Regression Project

Amirali Khatib

7/3/2022

Introduction

In the this classification and regression project, we'll work with `players_csv` file which consist of detailed attribute for each football player. we're following 2 purpose in this project.

Regression goal: We fit regression models to our data in order to predict players' value in the transfer market. So, football clubs' owners will know how much they should pay to sign a contract with a player that has a special combination of skills.

Classification goal: Our purpose in fitting classification models to our data is to build a high performance classifier that be able to recommend the head coach the squad position for each player with specific sort of abilities.

Dataset features:

To know some more detailes and explanation about the features that exist in the data set, I recommend you to check these two link bellow:

- Attributes expalation
- Example : Luka Modric attributes

Intialization

Import required libraries

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Warning: package 'tibble' was built under R version 4.1.3
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
## Warning: package 'forcats' was built under R version 4.1.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.1.3
```

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.1.3
```

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.1.3
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.1.3
```

```
library(nnet)
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.1.3
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(ellipse)
```

```
## Warning: package 'ellipse' was built under R version 4.1.3
```

```
##
## Attaching package: 'ellipse'
```

```
## The following object is masked from 'package:graphics':
##
##     pairs
```

```
library(lares)
```

```
## Warning: package 'lares' was built under R version 4.1.3
```

```
##
## Attaching package: 'lares'
```

```
## The following object is masked from 'package:e1071':
##
##     impute
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.1.3
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(ROSE)
```

```
## Warning: package 'ROSE' was built under R version 4.1.3
```

```
## Loaded ROSE 0.0-4
```

Read dataset

```
fifa = read.csv(file = 'D:\\Amirali\\University\\Applied Statistical Analysis\\R-tutorial\\Project\\players_22.cs
v', header = TRUE)
```

Data pre-processing

```
fifa = na.omit(fifa)
fifa = fifa %>%
  dplyr::select(-position, everything()) %>%
  dplyr::select(-value_eur, everything())
fifa1.0_reg = fifa
fifa1.0_reg[,c("sofifa_id","short_name","potential","club_name","league_name","club_jersey_number","nationality_n
ame","weight_kg", "height_cm")] <- list(NULL)
colnames(fifa1.0_reg)
```

```
##  [1] "overall"                  "potential_growth"
##  [3] "age"                      "international_reputation"
##  [5] "crossing"                 "finishing"
##  [7] "heading_accuracy"         "short_passing"
##  [9] "volleys"                  "dribbling"
## [11] "curve"                    "long_passing"
## [13] "ball_control"             "sprint_speed"
## [15] "agility"                  "reactions"
## [17] "shot_power"               "jumping"
## [19] "stamina"                  "strength"
## [21] "long_shots"               "aggression"
## [23] "interceptions"            "positioning"
## [25] "vision"                   "composure"
## [27] "marking_awareness"        "standing_tackle"
## [29] "sliding_tackle"           "gk_diving"
## [31] "gk_handling"              "gk_kicking"
## [33] "gk_positioning"           "gk_reflexes"
## [35] "position"                 "value_eur"
```

Data spliting

```
set.seed(23)
# test-train split
idx = sample(nrow(fifa1.0_reg), size = 0.8 * nrow(fifa1.0_reg))
fifa_trn = fifa1.0_reg[idx,]
fifa_tst = fifa1.0_reg[-idx,]

# estimation-validation split
idx1 = sample(nrow(fifa_trn), size = 0.8 * nrow(fifa_trn))
fifa_est = fifa_trn[idx1,]
fifa_val = fifa_trn[-idx1,]

