Assignment Model Prediction

Randell Rasiman - 4281209

January 2019

About the dataset

FIFA 19 is a football game by EA. This dataset contains all players present in the game, together with their in-game attributes. The dataset is retrieved from https://www.kaggle.com/karangadiya/fifa19/version/4

Required packages

For this assignment the tidyverse and glmnet package were used.

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------------------
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.2.1     v forcats 0.3.0
```

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

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

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

```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loaded glmnet 2.0-16
```

Prepare the R environment

```
#create a temporary df to store the data
fifa.temp <- read_csv("data/data.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
```

```
##   .default = col_character(),
##   X1 = col_double(),
##   ID = col_double(),
##   Age = col_double(),
##   Overall = col_double(),
##   Potential = col_double(),
##   Special = col_double(),
##   `International Reputation` = col_double(),
##   `Weak Foot` = col_double(),
##   `Skill Moves` = col_double(),
##   `Jersey Number` = col_double(),
##   Crossing = col_double(),
##   Finishing = col_double(),
##   HeadingAccuracy = col_double(),
##   ShortPassing = col_double(),
##   Volleys = col_double(),
##   Dribbling = col_double(),
##   Curve = col_double(),
##   FKAccuracy = col_double(),
##   LongPassing = col_double(),
##   BallControl = col_double()
##   # ... with 24 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
#transform the df into a tibble. For the prediction we only like to focus on the age and skill attribut

#Values are either mentioned in thousands or millions or they are worth €0. We first split them.
fifa.k <- fifa.temp %>% filter(grepl('K', Value))
fifa.m <- fifa.temp %>% filter(grepl('M', Value))
fifa.r <- fifa.temp %>% filter(grepl('\\€0', Value))

#mutate them to the same format.
fifa.k <- fifa.k %>% mutate(Value = as.numeric(gsub("[\\€K]", "", fifa.k$Value)) * 1000)
fifa.m <- fifa.m %>% mutate(Value = as.numeric(gsub("[\\€M]", "", fifa.m$Value)) * 1000000)
fifa.r <- fifa.r %>% mutate(Value = as.numeric(gsub("[\\€K]", "", fifa.r$Value)))

#put them back in 1 dataset. We exclude the players which have a Value of €0.
fifa.temp <- bind_rows(fifa.k, fifa.m)

#transform into a tibble
fifa <- tibble(Name = as.character(fifa.temp$Name),
               Age = as.numeric(fifa.temp$Age),
               Value = as.numeric(fifa.temp$Value),
               Wage = as.numeric(gsub("[\\€K]", "", fifa.temp$Wage)) * 1000, #Mentioned in thousands.
               Crossing = as.numeric(fifa.temp$Crossing),
               Finishing = as.numeric(fifa.temp$Finishing),
               HeadingAccuracy = as.numeric(fifa.temp$HeadingAccuracy),
               ShortPassing = as.numeric(fifa.temp$ShortPassing),
               Volleys = as.numeric(fifa.temp$Volleys),
               Dribbling = as.numeric(fifa.temp$Dribbling),
               Curve = as.numeric(fifa.temp$Curve),
               FKAccuracy = as.numeric(fifa.temp$FKAccuracy),
               LongPassing = as.numeric(fifa.temp$LongPassing),
```

