Football Transfers project

Stanislav Shtipkov

14 02 2020

Contents

1 Introduction

1.1 Overview

This document presents a Football Transfer fee system which is created on the base of Machine Learning algorithm. It allows predicting football transfer fees based on the current transfer market prices of the players. For a base is used a document which consists of the 250 most expensive transfers per year for an 19 year period of time and 4700 football transfers in total.

1.2 Executive summary

The following algorithm predicts football player's prices from the dataset. We are exploring and exploiting different options to improve our forecast despite the unpredictable nature of the data. Therefore, we would like to see which information already known has the biggest effect on the player's price.

We are evaluating the algorithm based on the RMSE (Root Mean Square Error) which we compute on the train dataset and practice on the validation dataset. Consequently, four models will be developed and we will compare the RMSE of each one of them.

After we reach a model which satisfies us, we will test it on the pre-separated test set.

2 Methods

2.1 Dataset, preparing and division

- We have preloaded dataset Football Transfers 2000 - 2018, which can be downloaded from: https://www.kaggle.com/vardan95ghazaryan/top-250-football-transfers-from-2000-to-2018
- We create two subsets of the data and we are leaving the "test" outside of our tuning and computation model to avoid overfitting. We will separate 20% of the data for that purpose since we would like to have enough information to make a reliable test with the final model. After that, we will divide the working data into train and validation sets to tune the methods.

```
# Download the database
url <- "https://raw.githubusercontent.com/stanshtipkov/FootballTransfers/master/top250-00-19.csv?fbclid=

if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")


dat <- read_csv(url)
download.file(url, "top250-00-19.csv")

data <-read.csv("top250-00-19.csv")

# Separate the test set from the rest of the data
set.seed(1, sample.kind="Rounding")
test_index <- createDataPartition(y = data$Transfer_fee, times = 1, p = 0.2, list = FALSE)
data1 <- data[-test_index,]
test <- data[test_index,]

test <- test %>% semi_join(data1, by="Name")

# Dividing on train and validation sets
set.seed(1, sample.kind="Rounding")
validation_index <- createDataPartition(y = data1$Transfer_fee, times = 1, p = 0.1, list = FALSE)
train <- data1[-validation_index,]
validation <- data1[validation_index,]

validation <- validation %>% semi_join(train, by="Name")
```

2.2 Prework data analyse

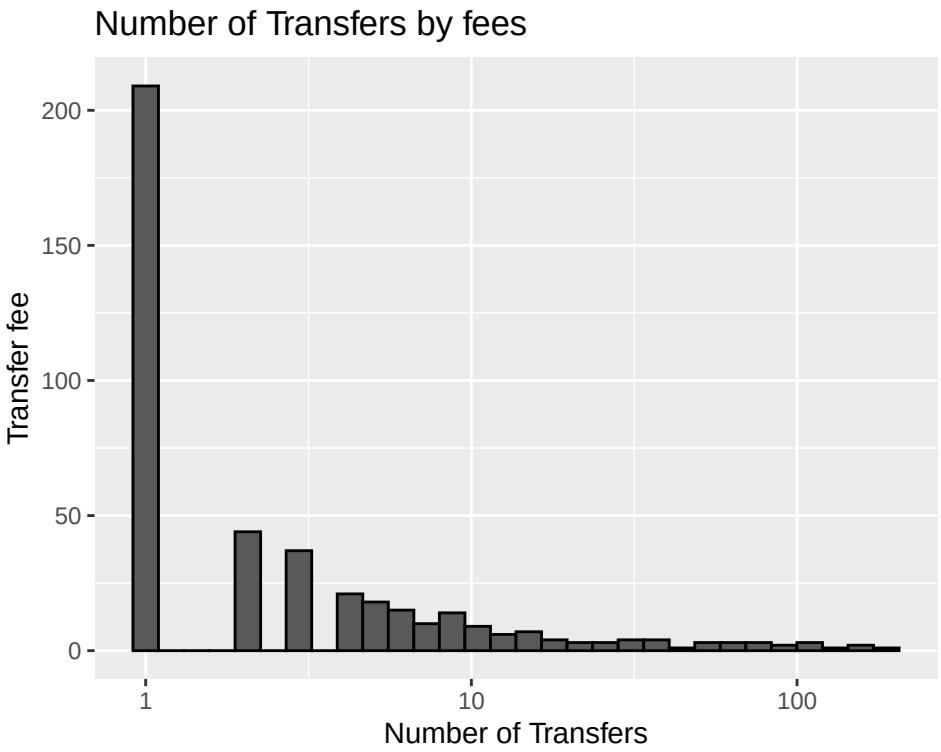
We are exploring and visualizing what train set we have to choose the best possible approach