# check data
length(fifa1.0_reg)
```

```
## [1] 36
```

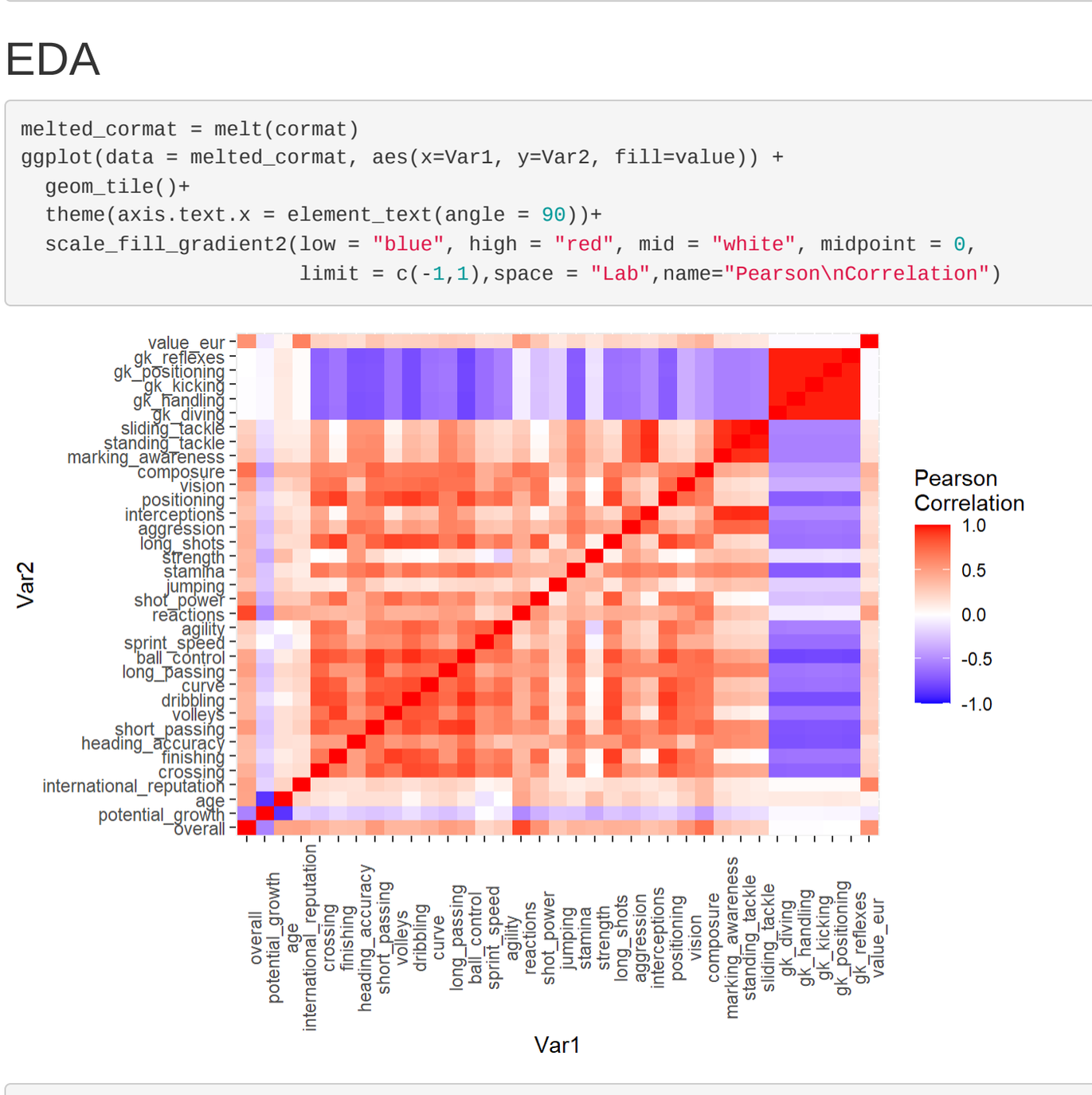
```
unique(fifa$position)
```

