Player value estimation on FIFA 23 dataset using linear regression

Linear Regression Model for FIFA23 data set

In this study, we will examine the linear regression model that will predict player values for the FIFA 23 data set, with the following parts.

- Describe data set
- Training linear regression model
- The performance of the regression model
- Checking under-fitting and over-fitting status
- Create new observation

Packages

The libraries used in this study are as follows.

```
library(stringr)
library(ggplot2)
library(gridExtra)
```

```
Warning: package 'gridExtra' was built under R version 4.1.3
```

FIFA23 Data set

The data set contains general information of +17k unique players and features offered by the famous video game. We downloaded the data set from https://www.kaggle.com/datasets/bryanb/fifa-player-stats-database/code?select=FIFA23_official_data.csv and got the data with the command to read csv file in R.

```
FIFAdata <- data.frame(read.csv("FIFA23_official_data.csv"))
```

We clean the data with NA in the dataset

```
FIFAdata<- na.exclude(FIFAdata)
```

We can look at the data set with the **str()** function. The function returns values containing the following information:

- 17625 observation
- 29 variables (features)

  The types and some values of the features are shown in the figure below.

```
str(FIFAdata)
```

```
'data.frame':   17625 obs. of  29 variables:
 $ ID                       : int  209658 212198 224334 192985 224232 212622 197445 187961 208
 $ Name                     : chr  "L. Goretzka" "Bruno Fernandes" "M. AcuÃ±a" "K. De Bruyne"
 $ Age                      : int  27 27 30 31 25 27 30 32 28 28 ...
 $ Photo                    : chr  "https://cdn.sofifa.net/players/209/658/23_60.png" "https:/
 $ Nationality              : chr  "Germany" "Portugal" "Argentina" "Belgium" ...
 $ Flag                     : chr  "https://cdn.sofifa.net/flags/de.png" "https://cdn.sofifa.n
 $ Overall                  : int  87 86 85 91 86 89 86 83 82 88 ...
 $ Potential                : int  88 87 85 91 89 90 86 83 82 88 ...
 $ Club                     : chr  "FC Bayern MÃ¼nchen" "Manchester United" "Sevilla FC" "Manc
 $ Club.Logo                : chr  "https://cdn.sofifa.net/teams/21/30.png" "https://cdn.sofi
 $ Value                    : chr  "â¬91M" "â¬78.5M" "â¬46.5M" "â¬107.5M" ...
 $ Wage                     : chr  "â¬115K" "â¬190K" "â¬46K" "â¬350K" ...
 $ Special                  : int  2312 2305 2303 2303 2296 2283 2277 2273 2271 2262 ...
 $ Preferred.Foot           : chr  "Right" "Right" "Left" "Right" ...
 $ International.Reputation: num  4 3 2 4 3 4 4 3 3 3 ...
 $ Weak.Foot                : num  4 3 3 5 3 4 4 4 4 4 ...
 $ Skill.Moves              : num  3 4 3 4 3 3 3 4 3 4 ...
 $ Work.Rate                : chr  "High/ Medium" "High/ High" "High/ High" "High/ High" ...
 $ Body.Type                : chr  "Unique" "Unique" "Stocky (170-185)" "Unique" ...
 $ Real.Face                : chr  "Yes" "Yes" "No" "Yes" ...
 $ Position                 : chr  "<span class=\"pos pos28\">SUB" "<span class=\"pos pos15\">
 $ Joined                   : chr  "Jul 1, 2018" "Jan 30, 2020" "Sep 14, 2020" "Aug 30, 2015"
 $ Loaned.From              : chr  "nan" "nan" "nan" "nan" ...
 $ Contract.Valid.Until     : chr  "2026" "2026" "2024" "2025" ...
 $ Height                   : chr  "189cm" "179cm" "172cm" "181cm" ...
```

```
$ Weight                  : chr  "82kg" "69kg" "69kg" "70kg" ...
$ Release.Clause          : chr  "â¸¬157M" "â¸¬155M" "â¸¬97.7M" "â¸¬198.9M" ...
$ Kit.Number              : num  8 8 19 17 23 6 4 15 23 7 ...
$ Best.Overall.Rating     : chr  "nan" "nan" "nan" "nan" ...
- attr(*, "na.action")= 'exclude' Named int [1:35] 683 909 1113 1927 2595 5032 5304 5936 63
 ..- attr(*, "names")= chr [1:35] "683" "909" "1113" "1927" ...
```

