Predicting Wage by Club & Player Attributes

Justo Garcia Miranda Heyer James Kistner Cady Stringer

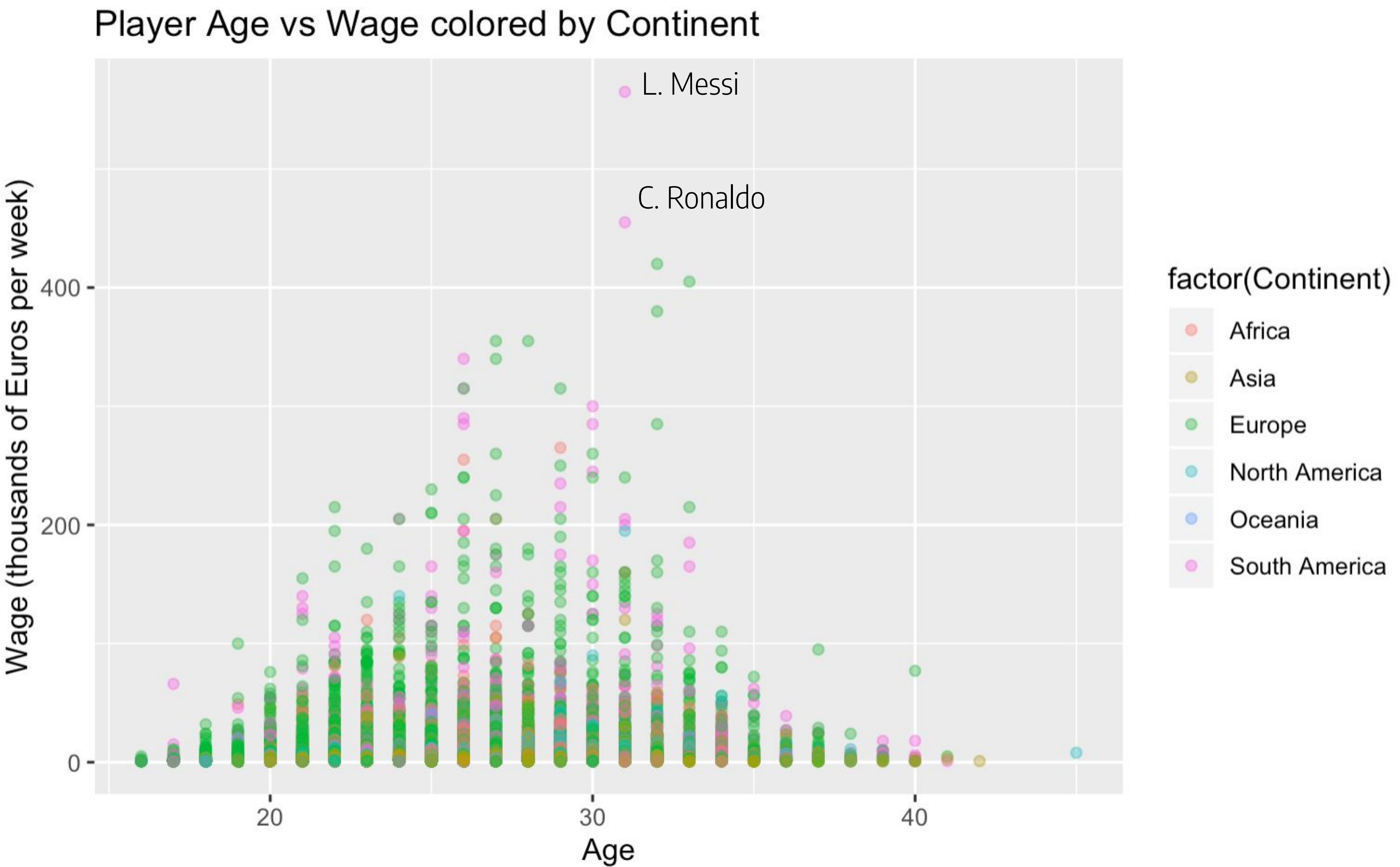
Dataset

- ❏ FIFA '19 Player Statistics
  - ❏ 18,000 players
  - ❏ 90 attributes

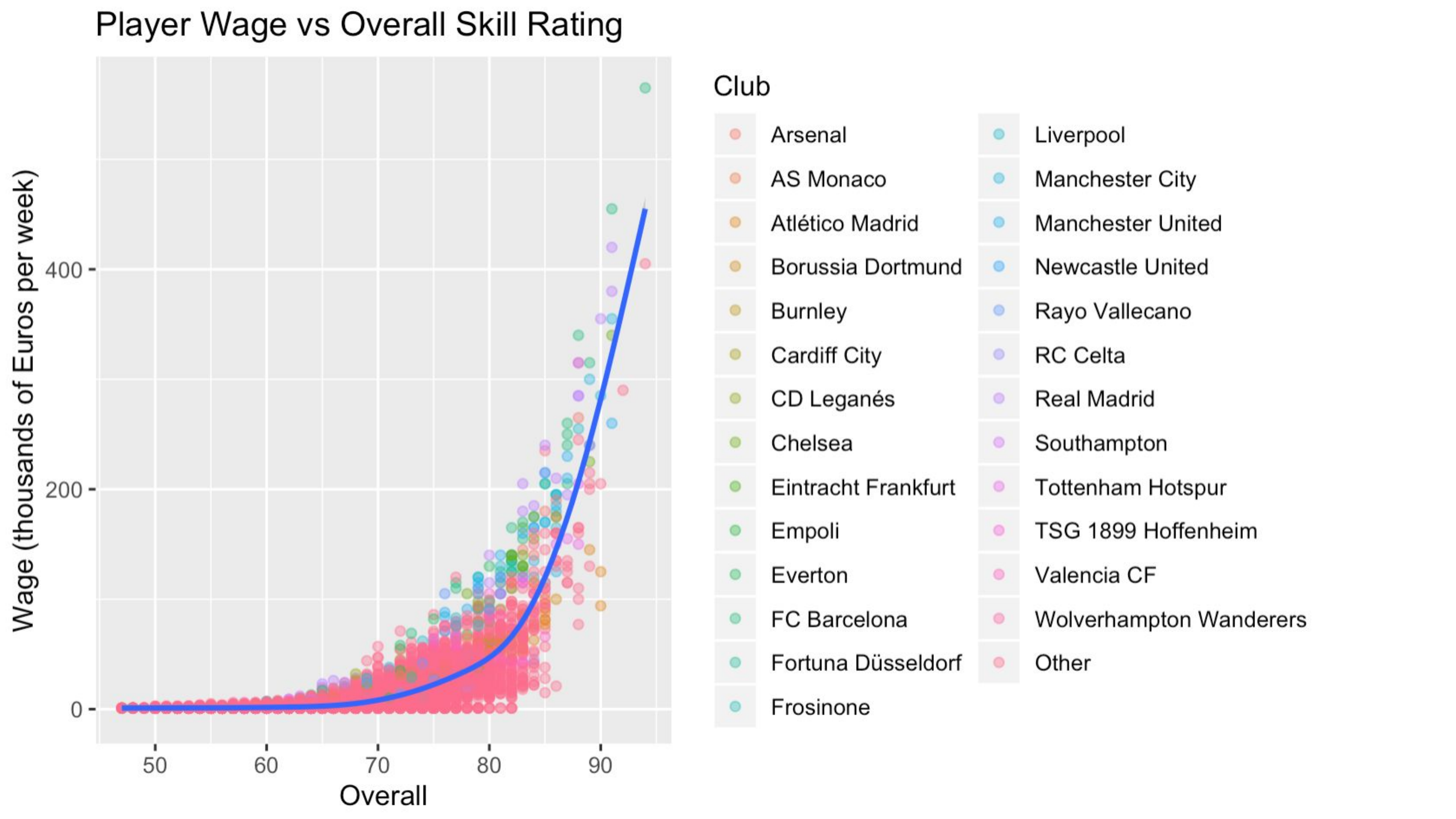
Business Use Case

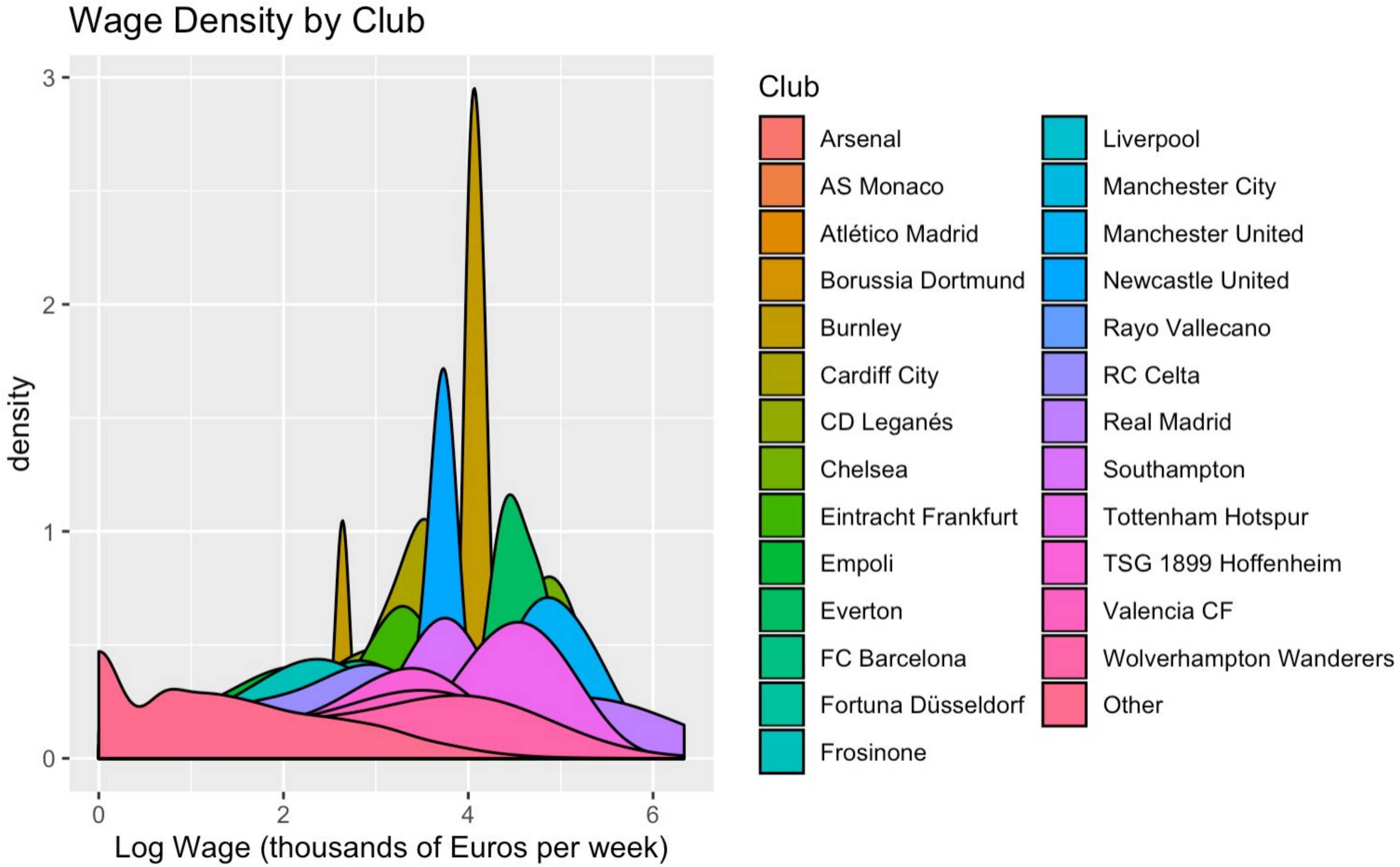
- Soccer team management can use our model to predict what a player is worth *before* negotiating a contract