```
## [1] "forward"    "midfielder" "goalkeeper" "defender"
```

```
fifa_est_c = fifa_est
fifa_est_c[,c('position')] <- list(NULL)
cormat = cor(fifa_est_c)
cormat = round(cormat,2)
```

EDA

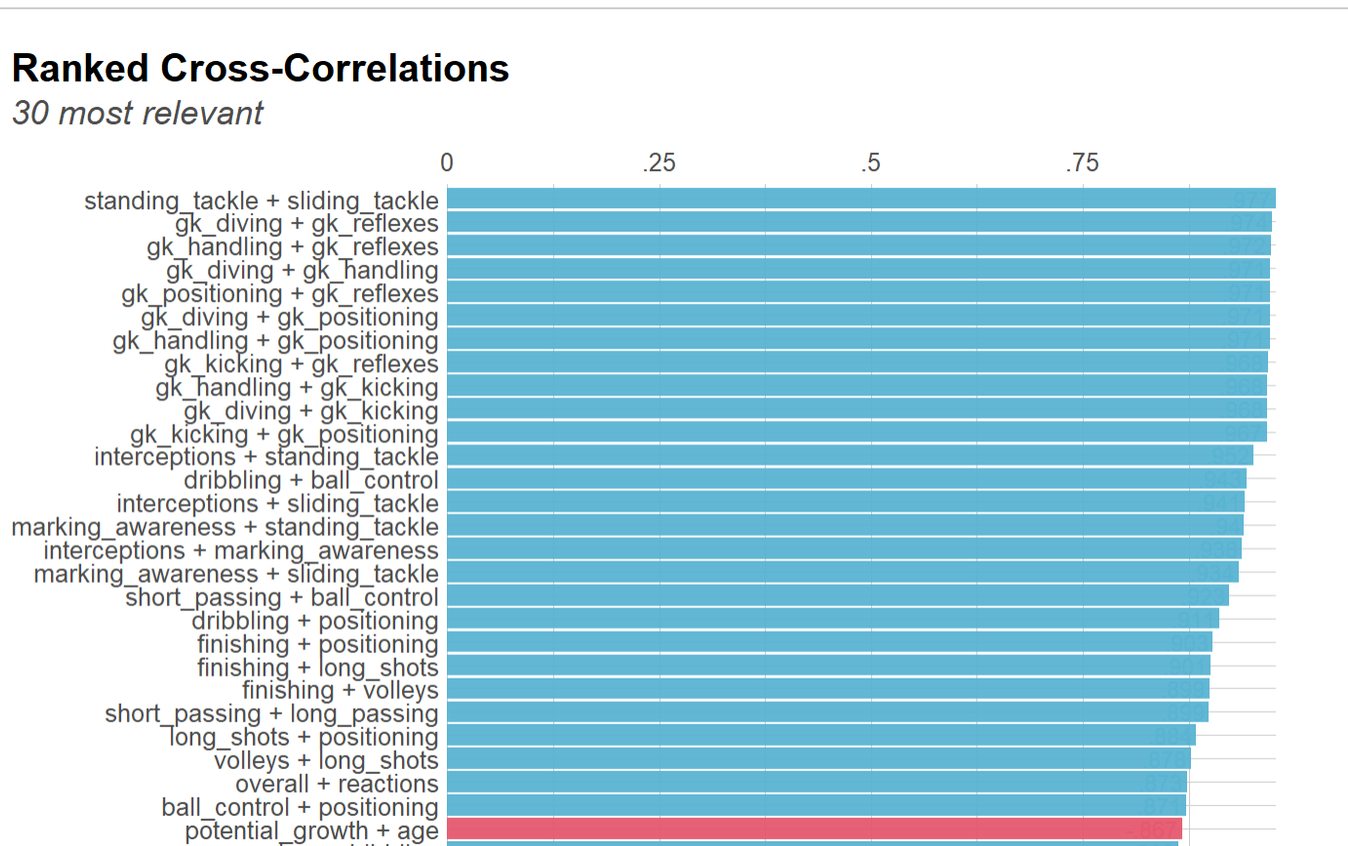
```
melted_cormat = melt(cormat)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()+
  theme(axis.text.x = element_text(angle = 90))+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,
                       limit = c(-1,1),space = "Lab",name="Pearson\nCorrelation")
```



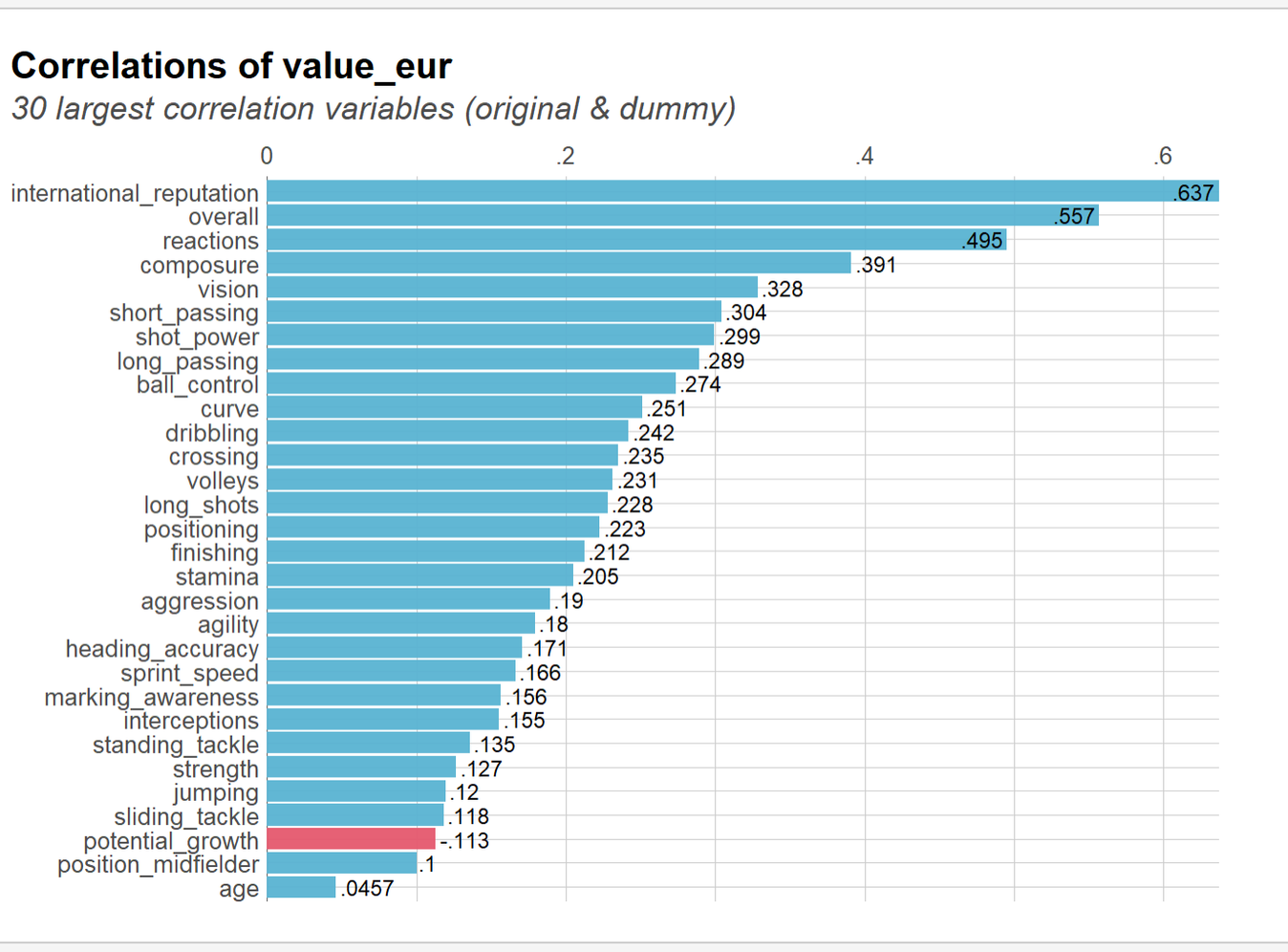