We get Value, Wage, Age, Potential and Overall variables from the dataset.

```
FIFAdata<-data.frame(
   Value=FIFAdata$Value,
   Wage=FIFAdata$Wage,
   Age=FIFAdata$Age,
   Potential=FIFAdata$Potential,
   Overall=FIFAdata$Overall
)
```

Task: Predict Football Player Values

In this study, we try to estimate the values of the players using the FIFA 23 data set to train the linear regression model.

Step1 - Value and wage feature editing of data set

The *value* and *wage* features in the dataset appear as character types. We need to make these features suitable for linear regression.

```
#Getting rid of the Euro expression in the Value and Wage columns

FIFAdata$Value<-gsub('[â¸¬]', '', FIFAdata$Value)
FIFAdata$Wage<-gsub('[â¸¬]', '', FIFAdata$Wage)


#Converting K and M values in Value variable

for (i in 1:length(FIFAdata$Value))
     {

  if(str_detect(FIFAdata$Value[i],"M")) {
    FIFAdata$Value[i]<-gsub('[M]', '', FIFAdata$Value[i])
```

```
    FIFAdata$Value[i]<-as.numeric(FIFAdata$Value[i])*1000000

  } else if(str_detect(FIFAdata$Value[i],"K")){
    FIFAdata$Value[i]<-gsub('[K]', '', FIFAdata$Value[i])
    FIFAdata$Value[i]<-as.numeric(FIFAdata$Value[i])*1000
  }

}

FIFAdata$Value<-as.numeric(FIFAdata$Value)

#Converting K values in Wage variable
FIFAdata$Wage<-gsub('[K]', '000', FIFAdata$Wage)
FIFAdata$Wage <-as.numeric(FIFAdata$Wage)
```

Step 2- Splitting the data set

We use the *sample()* function to split 80% of the data into the train and 20% into the test set and set the *seed()* to keep the same values for every future run.

```
set.seed(123)
index <-sample(1:nrow(FIFAdata),round(nrow(FIFAdata))*0.80)
traindata <-FIFAdata[index,]
testdata <-FIFAdata[-index,]
```

Here, randomly selected indexes are assigned to train and test sets.

Step 3- Train a linear regression

We use the "*lm()*" function to train a linear regression model. This function takes two parameters. The first is the model formula and the second is the dataset used to train the model. The formula of our model is defined as "y ~ x1 + x2 + x3 + x4". where "**y**" represents the value attributes of the players to be estimated, and "**x1,x2,x3 and x4**" represents the *Overall*, *Wage*, *Age* , *Potentail* variables, respectively. We give the train data to the model as a dataset.

```
lrm_FIFA <-lm(FIFAdata$Value ~ FIFAdata$Overall + FIFAdata$Wage + FIFAdata$Age
+ FIFAdata$Potential  ,data=traindata)
```

The output of our model is assigned to the lrm_FIFA object. The output of this model is:

```
lrm_FIFA
```

```
Call:
lm(formula = FIFAdata$Value ~ FIFAdata$Overall + FIFAdata$Wage +
    FIFAdata$Age + FIFAdata$Potential, data = traindata)

Coefficients:
       (Intercept)    FIFAdata$Overall       FIFAdata$Wage        FIFAdata$Age
        -3897826.2            327333.1               275.8           -419074.3
FIFAdata$Potential
          -94002.1
```

