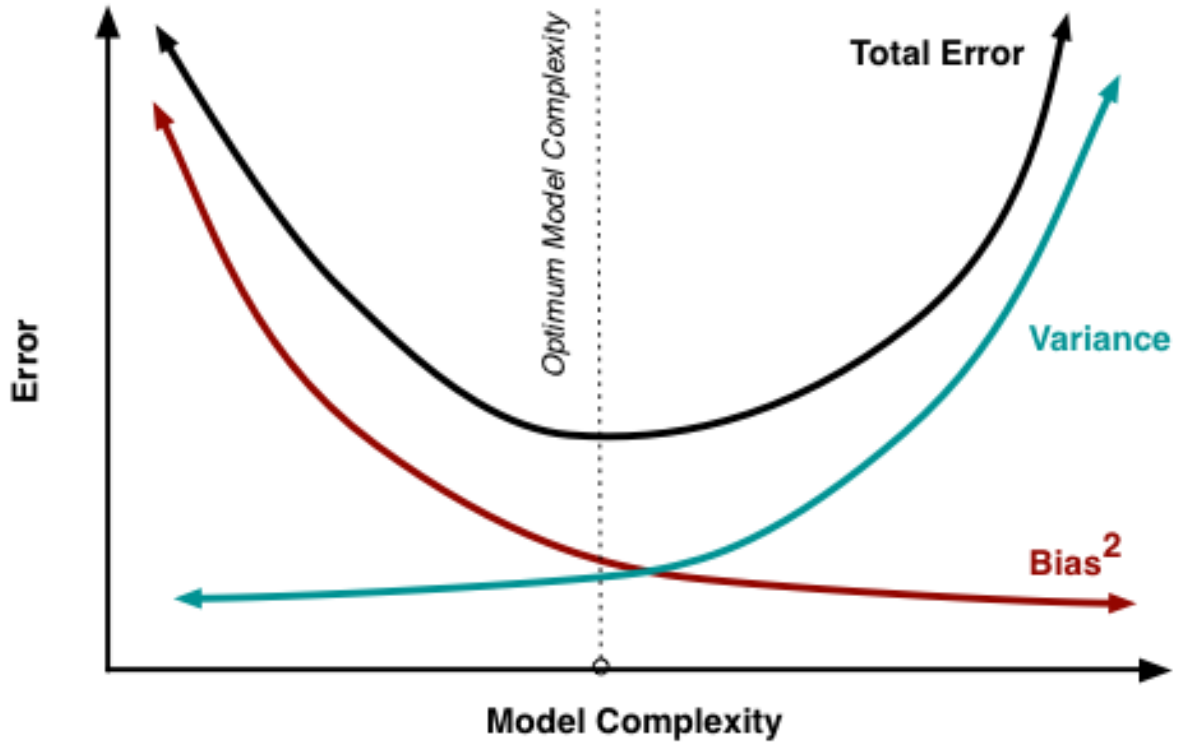


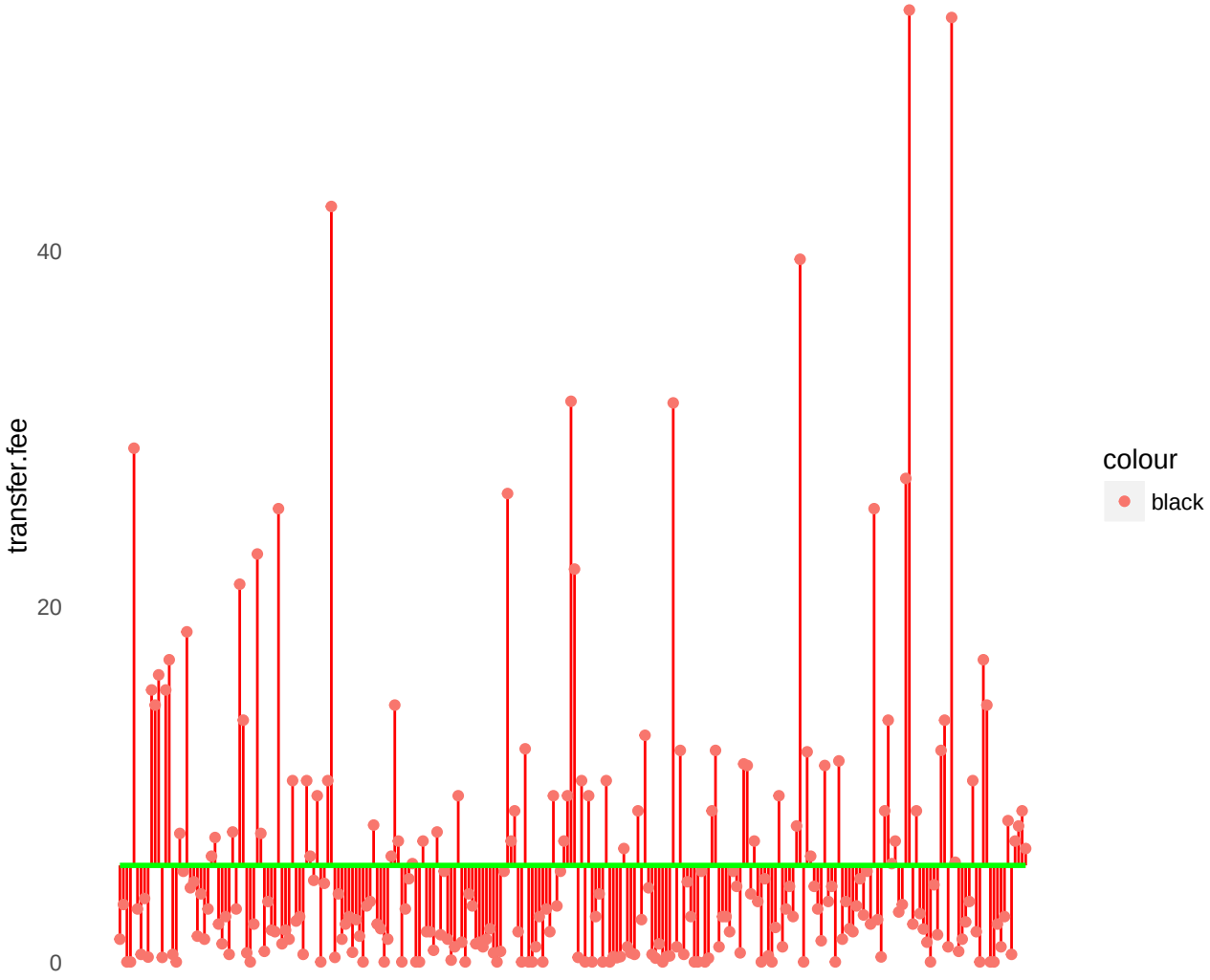
Figure 2: Image is from Roe 2016

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

4.3 Simple Average

As the first prediction model we use the simple transfer fee average. This is a good baseline model because it is the most simple way of estimating a player's transfer value without using any other variables as predictors. When we use the mean transfer fee in the training sample on the test sample we get a RMSE at 9.06.

Because the simple average is not using any predictors for the estimation we can not expect that it will do a very good job at predicting the player's individual transfer value. The high amount of prediction errors is illustrated below:



4.4 Ordinary least square

As the second prediction model we use the linear regression model - ordinary least square (OLS). OLS works by minimizing the *sum of squared residuals* (SSR) and is thereby also minimizing the bias. Unfortunately, the goal of minimizing the in-sample errors lead to a risk of overfitting and thereby increase the errors due to variance. Therefore, OLS is often not the best prediction methods. On the other hand, OLS is a very good method when inference is the goal because the model's significance can be tested. When we apply our OLS model on the test sample we receive a RMSE of 6.34, which is much lower than the RMSE for the simple average.

4.5 The Lasso model

One way to deal with the problem of overfitting is the Lasso model. The Lasso model punishes complexity by adding a loss function to the OLS-equation:

$$\underset{\beta_j}{\text{minimize}} : \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{i=1}^{p}\beta_j x_{ij}^2) + \lambda \sum_{j=1}^{p}|\beta_j|$$

From the equation above we see a clear trade-off between minimizing the SSR and the penalty term. The penalty is given by the sum of the absolute β coefficients. For a given λ the model returns a corner solution of the most significant variables. The λ parameter weights the penalty according to the complexity of the model. The larger λ, the heavier a penalty which lead to exclusion of more variables. The Lasso is performing so-called *variable selection*. The optimal size of λ is decided by running the regression on our training data for different values of λ. For each of the estimated models we estimate the expected transfer values in the test data. We then calculate the RMSE for all the different models and find the λ which minimizes the RMSE in our test data. The model with the optimal λ has a RMSE of 6.34. Therefore, overfitting doesn't seem to be an issue for our regular OLS estimate since the weighting of the penalty term is very low ($\alpha = 0.0009$) and the Lasso gives a slightly higher out of sample error compared to the regular OLS (the RMSE values in the paper are rounded).