```
corr_cross(fifa_est, max_pvalue = 0.05, top = 30)
```

```
## Returning only the top 30. You may override with the 'top' argument
```

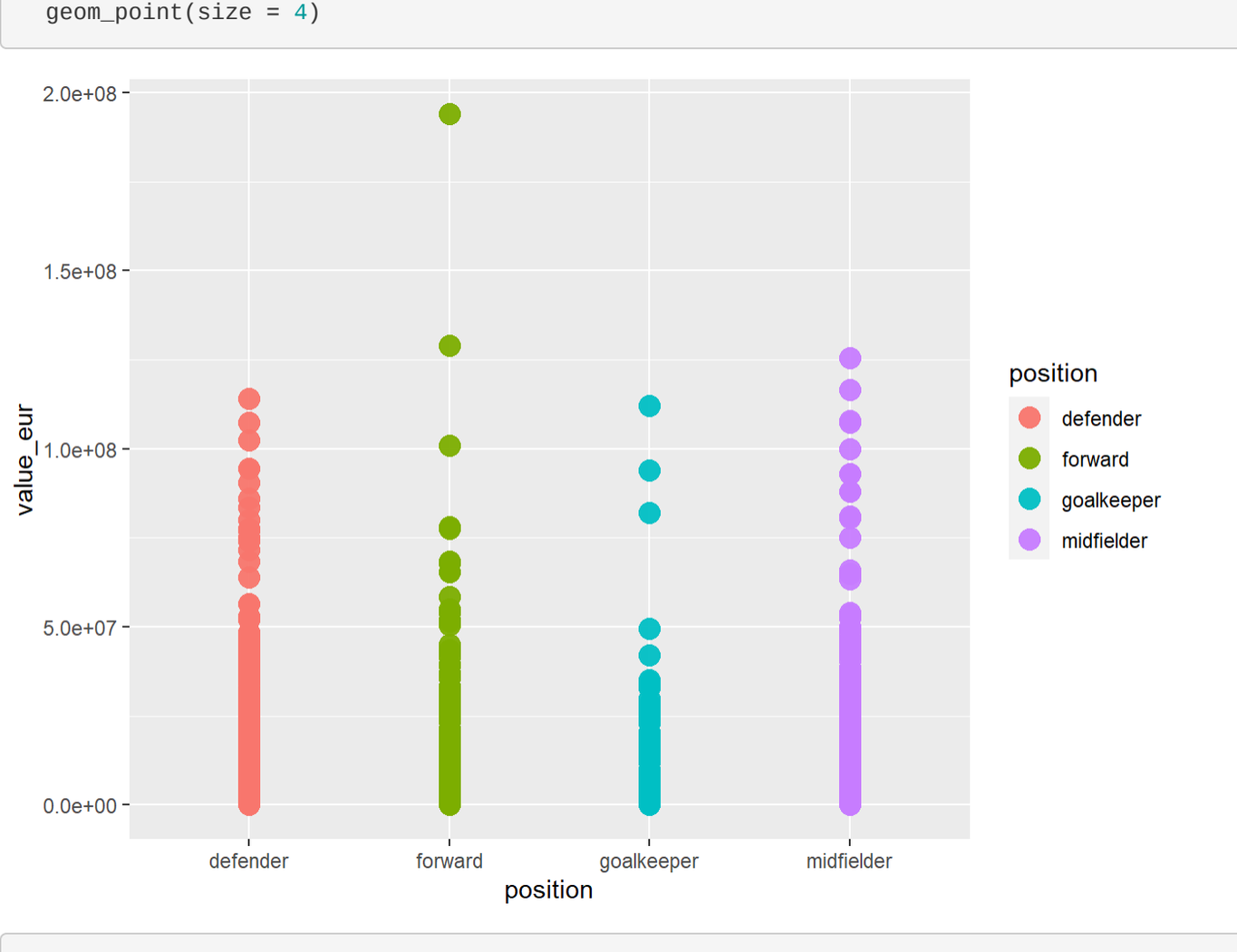
```
## Warning in .font_global(font, quiet = FALSE): Font 'Arial Narrow' is not
## installed, has other name, or can't be found
```




```
corr_var(fifa_est, value_eur, top = 30)
```




```
ggplot(data = fifa_est, aes(x = position, y = value_eur, group = position, color = position))+
  geom_point(size = 4)
```




```
table(fifa_est$position)
```

```
##
##   defender    forward goalkeeper midfielder
##       7127       2897        454       1787
```

RMSE Function