We can see detailed information about the model with the *summary()* function.

```
summary(lrm_FIFA)
```

```
Call:
lm(formula = FIFAdata$Value ~ FIFAdata$Overall + FIFAdata$Wage +
    FIFAdata$Age + FIFAdata$Potential, data = traindata)

Residuals:
      Min        1Q    Median        3Q       Max
-63176953   -897797    -86945    765542 119755548

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        -3.898e+06  5.305e+05  -7.348 2.10e-13 ***
FIFAdata$Overall    3.273e+05  1.216e+04  26.928  < 2e-16 ***
FIFAdata$Wage       2.758e+02  2.014e+00 136.904  < 2e-16 ***
FIFAdata$Age       -4.191e+05  1.448e+04 -28.949  < 2e-16 ***
FIFAdata$Potential -9.400e+04  1.167e+04  -8.056 8.36e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4359000 on 17620 degrees of freedom
Multiple R-squared:  0.6916,    Adjusted R-squared:  0.6915
F-statistic:  9878 on 4 and 17620 DF,  p-value: < 2.2e-16
```

Step 4- Measuring model performance

Checking the performance of the model with the test data set is necessary for the model to have good generalization. To do this, we first calculate the *Value* values in the test set. When calculating this, we need to exclude this feature from the test set.

```
predicted_values <- predict(lrm_FIFA,testdata[,-1])
```

```
head(predicted_values)
```

```
       1         2         3         4         5         6
36704902  57153268  16047858 100858581  35742936  41307904
```

Then we compare the estimated values with the actual values.

```
error<-testdata$Value-predicted_values
head(error)
```

```
       1         2         3         4         5         6
-8204902 -28153268  37452142 -45358581   4257064 -10807904
```

There are 3 main metrics used in regression analysis. These are Mean Squared Error(MSE), Root Mean Squared Error(RMSE) and Median Absolute Error (MAE). We calculate all 3 errors according to our trained model.

```
mse_model<-mean(error^2)
rmse_model <-sqrt(mean(error^2))
mae_model <- median(abs(error))
```

Error values are given below.

```
mse_model
```

```
[1] 8.53622e+13
```

```
rmse_model
```

```
[1] 9239167
```

```
mae_model
```

```
[1] 1701798
```

Checking the over and under-fitting problem

At this stage, the way to check if there are any issues with over- or under-fitting in our model is to compare the model performance on the train and test set. For this, Mean Squaered Error (MSE), Root Mean Squaered Error (RMSE) and Median absolute error (MAE) are used.

In this application we use the RMSE (Root Mean Squared Error) estimation error, which represents the mean difference between the known outcome values observed in the test data and the outcome values predicted by the model. The lower the RMSE value, the better the model.

```
rmsetrain<- sqrt(mean((lrm_FIFA$residuals)^2))
rmsetest<-rmse_model
```

Then let's calculate the difference between the RMSE values.

```
rmsetrain-rmsetest
```

```
[1] -4880864
```

The difference in RMSE values was negative, which means that the test set is better than the train set. The R^2 (0.6915) value on the Train set also showed us this. It may be useful to use more features or cross-validation to increase this value.

Create New Observation

In this section, we will estimate the player value based on the player featues in our own data set.

```
newObservation<-data.frame(

  Overall=c(87,75,90,68,50),
  Wage=c(250000,120000,70000,40000,13000),
  Age=c(21,24,18,26,30),
  Potential=c(91,80,95,74,70)
```

```
)
```

We test our new observations in our model and calculate the player values.

```
predicted_newobs<-predict(lrm_FIFA,newObservation)
```

```
head(predicted_newobs)
```

```
       1          2          3          4          5
36704902  57153268  16047858 100858581  35742936
```

Figures of the relationship between player characteristics and player values are as follows.