- Dataset columns

```
##                 Name         Position Age       Team_from    League_from
## 1        Lu s Figo     Right Winger  27    FC Barcelona         LaLiga
## 3     Marc Overmars      Left Winger  27         Arsenal Premier League
## 4 Gabriel Batistuta   Centre-Forward  31      Fiorentina        Serie A
## 5    Nicolas Anelka   Centre-Forward  21     Real Madrid         LaLiga
## 7 Fl vio Conceicao Central Midfield  26 Dep. La Coru ±a         LaLiga
## 8    Savo Milosevic   Centre-Forward  26   Real Zaragoza         LaLiga
##        Team_to League_to    Season Market_value Transfer_fee
## 1  Real Madrid    LaLiga 2000-2001           NA     60000000
## 3 FC Barcelona    LaLiga 2000-2001           NA     40000000
## 4      AS Roma   Serie A 2000-2001           NA     36150000
## 5     Paris SG   Ligue 1 2000-2001           NA     34500000
## 7  Real Madrid    LaLiga 2000-2001           NA     25000000
## 8        Parma   Serie A 2000-2001           NA     25000000
```

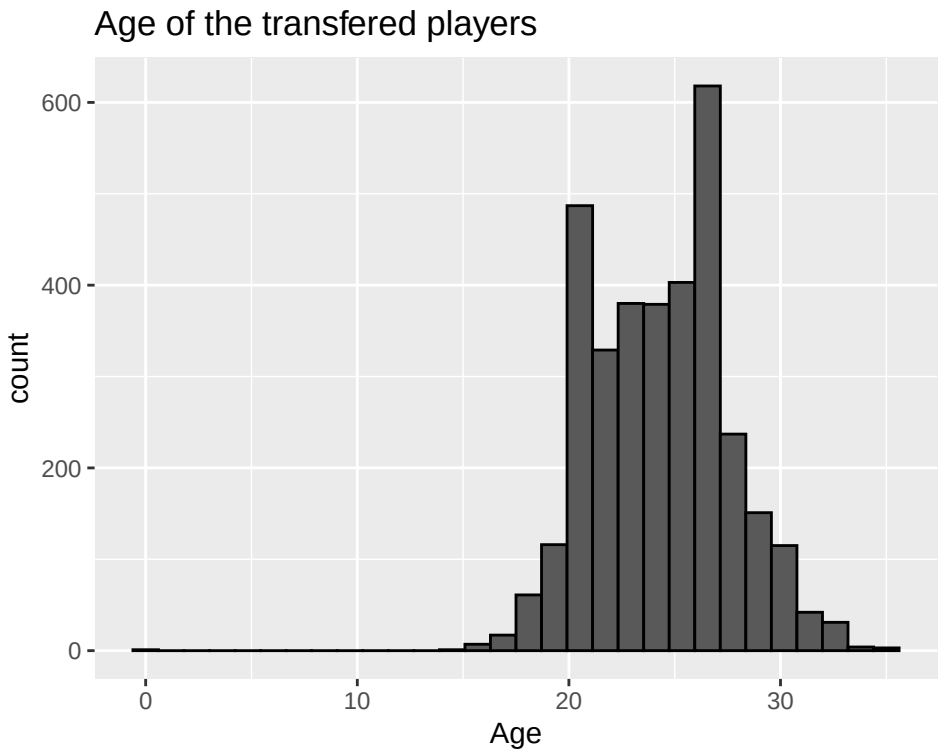
- Number of transfers by the paid fees in the train set:

```
train %>%
 count(Transfer_fee) %>%
 ggplot(aes(n)) +
 geom_histogram(bins=30,color="black")+
 scale_x_log10()+
 xlab("Number of Transfers")+
 ylab("Transfer fee")+
 ggtitle("Number of Transfers by fees")
```



- Transfers by age:

```
train %>%
 ggplot(aes(Age))+
 geom_histogram(bins=30, color="black")+
 ggtitle("Age of the transfered players")
```



2.3 Computation

We will evaluate our models based on the loss function

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$$

Then we will predict with our model how well we did comparing to our "validation" set

```
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

3 Modelling

3.1 Average prediction

Firstly, we are predicting the same fees for all transfers, so we calculate the average of the dataset "train".

```
mu <- mean(train$Transfer_fee)
mu
```

```
## [1] 9429293
```

Therefore if we predict all unknown fees with μ or mu, we obtain the first naive RMSE

```
naive_rmse <- RMSE(validation$Transfer_fee,mu)
naive_rmse
```

```
## [1] 12013164
```

Here, we introduce our results table to collect all the outcomes

```
rmse_results <- tibble(method = "Just the average", RMSE = naive_rmse)
rmse_results %>% knitr::kable()
```

method	RMSE
Just the average	12013164

However, we would like to see if it is possible to reduce this number by applying methods which include some of the variables into the dataset since it could have a relationship to the transfer fees of the players.

3.2 Age model

In our second model, we are accepting that some Age of the players could have an effect on the transfer fee. We compute the estimated deviation of each transfer means fee from the total mean of all transfers μ.