```
rmse =function(predicted, actual){sqrt(mean((actual - predicted)^2))}
```

Regression Models

Linear models

Training on estimation set

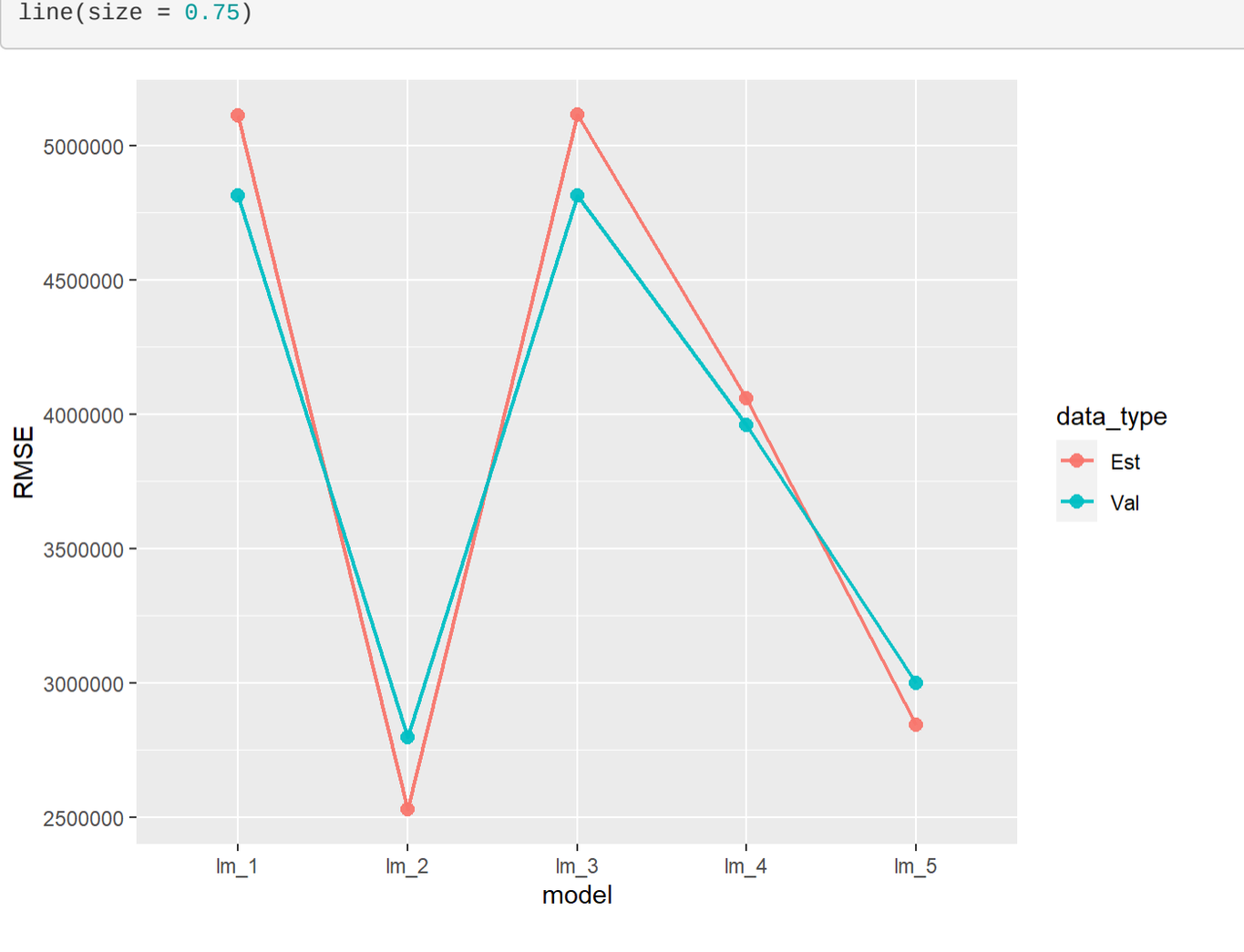
```
lm_model_list = list(
  lm_1 = lm(formula = value_eur ~ ., data = fifa_est),
  lm_2 = lm(formula = value_eur ~ .^2, data = fifa_est),
  lm_3 = step(lm(formula = value_eur ~ ., data = fifa_est), trace = FALSE, direction = 'both'),
  lm_4 = lm(formula = value_eur ~ poly(overall,2)+poly(age,2)+poly(international_reputation,2)+
            poly(finishing,2)+poly(long_passing,2)+poly(ball_control,2)+
            poly(sprint_speed,2)+poly(strength,2)+poly(interceptions,2)+
            poly(vision,2)+poly(sliding_tackle,2)+poly(gk_diving,2),data = fifa_est),
  lm_5 = lm(formula = value_eur ~ poly(overall,3)+poly(age,3)+poly(international_reputation,3)+
            poly(finishing,3)+poly(long_passing,3)+poly(ball_control,3)+
            poly(sprint_speed,3)+poly(strength,3)+poly(interceptions,3)+
            poly(vision,3)+poly(sliding_tackle,3)+poly(gk_diving,3),data = fifa_est))
```

Predict value and RMSE value of linear models

```
fifa_est_lm_predicted_list = lapply(lm_model_list, predict, fifa_est)
fifa_val_lm_predicted_list = lapply(lm_model_list, predict, fifa_val)

fifa_est_lm_rmse_vector = sapply(fifa_est_lm_predicted_list, rmse, fifa_est$value_eur)
fifa_val_lm_rmse_vector = sapply(fifa_val_lm_predicted_list, rmse, fifa_val$value_eur)
```

```
rmse_lm = data.frame(fifa_est_lm_rmse_vector, fifa_val_lm_rmse_vector)
colnames(rmse_lm) = c('Est','Val')
rownames(rmse_lm) = c('lm_1','lm_2','lm_3','lm_4','lm_5')
```