  - Better allocate market cap (all the money they have for player salaries) to players
  - Helps them decide when to splurge and when to save

Data Exploration



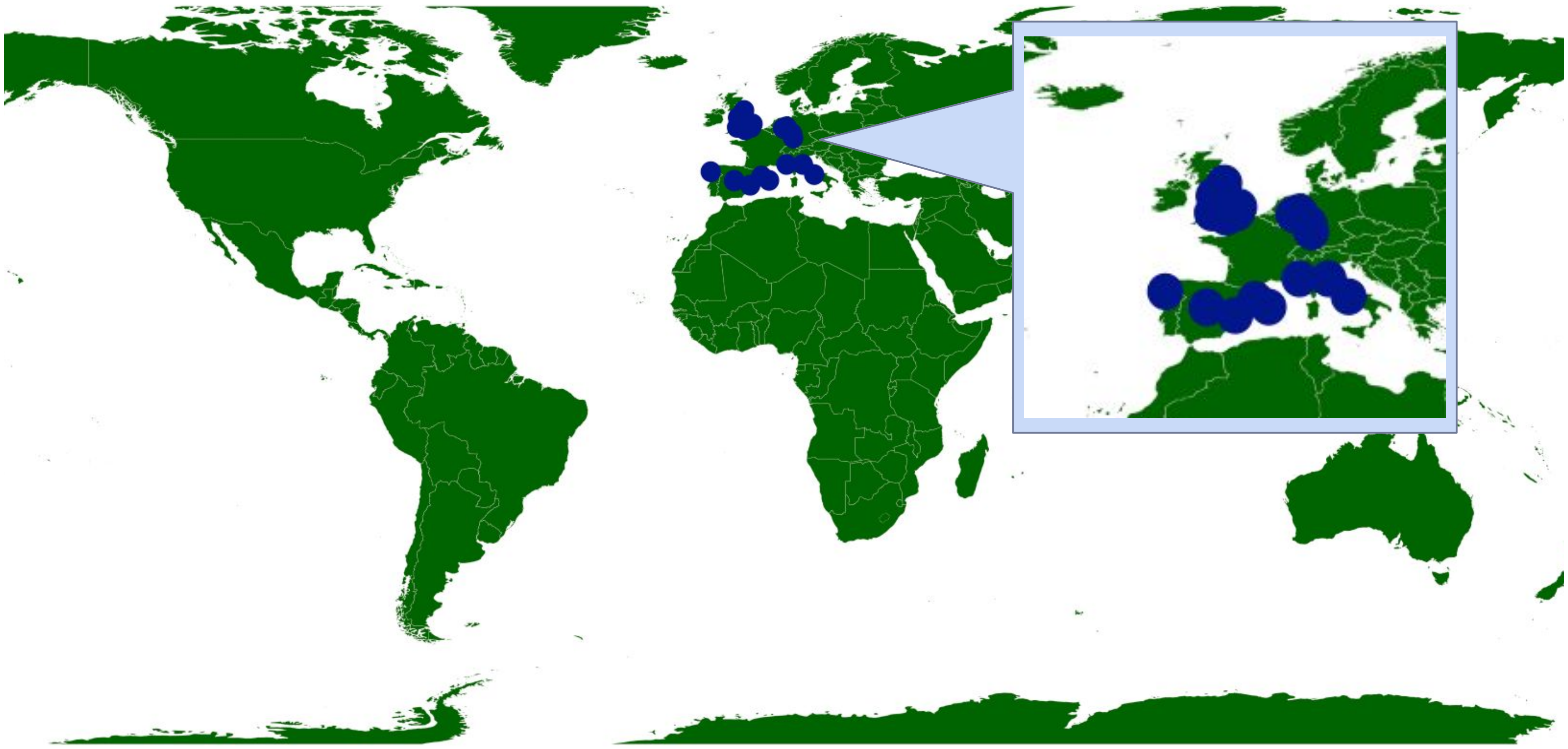
Player Age vs Wage colored by Continent
L. Messi
C. Ronaldo
Wage (thousands of Euros per week)
0
200
400
20
30
40
Age
factor(Continent)
Africa
Asia
Europe
North America
Oceania
South America