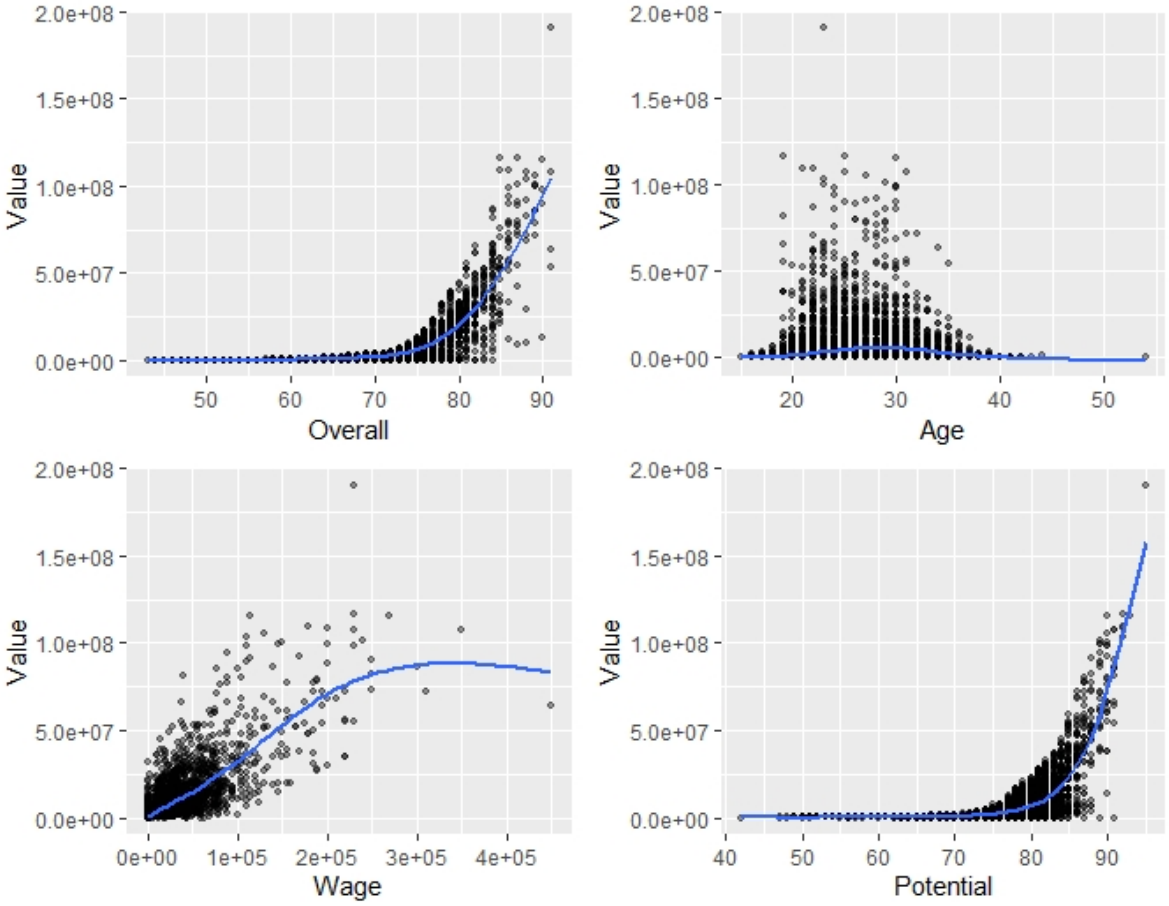
```
plot1<- ggplot(traindata,aes(traindata$Overall,traindata$Value))+
  geom_point(size=1,alpha=.4)+
  geom_smooth(method = "gam", formula = y~s(x,bs="cs") ,se=FALSE)+
  scale_y_continuous("Value")+
  xlab("Overall")

plot2<- ggplot(traindata,aes(traindata$Age,traindata$Value))+
  geom_point(size=1,alpha=.4)+
  geom_smooth(method = "gam", formula = y~s(x,bs="cs") ,se=FALSE)+
  scale_y_continuous("Value")+
  xlab("Age")

plot3<- ggplot(traindata,aes(traindata$Wage,traindata$Value))+
  geom_point(size=1,alpha=.4)+
  geom_smooth(method = "gam", formula = y~s(x,bs="cs") ,se=FALSE)+
  scale_y_continuous("Value")+
  xlab("Wage")

plot4<- ggplot(traindata,aes(traindata$Potential,traindata$Value))+
  geom_point(size=1,alpha=.4)+
  geom_smooth(method = "gam", formula = y~s(x,bs="cs") ,se=FALSE)+
  scale_y_continuous("Value")+
  xlab("Potential")
```

```
gridExtra::grid.arrange(plot1,plot2,plot3,plot4, nrow=2)
```


Value
2.0e+08
1.5e+08
1.0e+08
5.0e+07
0.0e+00
50
60
70
80
90
Overall
20
30
40
50
Age
0e+00
1e+05
2e+05
3e+05
4e+05
Wage
40
50
60
70
80
90
Potential