4.6 Regression Trees

Decision trees is a machine learning method which can be applied on both categorical and continuous variables. When used for predicting continuous outcomes (such as transfer value) the methods is called *regression trees*. The decision tree is grown by using our training data in the following way: First the predictor space (set of all the possible values for our explanatory variables) is divided into J different regions. For all observations in one region the prediction is given by the mean of the observed dependent variable. The regions are constructed in a way that minimizes the SSR:

$$\sum_{j=1}^{J}\sum_{i \in R_j}(y_i - \hat{y}_{R_m})^2$$

It is computationally infeasible to consider all possible partitions of the feature spaces into J regions. Therefore an a approach called *recursive binary splitting* is used (*An Introduction to Statistical Learning*). In our case the first split is searchresults>50300 which means that this division of the observations provides the greatest reduction of the SSR of all possible first splits across all predictors and all values.

There is a risk that this procedure will lead to overfitting. To avoid this we can use a procedure called pruning. To prune the tree we use the following equation:

$$\sum_{m=1}^{|T|} \sum_{x_i \in R_j} (y_i - \hat{y}_{R_j})^2 + \alpha |T|$$

$|T|$ is the number of terminal nodes in the subtree T, R_m is the subspace of the region M, and y_{R_m} is the predicted response associated with R_m. When pruning the tree we obtain a sequence of the best subtrees as a function of α. We are then using *K-fold cross-validation* (that is we are dividing the training data into K folds) to choose the right α value. A new tree is grown on all but the kth fold of the training data. The process is repeated for each value of α. All the grown trees are evaluated by calculating the MSE on the kth fold. The α value with the lowest average MSE across the folds are choosen and then used on the original tree. As with the Lasso model overfitting wasn't an issue and the original tree produced a lower RMSE than the pruned one. The RMSE was 6.60.

The figure illustrates the regression tree made to predict the transfer value. The first node assign observations with less than 50300 search results to the left branch. The next node of the left branch assign players with less than 31.885 month of the contract left to the left branch and so on. In our model the tree segments the players into 11 leaves and predicts a transfer fee for each of these subgroups. The variables at the top of the tree is the most important ones and the lenght of a branch shows the relative importance of the split. Hence, in our model search results is the most important predictor of a players transfer value.