```
                BallControl = as.numeric(fifa.temp$BallControl),
                Acceleration = as.numeric(fifa.temp$Acceleration),
                SprintSpeed = as.numeric(fifa.temp$SprintSpeed),
                Agility = as.numeric(fifa.temp$Agility),
                Balance = as.numeric(fifa.temp$Balance),
                ShotPower = as.numeric(fifa.temp$ShotPower),
                Jumping = as.numeric(fifa.temp$Jumping),
                Stamina = as.numeric(fifa.temp$Stamina),
                Strength = as.numeric(fifa.temp$Strength),
                LongShots = as.numeric(fifa.temp$LongShots),
                Aggression = as.numeric(fifa.temp$Aggression),
                Interceptions = as.numeric(fifa.temp$Interceptions),
                Positioning = as.numeric(fifa.temp$Positioning),
                Vision = as.numeric(fifa.temp$Vision),
                Penalties = as.numeric(fifa.temp$Penalties),
                Composure = as.numeric(fifa.temp$Composure),
                Marking = as.numeric(fifa.temp$Marking),
                StandingTackle = as.numeric(fifa.temp$StandingTackle),
                SlidingTackle = as.numeric(fifa.temp$SlidingTackle),
                GKDiving = as.numeric(fifa.temp$GKDiving),
                GKHandling = as.numeric(fifa.temp$GKHandling),
                GKKicking = as.numeric(fifa.temp$GKKicking),
                GKPositioning = as.numeric(fifa.temp$GKPositioning),
                GKReflexes = as.numeric(fifa.temp$GKReflexes)
                )

#Only keep the values which are not empty
fifa <-fifa %>% filter(!is.na(Name),
                       !is.na(Age),
                       !is.na(Crossing),
                       !is.na(Finishing),
                       !is.na(HeadingAccuracy),
                       !is.na(ShortPassing),
                       !is.na(Volleys),
                       !is.na(Dribbling),
                       !is.na(Curve),
                       !is.na(FKAccuracy),
                       !is.na(LongPassing),
                       !is.na(BallControl),
                       !is.na(Acceleration),
                       !is.na(SprintSpeed),
                       !is.na(Agility),
                       !is.na(Balance),
                       !is.na(ShotPower),
                       !is.na(Jumping),
                       !is.na(Stamina),
                       !is.na(Strength),
                       !is.na(LongShots),
                       !is.na(Aggression),
                       !is.na(Interceptions),
                       !is.na(Positioning),
                       !is.na(Vision),
                       !is.na(Penalties),
```

```
                        !is.na(Composure),
                        !is.na(Marking),
                        !is.na(StandingTackle),
                        !is.na(SlidingTackle),
                        !is.na(GKDiving),
                        !is.na(GKHandling),
                        !is.na(GKKicking),
                        !is.na(GKPositioning),
                        !is.na(GKReflexes)
                        )

#remove the temporary df's
rm(fifa.temp)
rm(fifa.k)
rm(fifa.m)
rm(fifa.r)
```

Summary of the data

```
 fifa %>% summarise("mean age" = mean(Age),
                    "variance in age" = var(Age),
                    "mean wage" = mean(Wage),
                    "variance in wage" = var(Wage),
                    "mean value" = mean(Value),
                    "variance in value" = var(Value)
            )
```

```
## # A tibble: 1 x 6
##   `mean age` `variance in ag~ `mean wage` `variance in wa~ `mean value`
##        <dbl>            <dbl>       <dbl>            <dbl>        <dbl>
## 1       25.1             21.7       9888.       490562961.     2450133.
## # ... with 1 more variable: `variance in value` <dbl>
```

Split the data into training (50%), validation (30%) and test (20%) data sets.

```
split <- c(rep("train", 8954), rep("valid", 5372), rep("test",  3581))

fifa <- fifa %>% mutate(Split = sample(split))

fifa_train <- fifa %>% filter(Split == "train")
fifa_valid <- fifa %>% filter(Split == "valid")
fifa_test  <- fifa %>% filter(Split == "test")
```

Create a function which calculate the MSE.

