Predicting Player Rating in FIFA

Junqi Liao, Kaiwen Dong, Raymond Tan

07/08/2019

Contents

Motivation and introduction of Report

Data

Preprocessing

Data Visualization

Smoothing Methods

Linear Models

Multiple Linear Regression

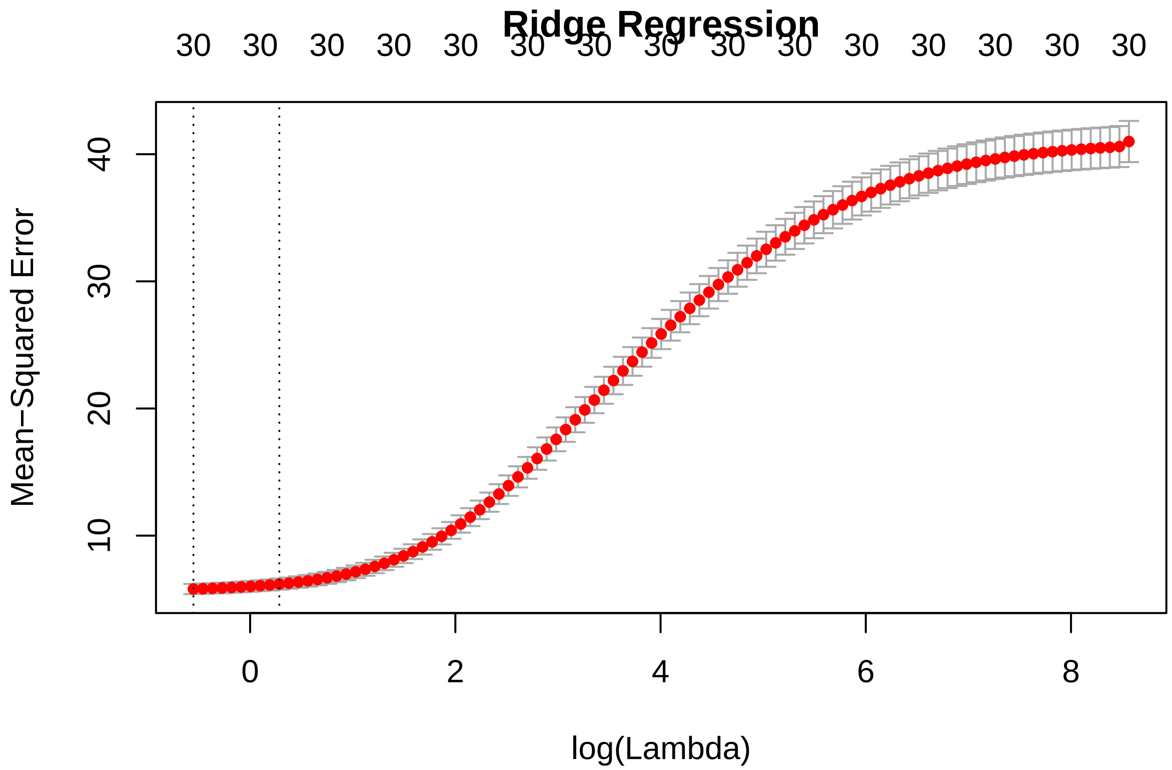
LASSO Regression

Ridge Regression

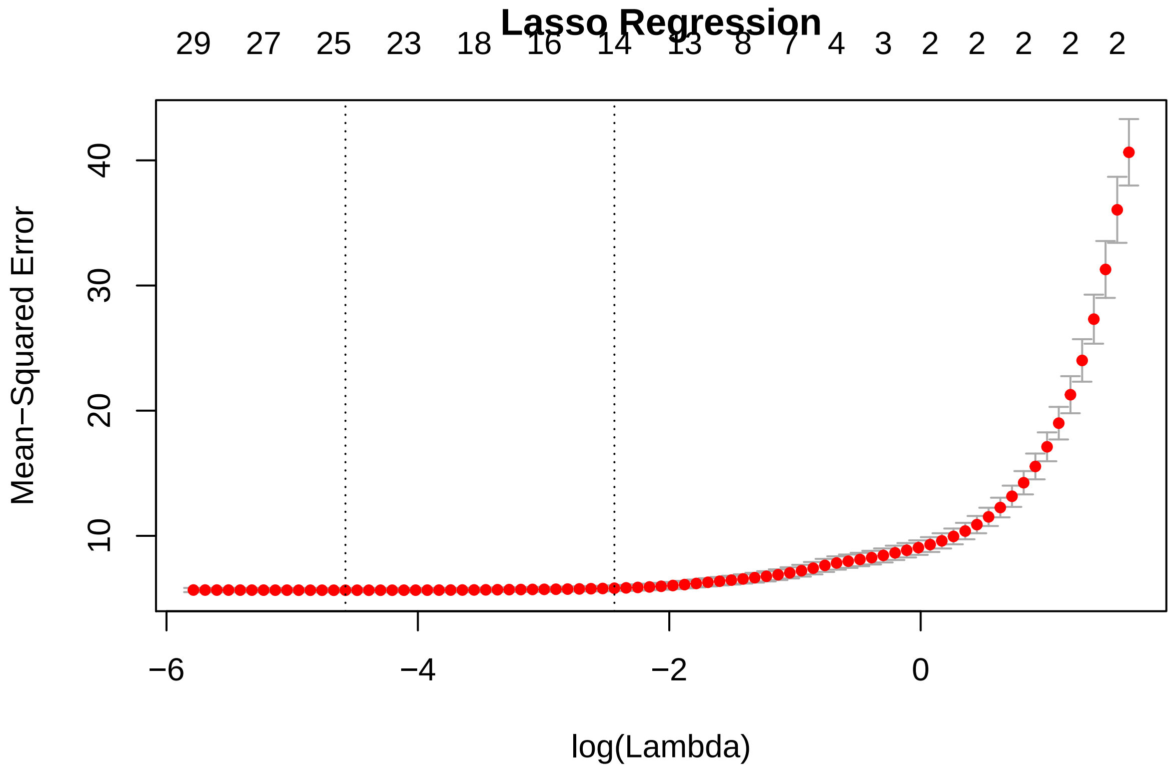
```
soccer <- soccer.raw
soccer$player_name <- NULL
soccer$set <- ifelse(runif(n=nrow(soccer)) > 0.85,yes = 1,no = 2)
#Split data into training set and testing set
soccer.train <- soccer[which(soccer$set == 1),]
soccer.test <- soccer[which(soccer$set ==2),]
soccer.train$set <- NULL
soccer.test$set <- NULL
soccer.train.x <- soccer.train[,2:length(soccer.train)]



ridge_model <- lm.ridge(soccer.train$overall_rating ~ .,data = soccer.train,lambda = log(seq(0, 10, .00
select(ridge_model)
```

```
## modified HKB estimator is 5.882207
## modified L-W estimator is 4.264822
## smallest value of GCV  at 2.302585
```

```
ridge_model <- cv.glmnet(as.matrix(soccer.train.x),soccer.train$overall_rating,alpha = 0,nfolds = 5)
lasso_model <- cv.glmnet(as.matrix(soccer.train.x),soccer.train$overall_rating,alpha = 1,nfolds = 5)
ridge <- lm.ridge(soccer.train$overall_rating ~ .,data = soccer.train)
best_lambda.ridge <- ridge_model$lambda.1se
best_lambda.lasso <- lasso_model$lambda.1se
plot(ridge_model,main = "Ridge Regression")
```