```
rmse_lm = data.matrix(rmse_lm)
rmse_lm = melt(rmse_lm)
colnames(rmse_lm) = c('model','data_type','RMSE')

ggplot(data = rmse_lm, aes(x = model, y = RMSE,group = data_type,color = data_type))+geom_point(size = 2.5)+geom_
line(size = 0.75)
```



KNN models

Feature scaling

Estimation set

```
fifa_est_scaled = data.frame(rep(0,nrow(fifa_est)))
center_vec = vector()
scale_vec = vector()

for (i in 1:(length(fifa_est)-2)){
  scaled_feature = scale(fifa_est[,i])
  fifa_est_scaled[,i] = scaled_feature
  center_vec = append(center_vec, attr(scaled_feature, 'scaled:center'))
  scale_vec = append(scale_vec, attr(scaled_feature, 'scaled:scale'))
}
fifa_est_scaled[,length(fifa_est_scaled)+1] = fifa_est$position
fifa_est_scaled[,length(fifa_est_scaled)+1] = fifa_est$value_eur
colnames(fifa_est_scaled) = colnames(fifa_est)
```

Validation

```
fifa_val_scaled = data.frame(rep(0,nrow(fifa_val)))
for (i in 1:(length(fifa_val)-2)){
  scaled_feature = scale(fifa_val[,i], scale = scale_vec[i], center = center_vec[i])
  scaled_feature = as.vector(scaled_feature)
  fifa_val_scaled[,i] = scaled_feature}
class(scaled_feature)
```

```
## [1] "numeric"
```

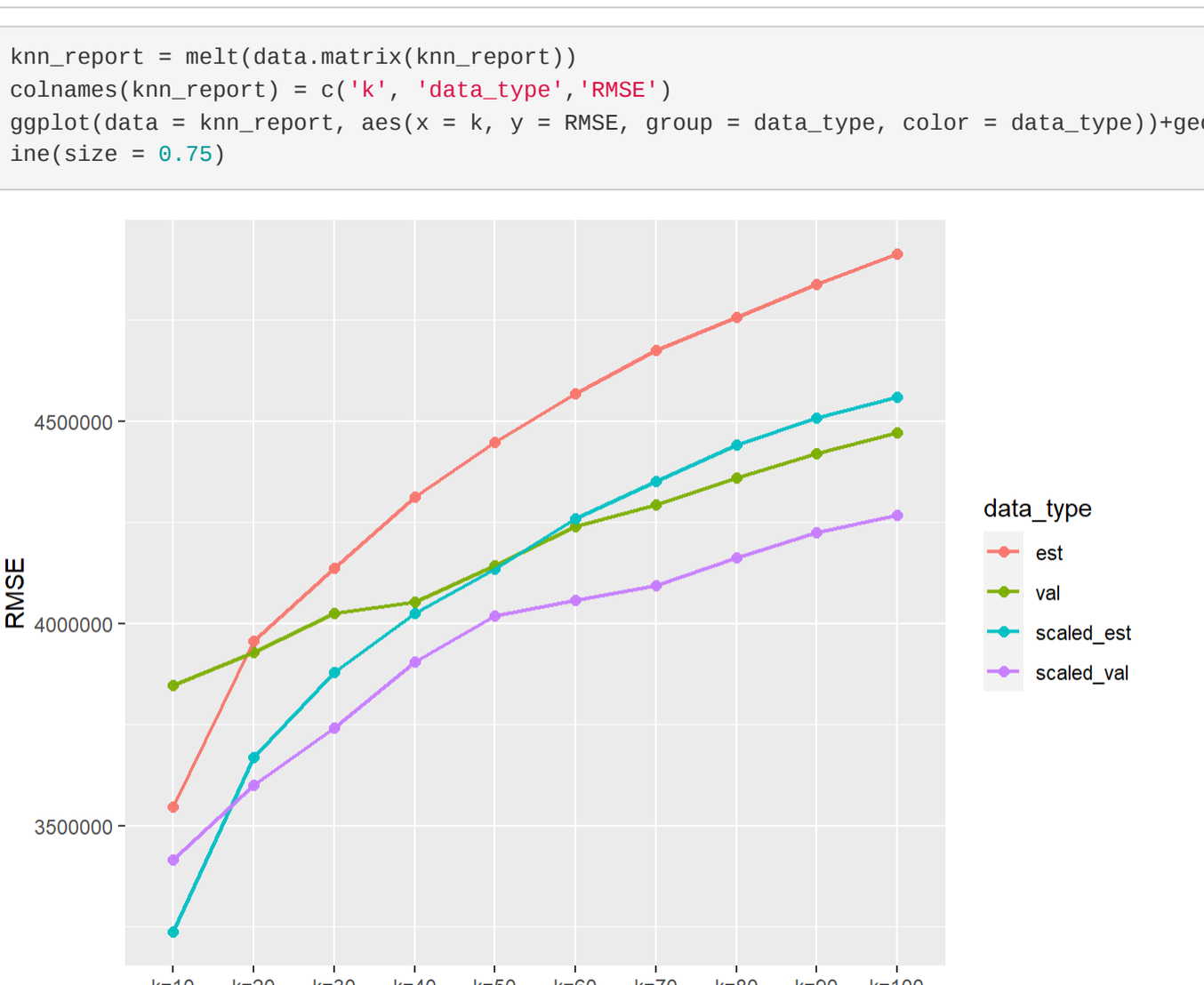
```
fifa_val_scaled[,length(fifa_val_scaled)+1] = fifa_val$position
fifa_val_scaled[,length(fifa_val_scaled)+1] = fifa_val$value_eur
colnames(fifa_val_scaled) = colnames(fifa_val)
```