```
mse <- function(y_true, y_pred) {
  mean((y_true - y_pred)^2)
}
```

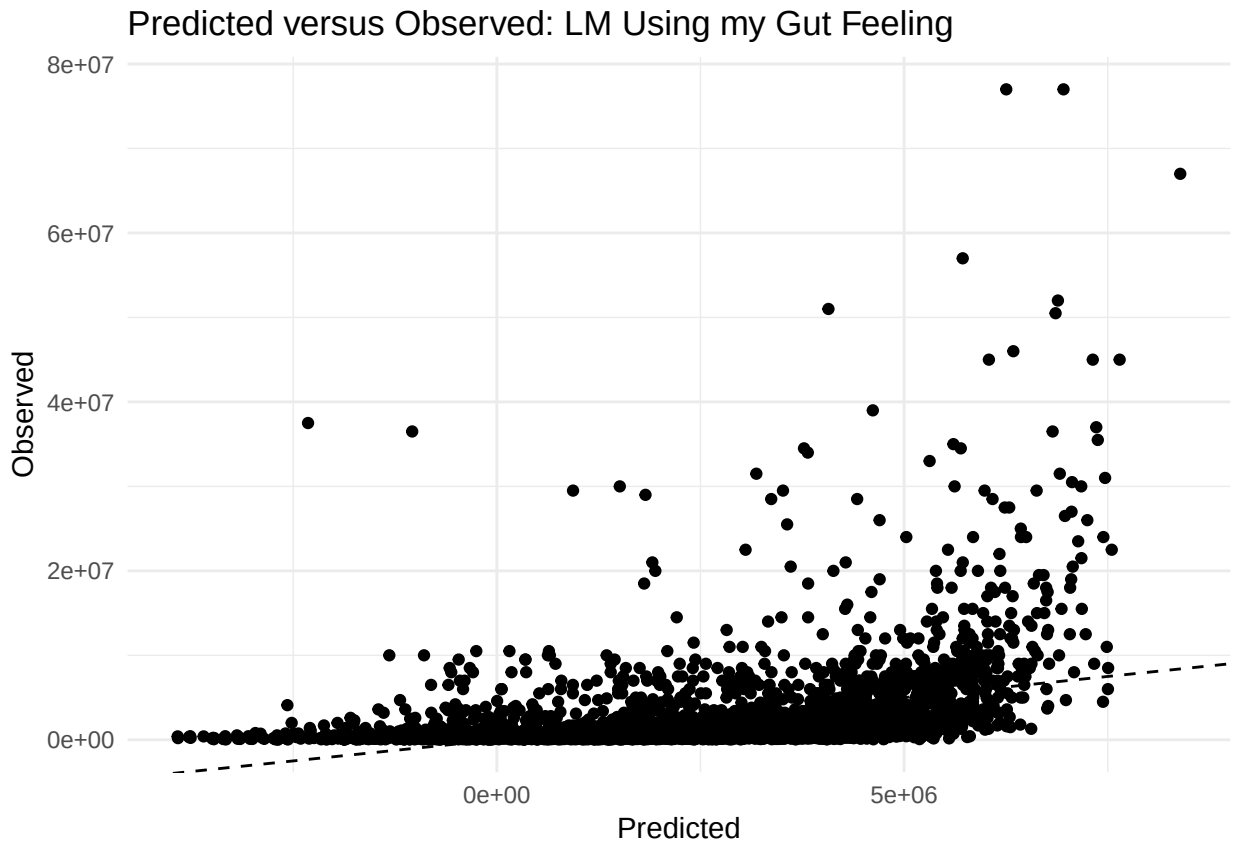
Best Linear Prediction Using Gut Feeling

Offensive Players are in general the most expensive. So good offensive attributes lead to a higher value? We tried the predictors Finishing, Free Kick Accuracy and Vision to predict the Value and fitted a linear regression model.

Plot Predicted vs Observed

```r
y_pred_lm <- predict(lm(Value ~ Finishing + FKAccuracy + Vision, fifa_train), newdata = fifa_test)