Player Wage vs Overall Skill Rating
Wage (thousands of Euros per week)
400
200
0
50
60
70
80
90
Overall
Club
Arsenal
AS Monaco
Atlético Madrid
Borussia Dortmund
Burnley
Cardiff City
CD Leganés
Chelsea
Eintracht Frankfurt
Empoli
Everton
FC Barcelona
Fortuna Düsseldorf
Frosinone
Liverpool
Manchester City
Manchester United
Newcastle United
Rayo Vallecano
RC Celta
Real Madrid
Southampton
Tottenham Hotspur
TSG 1899 Hoffenheim
Valencia CF
Wolverhampton Wanderers
Other



Wage Density by Club
density
3
2
1
0
0
2
4
6
Log Wage (thousands of Euros per week)
Club
Arsenal
AS Monaco
Atlético Madrid
Borussia Dortmund
Burnley
Cardiff City
CD Leganés
Chelsea
Eintracht Frankfurt
Empoli
Everton
FC Barcelona
Fortuna Düsseldorf
Frosinone
Liverpool
Manchester City
Manchester United
Newcastle United
Rayo Vallecano
RC Celta
Real Madrid
Southampton
Tottenham Hotspur
TSG 1899 Hoffenheim
Valencia CF
Wolverhampton Wanderers
Other

Top 28 Club Locations



Data Cleaning

Data Cleaning

- Remove unneeded columns
  - Row #, ID #, Value, Release Clause, Name, Photo, Flag, Club Logo, Special, Body Type, Real Face, Jersey #, Loaned From, Joined Date
- Removed symbols/letters in numeric columns
  - Wage & weight
  - Height to inches function removes apostrophe, replaces it with decimal and changes it to inches 5'11 to 71 inches
- Factoring
  - Over 200 countries, so we changed countries to continents by a right_join
  - Factored clubs into top 28 & other

Models

Linear Regression, Linear Regression with Lasso Variable Selection, Elastic Net, Random Forest, Neural Network

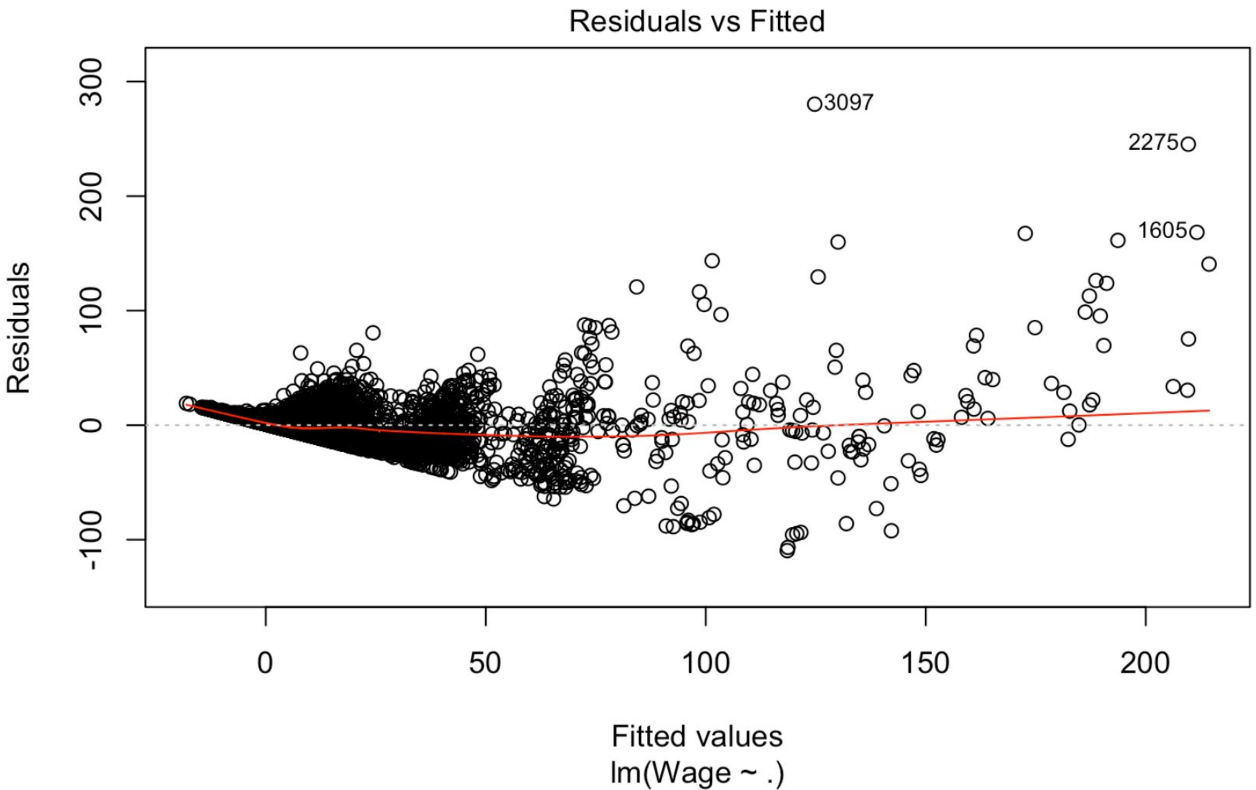
Multiple Linear Regression

```
fifa_lm_mod <- lm(Wage ~ .,
                  fifa_train)
preds_DF_lm <- data.frame(fifa_lm_preds = predict(fifa_lm_mod, newdata=fifa_test))
postResample(preds_DF_lm$fifa_lm_preds,fifa_test$Wage)
```