```
fifa_knn_model_list = list()
k_list = seq(10,100,10)
for(i in 1:length(k_list)){fifa_knn_model_list[[i]] = knnreg(formula = value_eur ~ . ,
                                                              data = fifa_est,
                                                              k = k_list[i])}

fifa_knn_predicted_est_list = lapply(fifa_knn_model_list, predict, fifa_est)
fifa_rmse_est_list = sapply(fifa_knn_predicted_est_list, rmse, fifa_est$value_eur)

fifa_knn_predicted_val_list = lapply(fifa_knn_model_list, predict, fifa_val)
fifa_rmse_val_list = sapply(fifa_knn_predicted_val_list, rmse, fifa_val$value_eur)


fifa_scaled_knn_model_list = list()
for(i in 1:length(k_list)){fifa_scaled_knn_model_list[[i]] = knnreg(formula = value_eur ~ . ,
                                                                     data = fifa_est_scaled,
                                                                     k =  k_list[i])}

fifa_scaled_knn_predicted_est_list = lapply(fifa_scaled_knn_model_list, predict, fifa_est_scaled)
fifa_scaled_knn_rmse_est_list = sapply(fifa_scaled_knn_predicted_est_list, rmse, fifa_est_scaled$value_eur)

fifa_scaled_knn_predicted_val_list = lapply(fifa_scaled_knn_model_list, predict, fifa_val_scaled)
fifa_scaled_knn_rmse_val_list = sapply(fifa_scaled_knn_predicted_val_list, rmse, fifa_val_scaled$value_eur)
```

```
knn_report = data.frame(fifa_rmse_est_list,fifa_rmse_val_list,
                        fifa_scaled_knn_rmse_est_list, fifa_scaled_knn_rmse_val_list)

colnames(knn_report) = c('est','val','scaled_est','scaled_val')
rownames(knn_report) = c('k=10','k=20','k=30','k=40','k=50','k=60','k=70','k=80','k=90','k=100')
knn_report
```

```
##           est     val scaled_est scaled_val
## k=10  3546599 3847711    3236364    3415462
## k=20  3957183 3928143    3669803    3599962
## k=30  4136475 4024404    3878460    3740975
## k=40  4312672 4052335    4024642    3904888
## k=50  4449114 4143601    4134414    4019128
## k=60  4569169 4239056    4260066    4056857
## k=70  4676300 4293656    4350526    4094884
## k=80  4756708 4359516    4441625    4162836
## k=90  4838993 4420409    4507762    4224761
## k=100 4913750 4472461    4560098    4267872
```

```
knn_report = melt(data.matrix(knn_report))
colnames(knn_report) = c('k', 'data_type','RMSE')
ggplot(data = knn_report, aes(x = k, y = RMSE, group = data_type, color = data_type))+geom_point(size = 2)+geom_l
ine(size = 0.75)
```

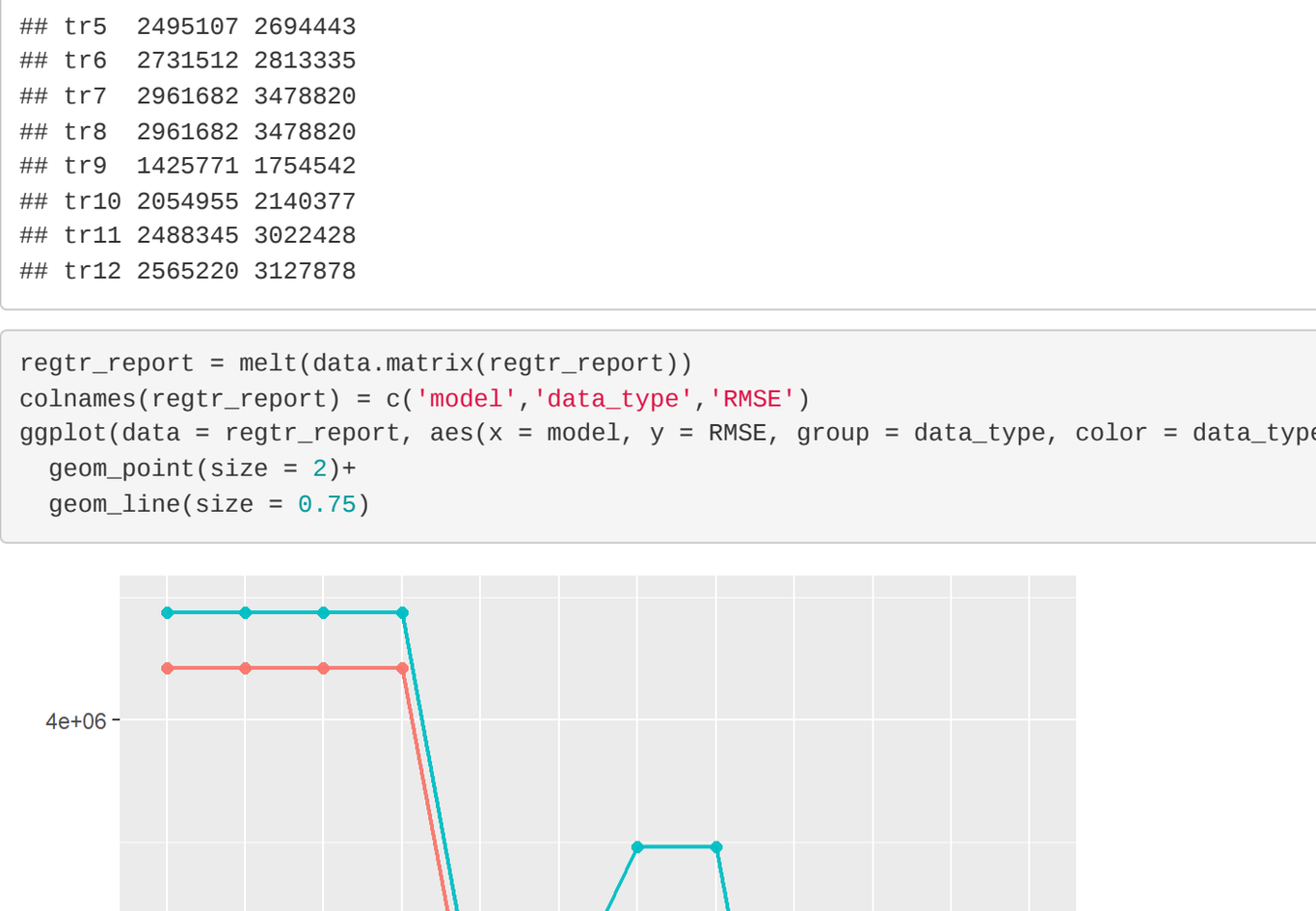


Regression tree models