tibble(Predicted = y_pred_lm, Observed = fifa_test$Value) %>%
  ggplot(aes(x = Predicted, y = Observed)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, lty = 2) +
  theme_minimal() +
  labs(title = "Predicted versus Observed: LM Using my Gut Feeling")
```



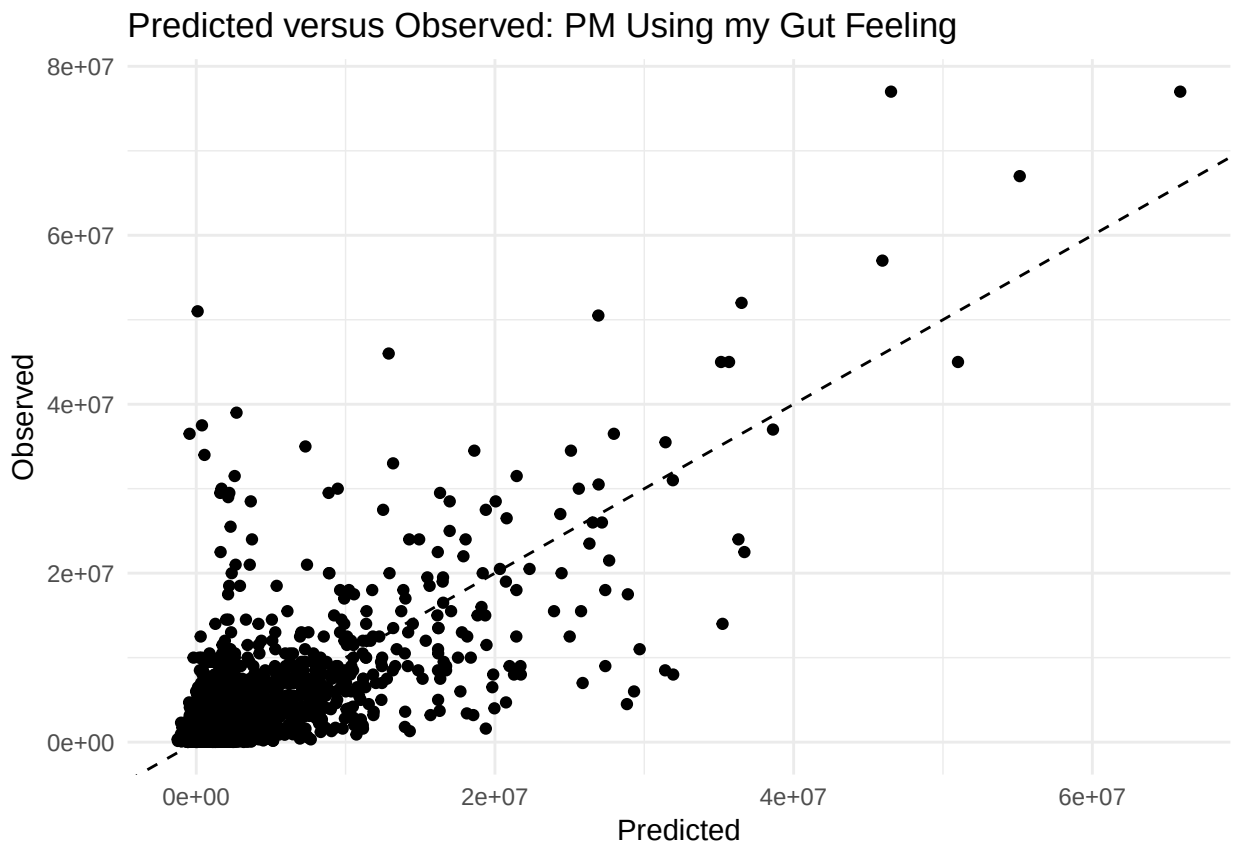

```r
mse_lm <- mse(fifa_test$Value, y_pred_lm)
```

Best Polynomial Regression Using Gut Feeling

It turns out our Gut Feeling is not that good... Maybe try a polynomial regression with the same predictors?