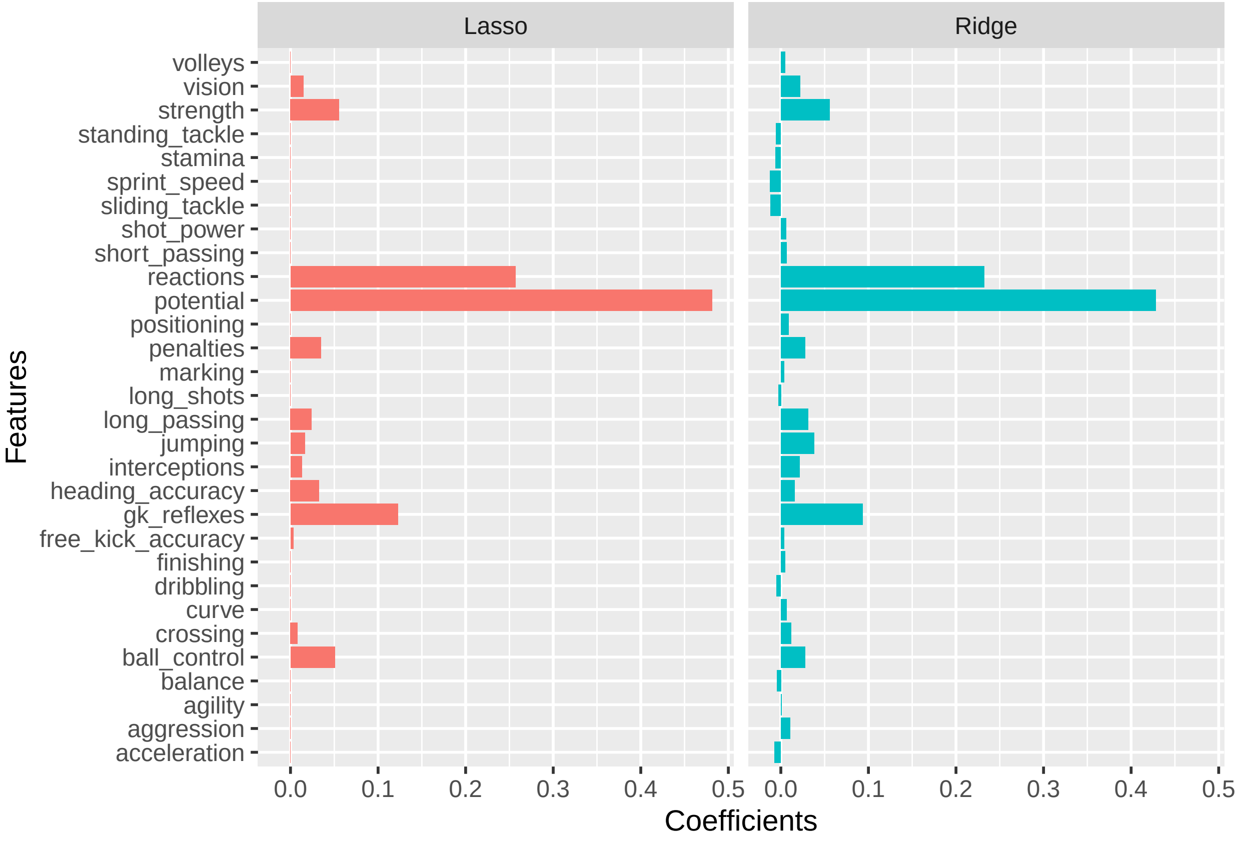


```r
plot(lasso_model,main = "Lasso Regression")
```



```r
ridge_coeff <- ridge_model$glmnet.fit$beta[,ridge_model$glmnet.fit$lambda == best_lambda.ridge]
lasso_coeff <- lasso_model$glmnet.fit$beta[,lasso_model$glmnet.fit$lambda == best_lambda.lasso]
```

```
#Compare Coefficients:
coeff <- data.table(Lasso = lasso_coeff,Ridge = ridge_coeff)
coeff[,Features :=names(ridge_coeff)]
to_plot <- melt(data = coeff,id.vars = 'Features',variable.name = 'Model',value.name = 'Coefficients')
ggplot(to_plot,aes(x=Features,y=Coefficients,fill=Model)) + coord_flip() + geom_bar(stat = 'identity')
```