```
age_model <- train %>%
  group_by(Age) %>%
  summarize(b_i = mean(Transfer_fee - mu))
```

Our prediction improved slightly. However, we would like to explore if it is possible to enhance more our prediction.

```
predicted_fee <- mu + validation %>%
  left_join(age_model, by='Age') %>%
  pull(b_i)
model_1_rmse <- RMSE(predicted_fee, validation$Transfer_fee)
rmse_results <- bind_rows(rmse_results,
                          tibble(method="Age Model",
                                 RMSE = model_1_rmse ))
rmse_results %>% knitr::kable()
```

method	RMSE
Just the average	12013164
Age Model	12004417

3.3 Age and Position model

In this model, we are adding the idea that player's position influence its price and we will combine it with the already existing Age model.

```
pos_model<- train %>%
  left_join(age_model, by='Age') %>%
  group_by(Position) %>%
  summarize(b_u = mean(Transfer_fee - mu - b_i))
```

This model has a little effect on our prediction! However, we will see if it is possible to advance even further with an additional tuning parameter.

```
predicted_fee <- validation %>%
  left_join(age_model, by='Age') %>%
  left_join(pos_model, by='Position') %>%
  mutate(pred = mu + b_i + b_u) %>%
  .$pred
model_2_rmse <- RMSE(predicted_fee, validation$Transfer_fee)
rmse_results <- bind_rows(rmse_results,
                          tibble(method="Age + Position Model",
                                 RMSE = model_2_rmse ))
rmse_results %>% knitr::kable()
```

method	RMSE
Just the average	12013164
Age Model	12004417
Age + Position Model	11820783

3.4 Age, Position and League model

To improve our results, we will add the League effect. We will add to our model the Championship of the buying team since we believe that particular leagues are more wealthy and can afford paying bigger money for the best players in the world.

```
leag_model<- train %>%
  left_join(age_model, by='Age') %>%
  left_join(pos_model, by='Position') %>%
  group_by(League_to) %>%
  summarize(b_l = mean(Transfer_fee - mu - b_i - b_u))
```

From the results, we see that our prediction improves significantly. Consequently, we can say that our prediction model worth testing on the "test" set.

```
predicted_fee <- validation %>%
  left_join(age_model, by='Age') %>%
  left_join(pos_model, by='Position') %>%
  left_join(leag_model, by='League_to') %>%
  mutate(pred = mu + b_i + b_u + b_l) %>%
  .$pred
model_3_rmse <- RMSE(predicted_fee, validation$Transfer_fee)
rmse_results <- bind_rows(rmse_results,
                          tibble(method="League + Age + Position Model",
                                 RMSE = model_3_rmse ))
```

```
rmse_results %>% knitr::kable()
```

method	RMSE
Just the average	12013164
Age Model	12004417
Age + Position Model	11820783
League + Age + Position Model	11489888

3.5 Test Check

We are testing our last model "Age, Position and League" on the test set to confirm or reject our findings so far.

```
predicted_fee <- test %>%
  left_join(age_model, by='Age') %>%
  left_join(pos_model, by='Position') %>%
  left_join(leag_model, by='League_to') %>%
  mutate(pred = mu + b_i + b_u + b_l) %>%
  .$pred
test_check <- RMSE(predicted_fee, test$Transfer_fee)
rmse_results <- bind_rows(rmse_results,
                          tibble(method="Test Check",
                                     RMSE = test_check))

rmse_results %>% knitr::kable()
```

method	RMSE
Just the average	12013164
Age Model	12004417
Age + Position Model	11820783
League + Age + Position Model	11489888
Test Check	9900471

We can see that our results are close and even lower at the Test set. This means that we can conclude our algorithm here.

4 Results

After the computation of four different models, we have our results, and the last model tested with the "Test" set.

method	RMSE
Just the average	12013164
Age Model	12004417
Age + Position Model	11820783
League + Age + Position Model	11489888

method	RMSE
Test Check	9900471

5 Conclusion

We can see that our Machine Learning Algorithm creates an environment for better prediction of Transfer fees of football players. However, is our outcome satisfying is another question since I believe that more sophisticated methods could be applied for even better results. Perhaps, in future, a model to predict the eventual price of a player could be developed taking into account much richer data than this which we had on our disposal.

In addition, more detailed database will be useful to improve our results.