```r
y_pred_pm <- predict(lm(Value ~ Finishing + I(Finishing^2) + I(Finishing^3) + I(Finishing^4) +
                                FKAccuracy + I(FKAccuracy^2) + I(FKAccuracy^3) + I(FKAccuracy^4) +
                                Vision + I(Vision^2) + I(Vision^3) + I(Vision^4),
                                fifa_train), newdata = fifa_test)

tibble(Predicted = y_pred_pm, Observed = fifa_test$Value) %>%
  ggplot(aes(x = Predicted, y = Observed)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, lty = 2) +
  theme_minimal() +
  labs(title = "Predicted versus Observed: PM Using my Gut Feeling")
```




```
mse_pm <- mse(fifa_test$Value, y_pred_pm)
mse_pm
```

```
## [1] 1.344713e+13
```

Somewhat better

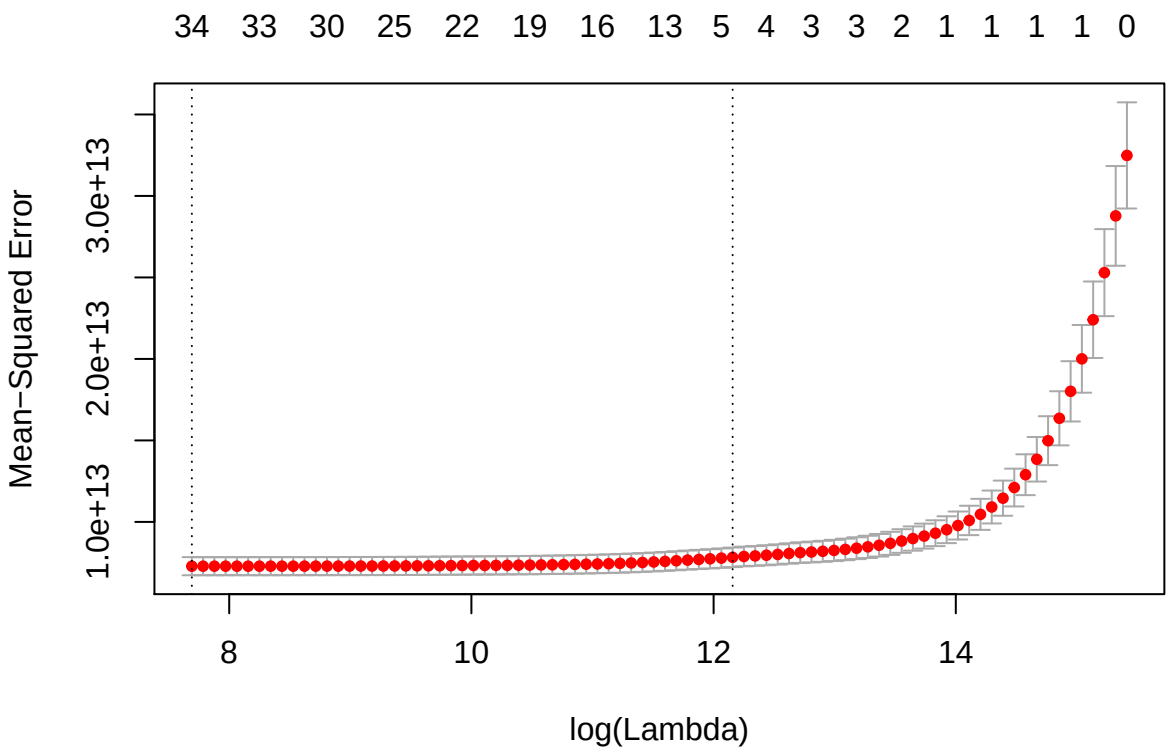
Best Prediction Using Lasso

Instead of choosing the predictors ourselves, we let the computer decide what's best. We use LASSO for this and let lambda (the penalty) be chosen using 15-fold cross validation.