Non-Linear Model

GAM

###Tensor Producting Smoothing

Regression Tree

Random Forest

Statistical Conclusions

Conclusion in the context of the problem

Future Work

Contribution

Appendix

Variables

- player_name: The name of the player
- finishing: The accuracy of shots using foot, inside the penalty area
- dribbling: The ability to kepp possesion of the ball.
- ball_control: The ability to keep your ball under your feet with velocity.
- reactions: How quickly a player respinds a situation.
- stamina: Determine the rate at which a player will tire during a game.
- interceptions: The ability to intercepts a pass where the ball is going and stop it from going there.
- marking: The ability to track and defend an opposing player.
- overall_rating: The rating of the player based on all attributes.
- heading_accuracy: The accuracy of he player either a pass or a shot by using head.
- curve: The ability to shoot the ball in a curved shape.
- acceleration: Increase in the rate of speed of a player.
- balance: The ability to maintain balance after a physical challenge.
- strength: The ability to win a physical challenge.
- positioning: The ability to read the game offensively, get into good positions, make effective runs, and avoid getting caught offside.
- standing_tackle:The ability of the player to time standing tackles so that they win the ball rather than give away a foul.
- potential: A peak in overall rating that a player could reach.
- short_passing: The ability to perform a pass in short distance.
- free_kick_accuracy: The accuracy of a direct free kick on goal.(Free kick: an unimpeded kick of the stationary ball awared to one side as a penalty for a foul by the other side)
- sprint_speed: The maximum speed over a short distance of a player.
- shot_power: How hard can the player hits the ball when taking a shot at goal.
- long_shots: The accuracy of shots from outside of the penalty area.
- vision: The player's awareness of the position of his team mates & opponents around him.
- sliding_tackle: The ability of the player to time sliding tackles so that they win the ball rather than give away a foul.
- crossing: The accuracy of the player crosses the ball.
- volleys: The accuracy of a player strike or hit the ball at goal before it touches the ground.
- long_passing: The ability to perform a long pass in the air and on the ground to his teammate.
- agility: The ability of a player to move or turn in game.
- jumping: The vertical distance of a player can jump from the ground.

- aggression: The frequency & aggression of jostling, tackling & slide tackling.
- penalties: The ability to take penalties.
- gk_reflexes: The ability to react a ball in movement at goal by the goal keeper.

R-Code

```
knitr::opts_chunk$set(echo = TRUE)
setwd("/Users/Raymond/Desktop/Raymond Tan/HW/4B/STAT444/soccer-rating-prediction/data")
soccer.raw <- read.table("rating_potential.csv",sep = " ",na.strings = "NA")
library(glmnet)
library(data.table)
library(ggplot2)
library(MASS)
set.seed(123)
soccer <- soccer.raw
soccer$player_name <- NULL
soccer$set <- ifelse(runif(n=nrow(soccer)) > 0.85,yes = 1,no = 2)
#Split data into training set and testing set
soccer.train <- soccer[which(soccer$set == 1),]
soccer.test <- soccer[which(soccer$set ==2),]
soccer.train$set <- NULL
soccer.test$set <- NULL
soccer.train.x <- soccer.train[,2:length(soccer.train)]



ridge_model <- lm.ridge(soccer.train$overall_rating ~ .,data = soccer.train,lambda = log(seq(0, 10, .00
select(ridge_model)


ridge_model <- cv.glmnet(as.matrix(soccer.train.x),soccer.train$overall_rating,alpha = 0,nfolds = 5)
lasso_model <- cv.glmnet(as.matrix(soccer.train.x),soccer.train$overall_rating,alpha = 1,nfolds = 5)
ridge <- lm.ridge(soccer.train$overall_rating ~ .,data = soccer.train)
best_lambda.ridge <- ridge_model$lambda.1se
best_lambda.lasso <- lasso_model$lambda.1se
plot(ridge_model,main = "Ridge Regression")
plot(lasso_model,main = "Lasso Regression")
ridge_coeff <- ridge_model$glmnet.fit$beta[,ridge_model$glmnet.fit$lambda == best_lambda.ridge]
lasso_coeff <- lasso_model$glmnet.fit$beta[,lasso_model$glmnet.fit$lambda == best_lambda.lasso]


#Compare Coefficients:
coeff <- data.table(Lasso = lasso_coeff,Ridge = ridge_coeff)
coeff[,Features :=names(ridge_coeff)]
to_plot <- melt(data = coeff,id.vars = 'Features',variable.name = 'Model',value.name = 'Coefficients')
ggplot(to_plot,aes(x=Features,y=Coefficients,fill=Model)) + coord_flip() + geom_bar(stat = 'identity')
```