```
cp_list = c(0.1,0.01,0.001)
minsplit_list = seq(10,100,30)

fifa_regtr_list = list(
  tr1 = rpart(value_eur ~ ., data = fifa_est, cp = 0.1, minsplit = 10),
  tr2 = rpart(value_eur ~ ., data = fifa_est, cp = 0.1, minsplit = 40),
  tr3 = rpart(value_eur ~ ., data = fifa_est, cp = 0.1, minsplit = 70),
  tr4 = rpart(value_eur ~ ., data = fifa_est, cp = 0.1, minsplit = 100),
  tr5 = rpart(value_eur ~ ., data = fifa_est, cp = 0.01, minsplit = 10),
  tr6 = rpart(value_eur ~ ., data = fifa_est, cp = 0.01, minsplit = 40),
  tr7 = rpart(value_eur ~ ., data = fifa_est, cp = 0.01, minsplit = 70),
  tr8 = rpart(value_eur ~ ., data = fifa_est, cp = 0.01, minsplit = 100),
  tr9 = rpart(value_eur ~ ., data = fifa_est, cp = 0.001, minsplit = 10),
  tr10 = rpart(value_eur ~ ., data = fifa_est, cp = 0.001, minsplit = 40),
  tr11 = rpart(value_eur ~ ., data = fifa_est, cp = 0.001, minsplit = 70),
  tr12 = rpart(value_eur ~ ., data = fifa_est, cp = 0.001, minsplit = 100))


fifa_regtr_predicted_est_list = lapply(fifa_regtr_list, predict, fifa_est)
fifa_regtr_rmse_est_list = sapply(fifa_regtr_predicted_est_list, rmse, fifa_est$value_eur)

fifa_regtr_predicted_val_list = lapply(fifa_regtr_list, predict, fifa_val)
fifa_regtr_rmse_val_list = sapply(fifa_regtr_predicted_val_list, rmse, fifa_val$value_eur)

regtr_report = data.frame(fifa_regtr_rmse_est_list,fifa_regtr_rmse_val_list)
colnames(regtr_report) = c('Est','Val')
rownames(regtr_report) = c('tr1','tr2','tr3','tr4','tr5','tr6',
                           'tr7','tr8','tr9','tr10','tr11','tr12')
regtr_report
```

```
##          Est     Val
## tr1  4212303 4438735
## tr2  4212303 4438735
## tr3  4212303 4438735
## tr4  4212303 4438735
## tr5  2495107 2694443
## tr6  2731512 2813335
## tr7  2961682 3478820
## tr8  2961682 3478820
## tr9  1425771 1754542
## tr10 2054955 2140377
## tr11 2488345 3022428
## tr12 2565220 3127878
```

```
regtr_report = melt(data.matrix(regtr_report))
colnames(regtr_report) = c('model','data_type','RMSE')
ggplot(data = regtr_report, aes(x = model, y = RMSE, group = data_type, color = data_type))+
  geom_point(size = 2)+
  geom_line(size = 0.75)
```




```
rpart.plot(fifa_regtr_list$tr9)
```

Fit final model on train data and calculate RMSE in predicting test set.

```
tr = rpart(value_eur ~ ., data = fifa_trn, cp = 0.001, minsplit = 10)
rmse(predict(tr, fifa_tst), fifa_tst$value_eur)
```

```
## [1] 2143079
```

```
rpart.plot(tr)
```