```
x_cv <- model.matrix(Value ~ ., bind_rows(fifa_train, fifa_valid) %>% select(-Split, -Name))[, -1]
result_cv <- cv.glmnet(x = x_cv, y = c(fifa_train$Value, fifa_valid$Value), nfolds = 15)
best_lambda <- result_cv$lambda.min
best_lambda
```

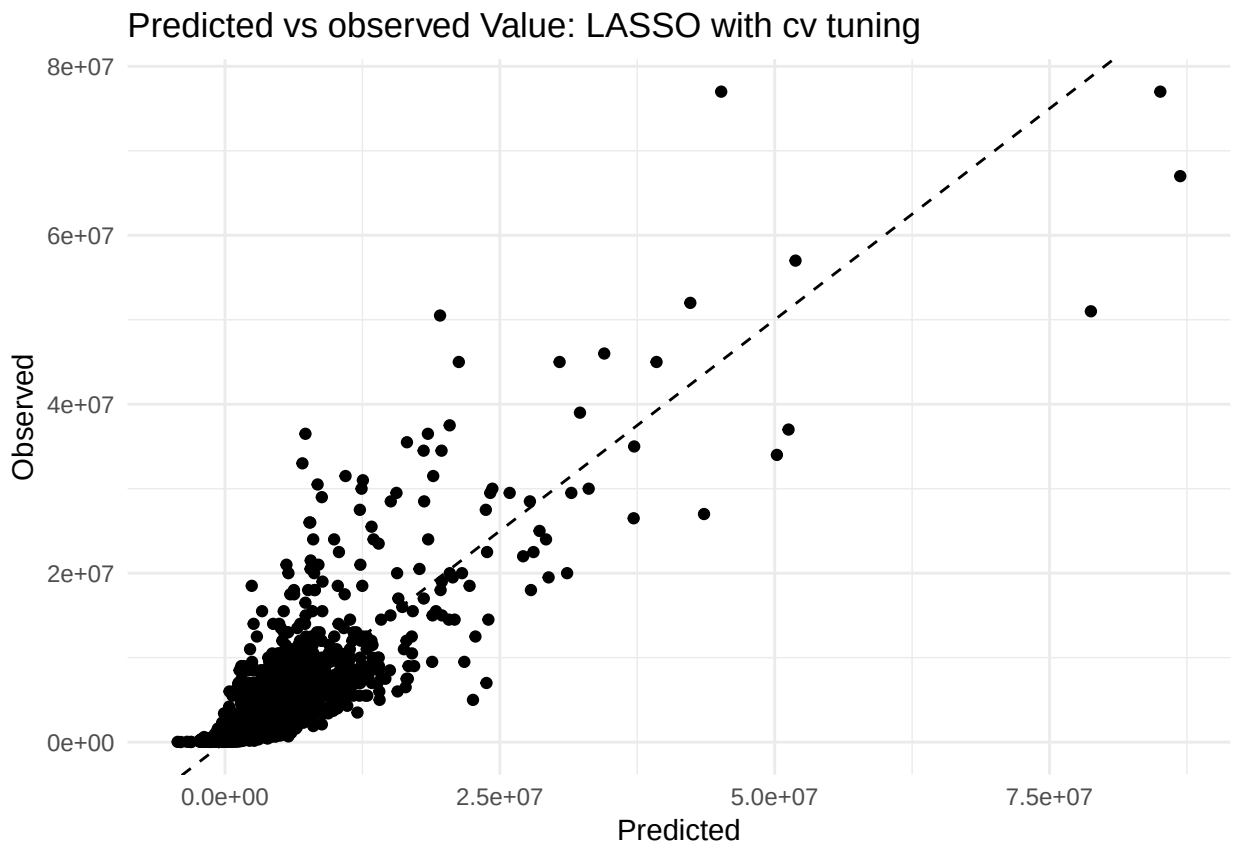
```
## [1] 2189.597
```

```
plot(result_cv)
```




```
x_test <- model.matrix(Value ~ ., data = fifa_test %>% select(-Split, -Name))[, -1]
y_pred <- as.numeric(predict(result_cv, newx = x_test, s = best_lambda))