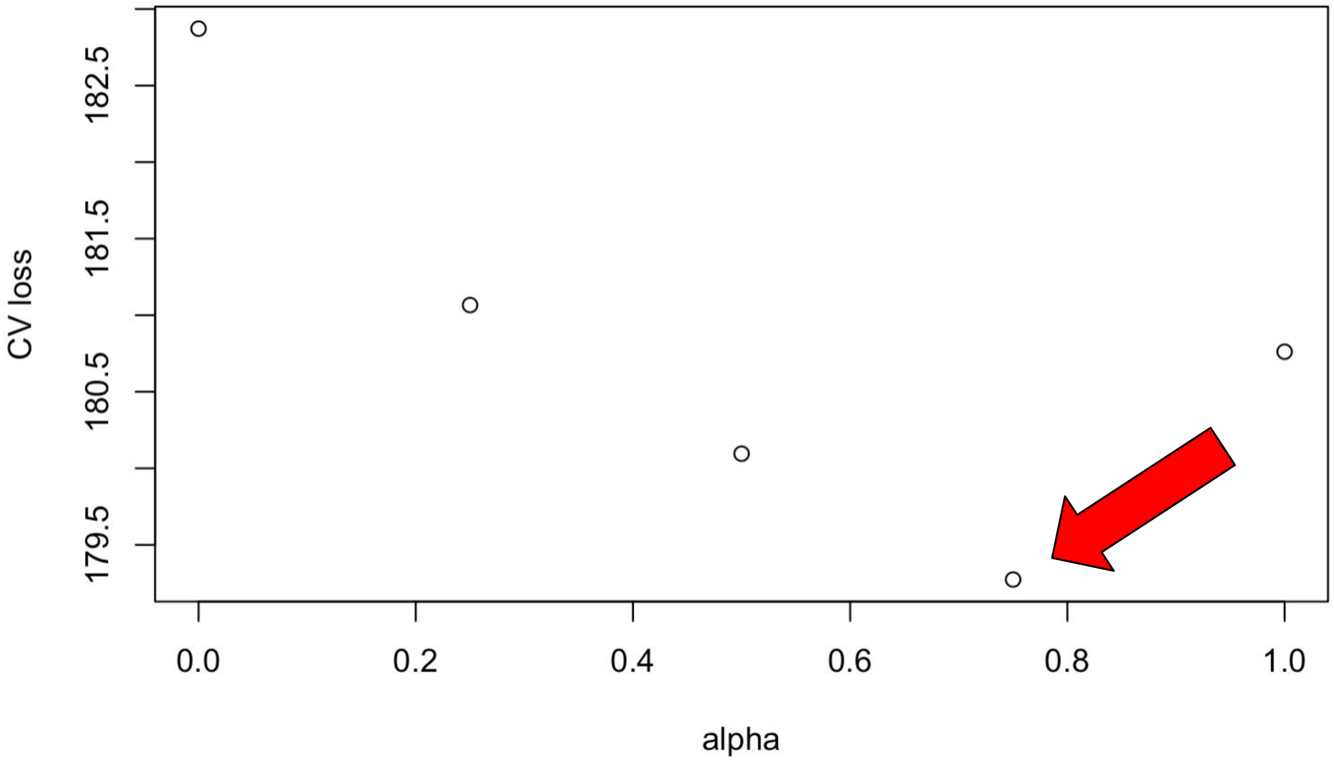
Linear Model Diagnostics

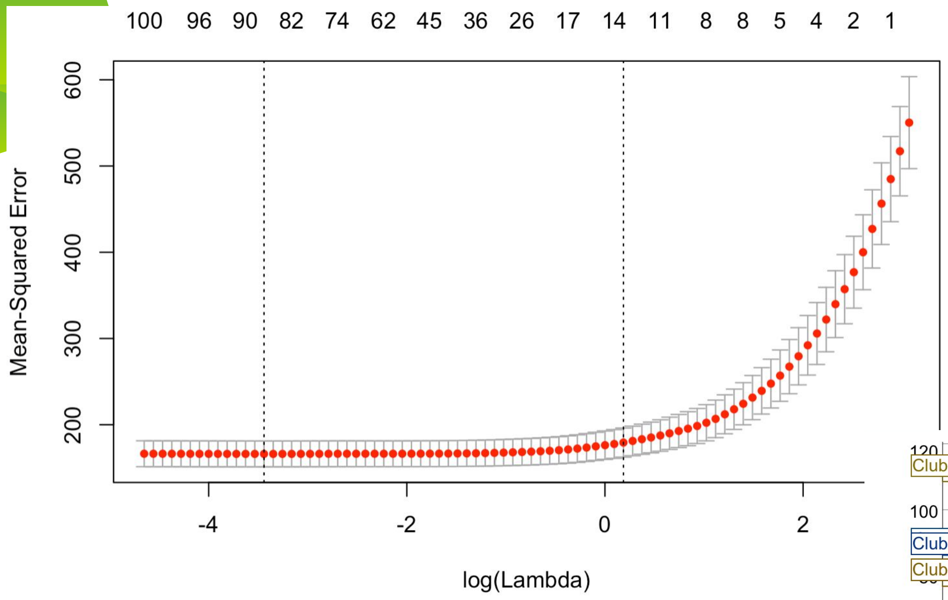
- Better than expected!
- R2: 0.66
- MAE: 6.4
- RMSE: 13.8

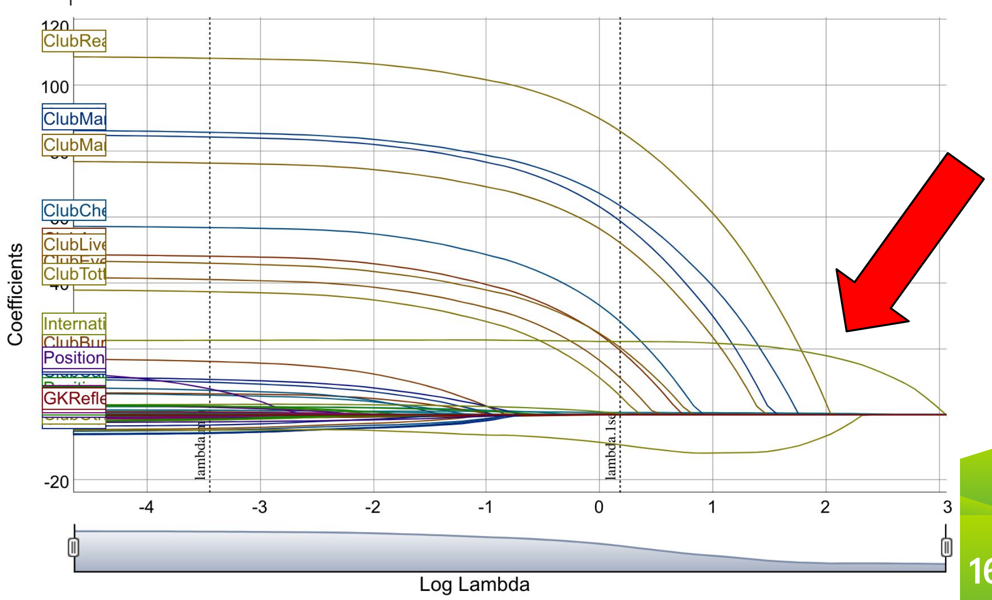
Heteroskedasticity



Elastic Net: Picking our Optimal Alpha







- *Clubs are some of the last variables to be set to zero*
- *International reputation last variable set to zero*

Elastic Net Model Diagnostics

- R2: 0.63
- MAE: 6.3
- RMSE: 14.5

Lasso Model

```
lasso_mod_fifa <- cv.glmnet(Wage ~ .,
                       data = fifa_train,
                       alpha = 1,
                       nfolds = 10)

fifa_lm_mod_lassopicks <- lm(Wage ~ International.Reputation + Overall + Potential + Club,
                             fifa_train)
preds_DF_lm_lassopicks <- data.frame(fifa_lm_preds = predict(fifa_lm_mod_lassopicks, newdata=fifa_test))
postResample(preds_DF_lm_lassopicks$fifa_lm_preds,fifa_test$Wage)
```