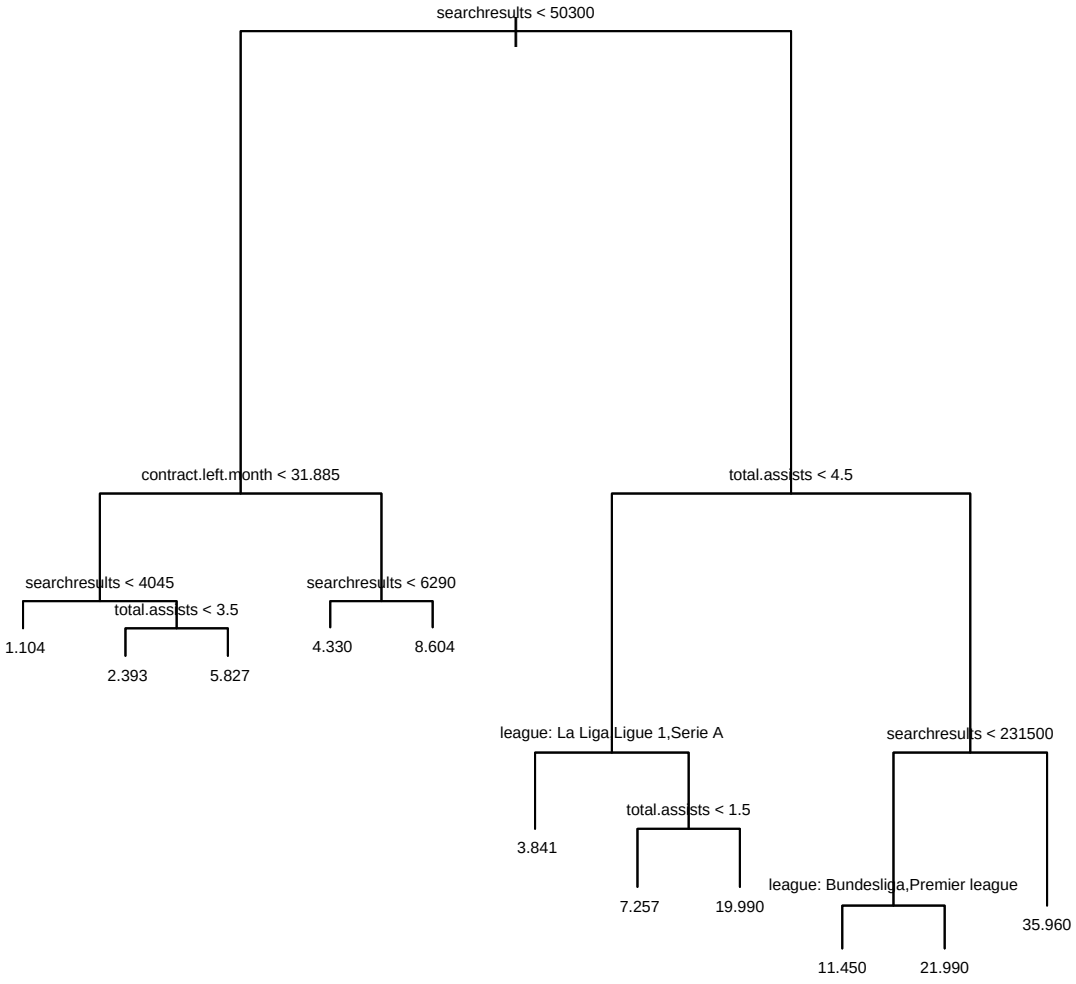


Figure 3: Decision Tree

4.7 Random Forest

As described above the decision tree has issues with overfitting which create errors due to variance. The issues can be reduced by another prediction method called *random forest* (RF). The RF is a socalled ensemple model which means that the model consist of several smaller decision trees (in this case regression trees). RF has great simalarities with the concept of *bagging* or *bottstrap aggregating* (Breiman, 1996) where the core idea is to reduce the error due to variance from one model by building several models and use their average prediction. A random forest prediction can by divided into 4 steps:
1) First, each regression tree is constructed by randomly draw and replace (called bootstrap sampling) 63.2 percent of the observations in the training sample. This is in our rapport done 500 (?) times so we in the end have a forest of 500 different trees with random selected observations from the traning sample.
2) In the next step each tree randomly select m number of the predictor variables in the data set. For regression trees likes these m is equal to the total number of all predictors divided by 3. All 500 trees use the same number of variables. The purpose of randomly choose predictors is to prevent very important variables from overshadowing the effect of weaker variables. This often happens because the underlying algorithm search for the split with the largest decrease in the loss function.
3) Thirdly, each tree calculate the out of bag error rate using the remaning 36.8 percent of the data.
4) In the last step the average prediction are calculated out of all predictions from each individual tree.

In theory, RF does not need a seperate test sample to examine the validity of the results. The validity is measured internally by the out of bag error rate. In this rapport we have used a seperate test sample. The reason is that we wanted to calculate the RMSE so we can compare the RF model's predictive power to the previous prediction models.
The RMSE of the RF on the test sample is 6.26.

We can also use the RF to calculate and rank the importance of each predictor in the data set. This is done by calculating *%IncMSE*. The %IncMSE states the percentage increase of

Table 2: Variable importance

Variables	%IncMSE
posistions	0.35
appearances	8.00
total.goals	5.44
total.assists	2.67
contract.left.month	6.32
transferage	4.26
league	2.52
Status	5.67
searchresults	16.21
transferage_sq	4.79

Table 3: Summary RMSE

Model	RMSE
Average	9.06
OLS	6.34
Lasso	6.34
Regression Tree	6.60
Random Forest	6.26

the mean squared error (estimated with the out-of-bag error rate) if the particular variable is permuted (which mean that the values are randomly shuffled). The higher the number the more important is the variable. The %IncMSE of different predictors are listed in the table below. As with the decision tree google searchresults is shown to be the most important variable.

5 Conclusion

(Write conclusion)