tibble(Predicted = y_pred, Observed = fifa_test$Value) %>%
  ggplot(aes(x = Predicted, y = Observed)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, lty = 2) +
  theme_minimal() +
  labs(title = "Predicted vs observed Value: LASSO with cv tuning")
```



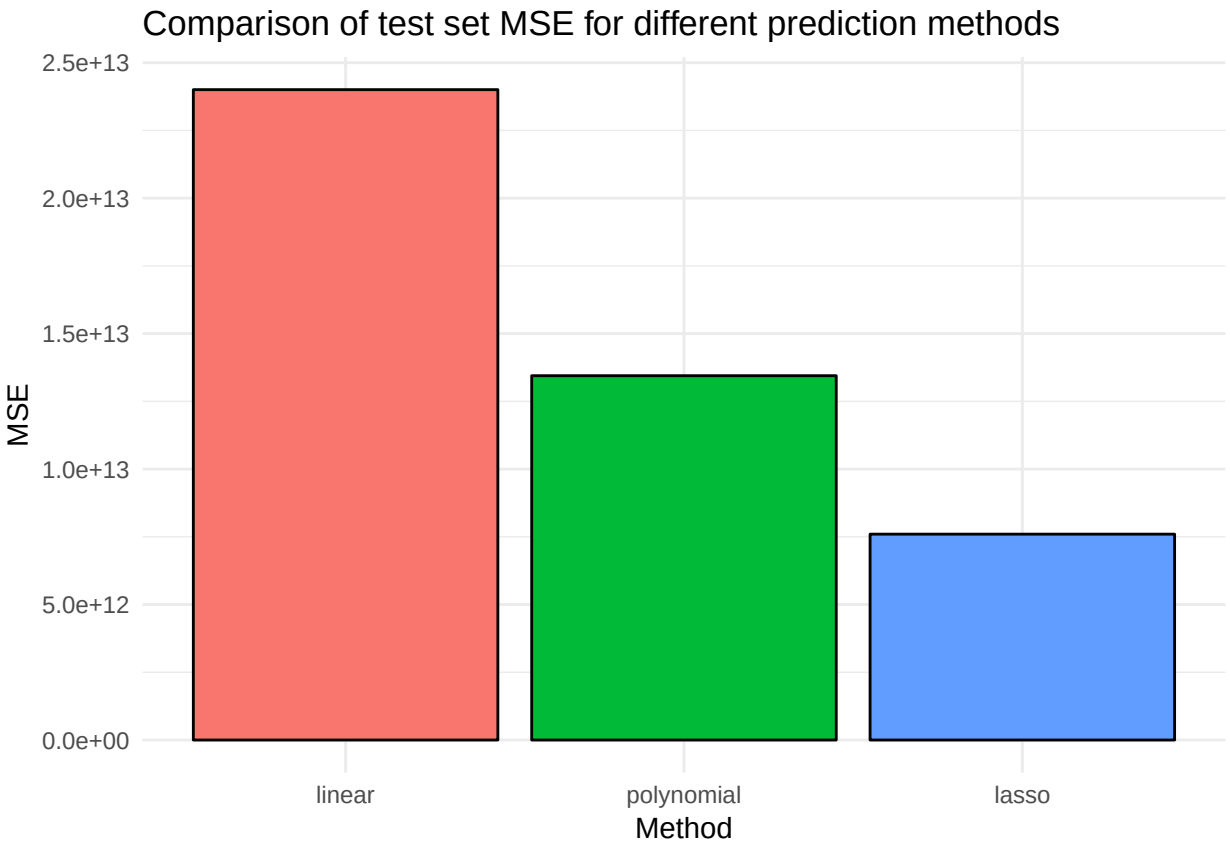

```
mse_lasso <- mse(fifa_test$Value, predict(result_cv, newx = x_test, s = best_lambda))
mse_lasso
```

```
## [1] 7.59846e+12
```

Which is the best pick?

```
mses <- c(mse_lm, mse_pm, mse_lasso)

tibble(Method = as_factor(c("linear", "polynomial", "lasso")), MSE = mses) %>%
  ggplot(aes(x = Method, y = MSE, fill = Method)) +
  geom_bar(stat = "identity", col = "black") +
  theme_minimal() +
  theme(legend.position = "none") +
  labs(title = "Comparison of test set MSE for different prediction methods")
```



Conclusion

Lasso with 15-fold Cross Validation produces the best predictions.