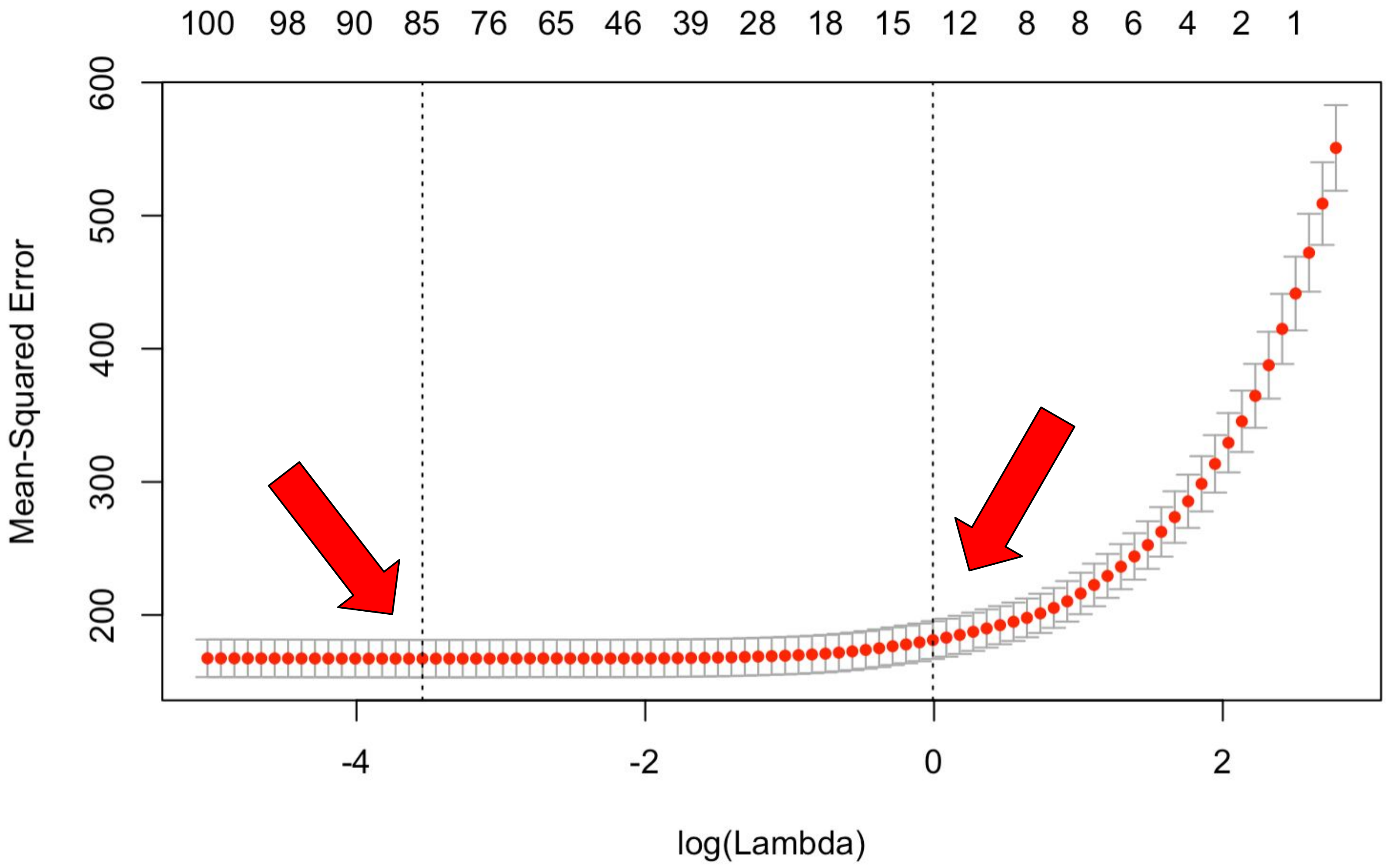
*Please ignore our terrible naming conventions

Lasso Model

- Lambda min picked 96 variables
- Lambda 1se picked 8 variables:
  - International.Reputation
  - Overall
  - Potential
  - 4 clubs and “Other” club factor

Lasso Model Diagnostics

- R2: 0.65
- MAE: 6.3
- RMSE: 14


100 98 90 85 76 65 46 39 28 18 15 12 8 8 6 4 2 1
600
500
400
300
200
Mean-Squared Error
-4
-2
0
2
log(Lambda)

Post Lasso Estimator

- We used the variable selections from Lasso (lambda.1se) to make a linear model to see how it performs compared to one with all variables

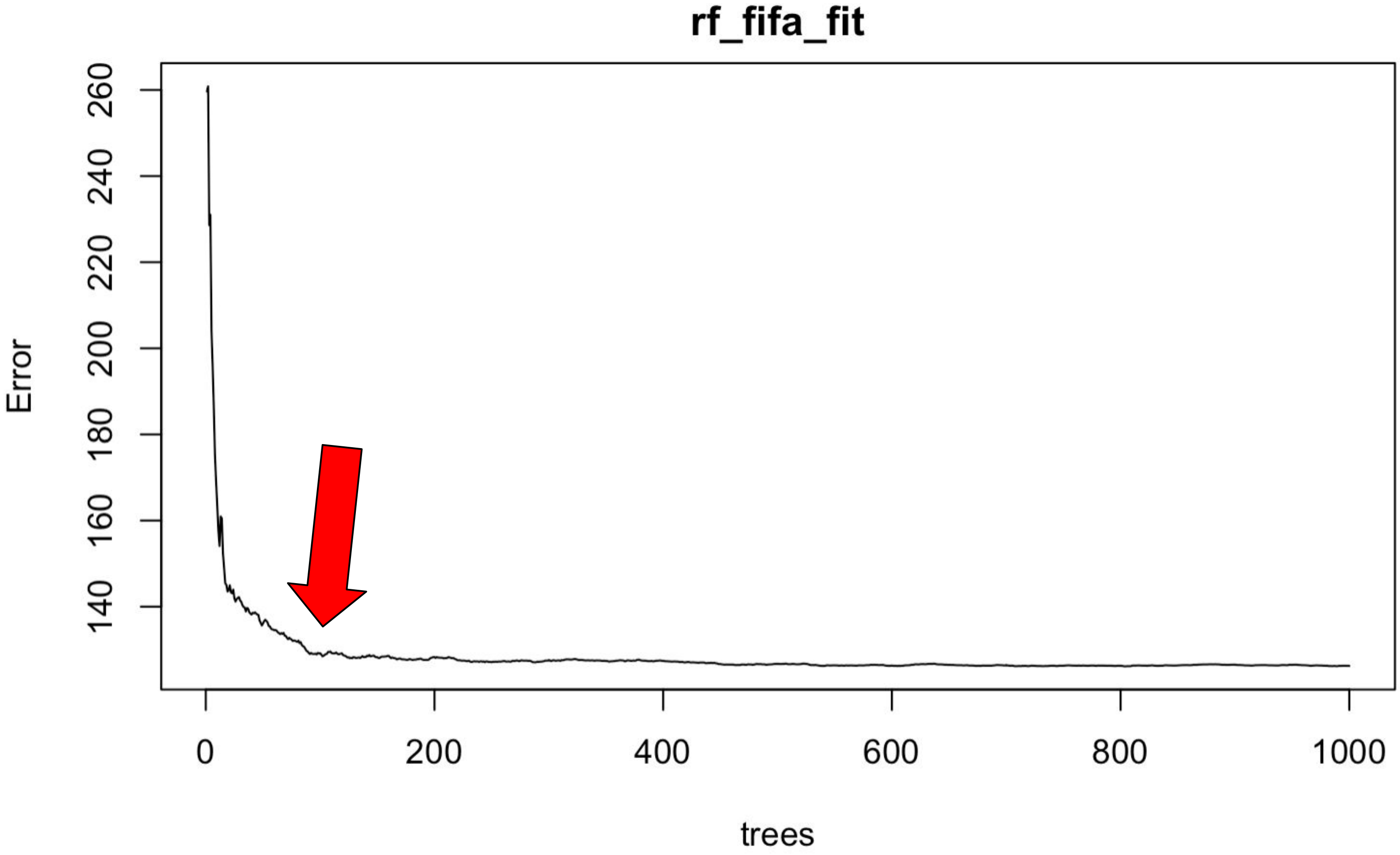
Post Lasso Estimator

- LM with Lasso variables
  - R2: 0.649
  - RMSE: 14.04
  - MAE: 6.27
- Regular LM
  - R2: 0.659
  - RMSE: 12.84
  - MAE: 6.39
- Those 8 variables do almost as well as a model using all the variables

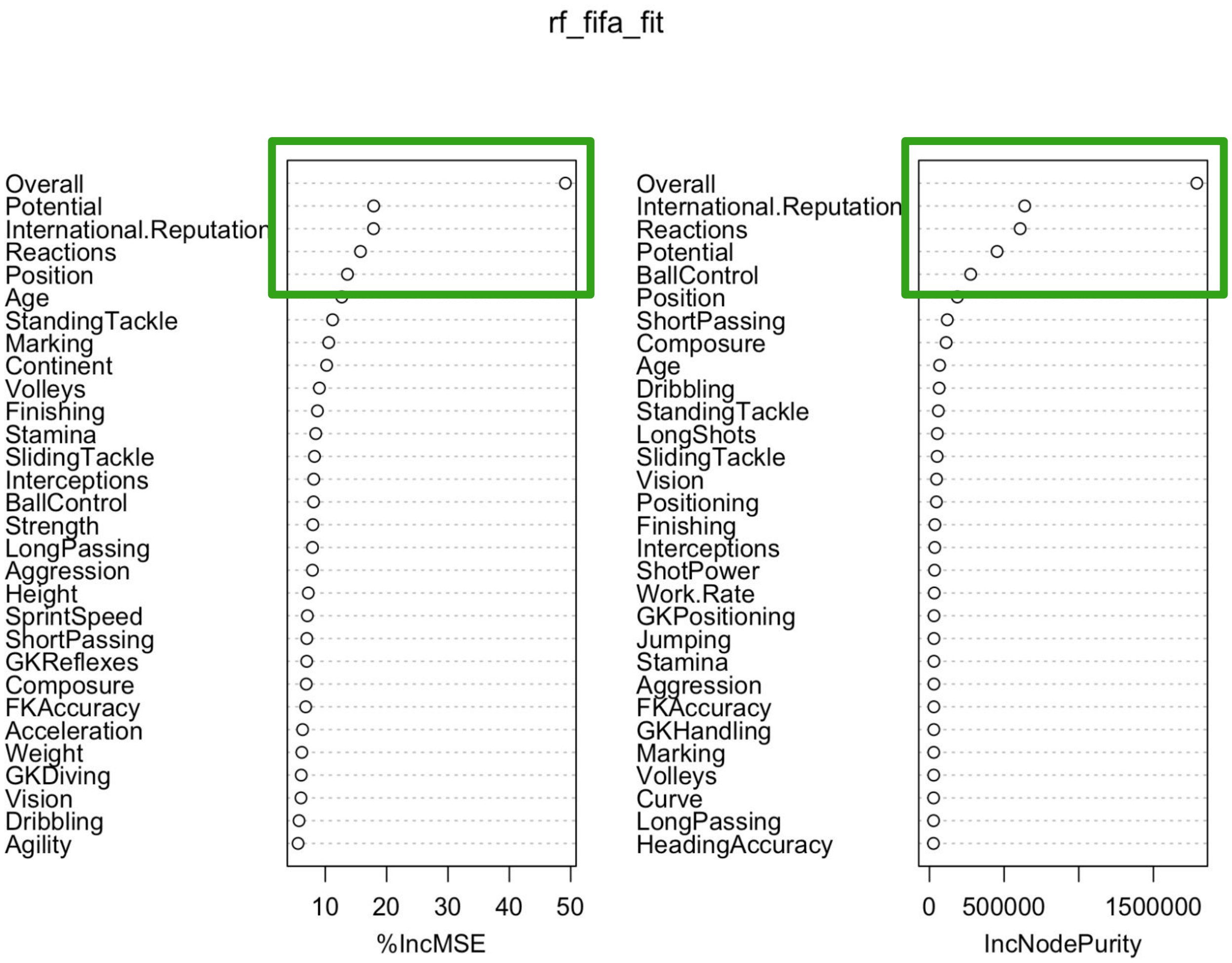
Random Forest

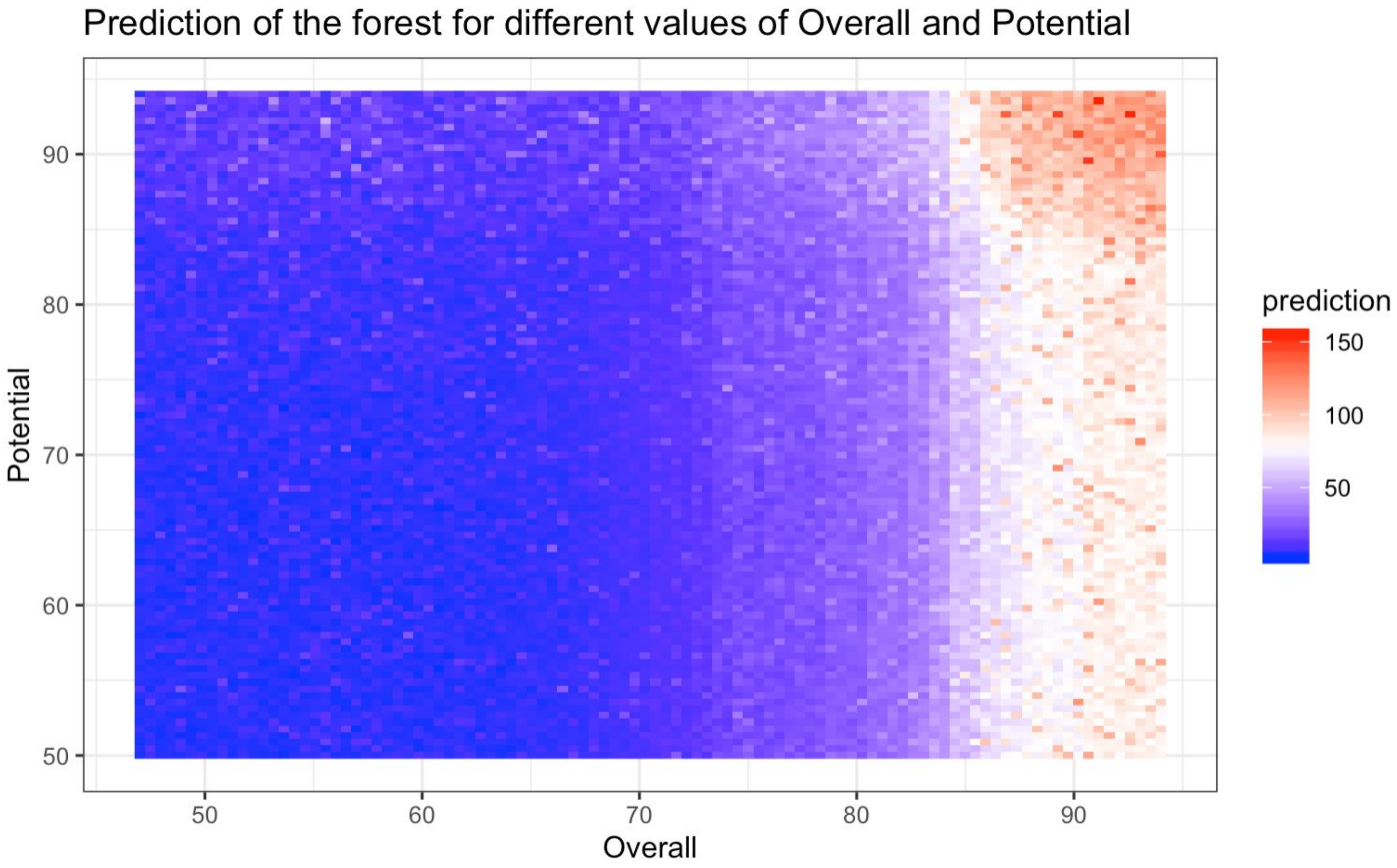
- Used everything but Club because it had too many categories
- Chose 100 trees to minimize error

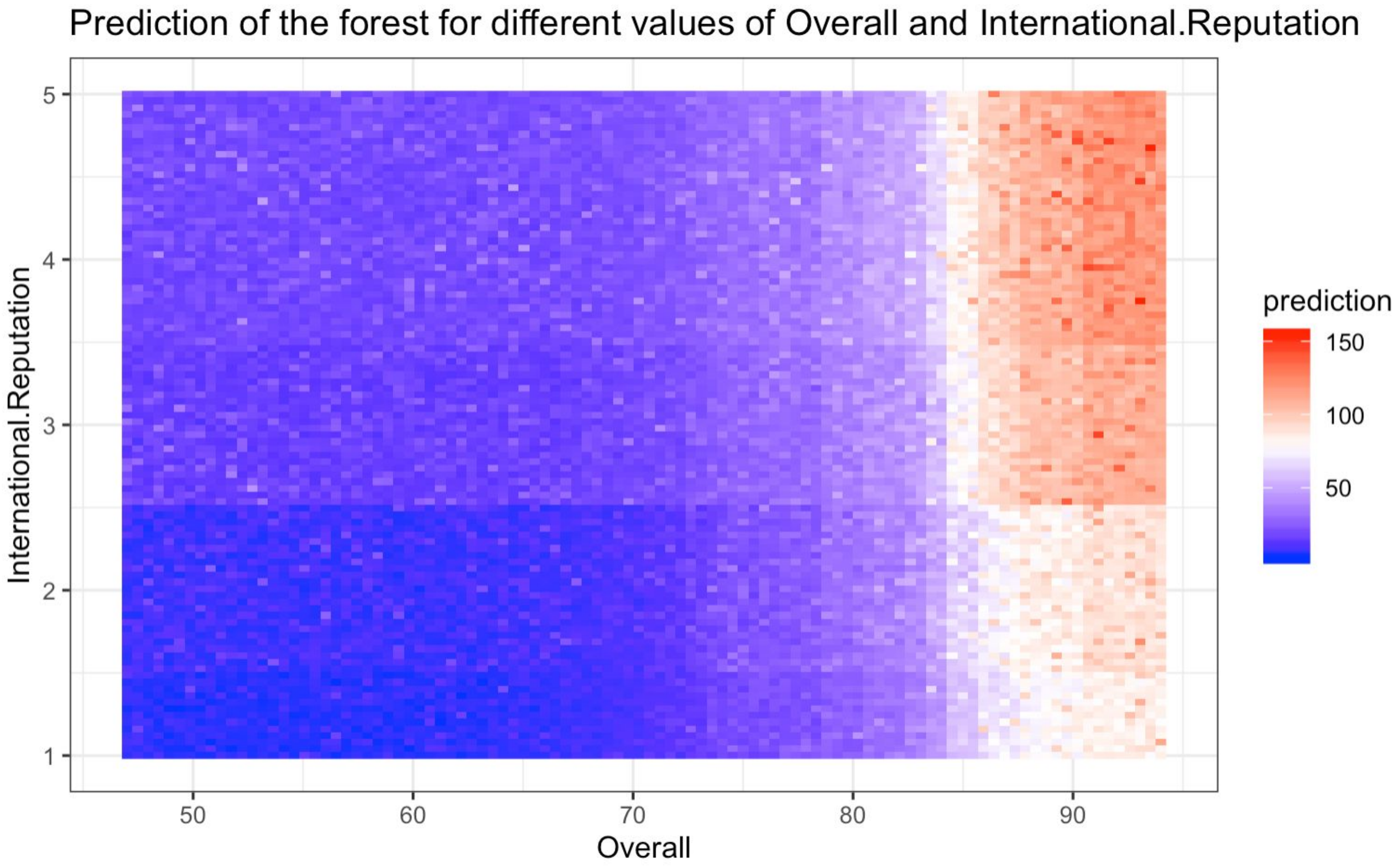
```r
fifa_train_noclub <- fifa_train %>% select(-"Club")

rf_fifa_fit <- randomForest(Wage ~.,
                            data = fifa_train_noclub,
                            type = regression,
                            ntree = 100,
                            importance = TRUE,
                            localImp = TRUE
                            )
```


rf_fifa_fit
Error
260
240
220
200
180
160
140
0
200
400
600
800
1000
trees

Top 5 variables




Prediction of the forest for different values of Overall and Potential
Potential
90
80
70
60
50
50
60
70
80
90
Overall
prediction
150
100
50



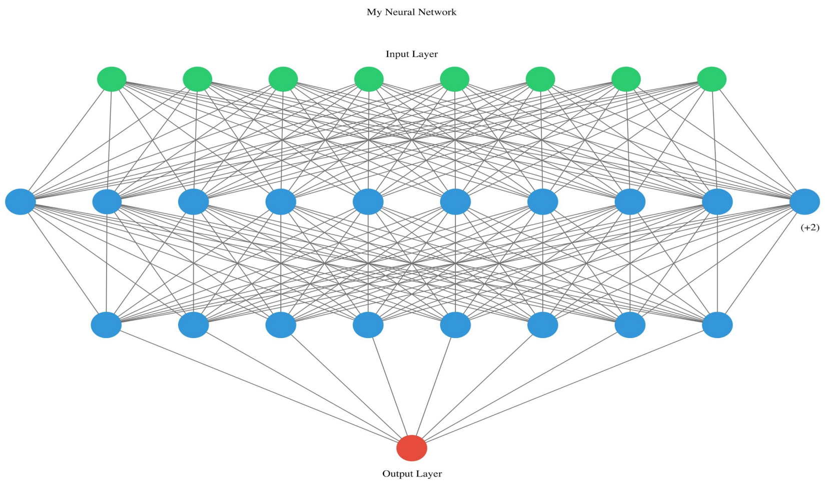
Prediction of the forest for different values of Overall and International.Reputation
International.Reputation
Overall
prediction
150
100
50
1
2
3
4
5
50
60
70
80
90

Random Forest Model Diagnostics

- R2: 0.79
- MAE: 4.5
- RMSE: 11.3

Neural Network

- Feed input into network
- Make prediction
- Update weights through backpropagation



Neural Network

```
import pandas as pd
import keras
import numpy as np
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
x_train = pd.read_csv("datasets/fifa_xtrain.csv")
x_test = pd.read_csv("datasets/fifa_xtest.csv")
y_train = pd.read_csv("datasets/fifa_ytrain.csv")
y_test = pd.read_csv("datasets/fifa_ytest.csv")
```

```
Using TensorFlow backend.
```

```
y_train = (y_train - y_train.mean()) / y_train.std()
x_train = (x_train - x_train.mean()) / x_train.std()
y_test = (y_test - y_test.mean()) / y_test.std()
x_test = (x_test - x_test.mean()) / x_test.std()
```

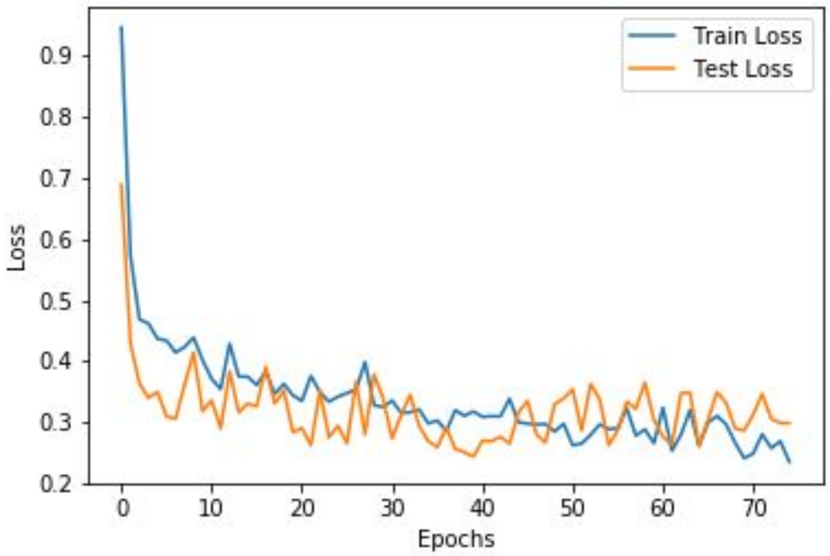
```
print(x_train.shape)
print(x_test.shape)
```

```
(9737, 42)
(4173, 42)
```

```
model = Sequential()
model.add(Dense(42, input_dim = 42, kernel_initializer = 'normal', activation='relu'))
model.add(Dense(21, kernel_initializer='normal', activation = 'relu'))
model.add(Dense(10, kernel_initializer='normal', activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(5, kernel_initializer='normal', activation = 'relu'))
model.add(Dense(1, kernel_initializer='normal'))
model.compile(loss='mean_squared_error', optimizer='adam', metrics = ['mape'])
```

```
history = model.fit(
    x_train, y_train,    # training data to learn from
    batch_size= 107, # size of batches
    epochs= 75, # how many iterations we train for
    verbose=1,# type of logging
    validation_data=(x_test, y_test))
```

Neural Network Diagnostics

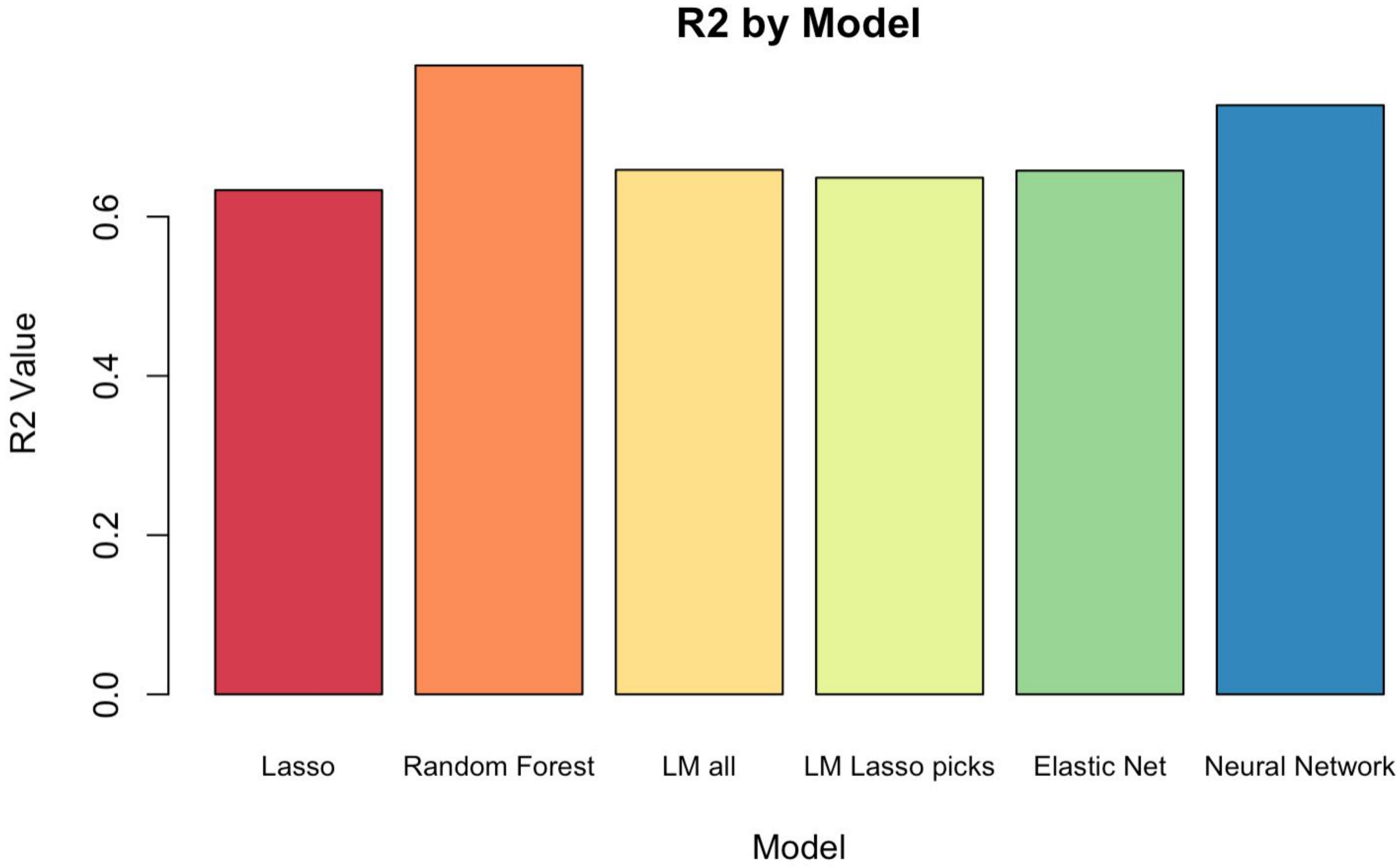


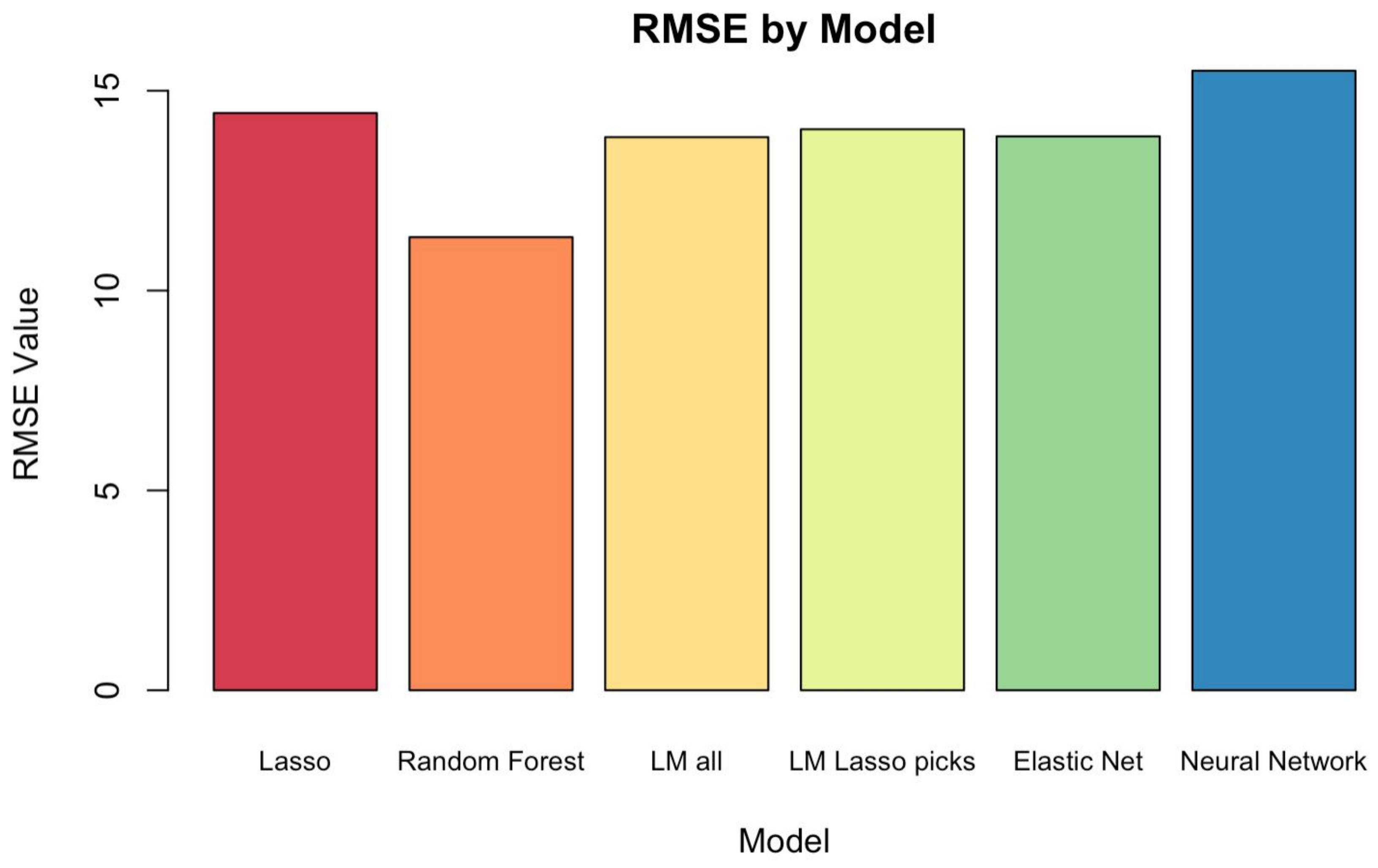
- R2: 0.74

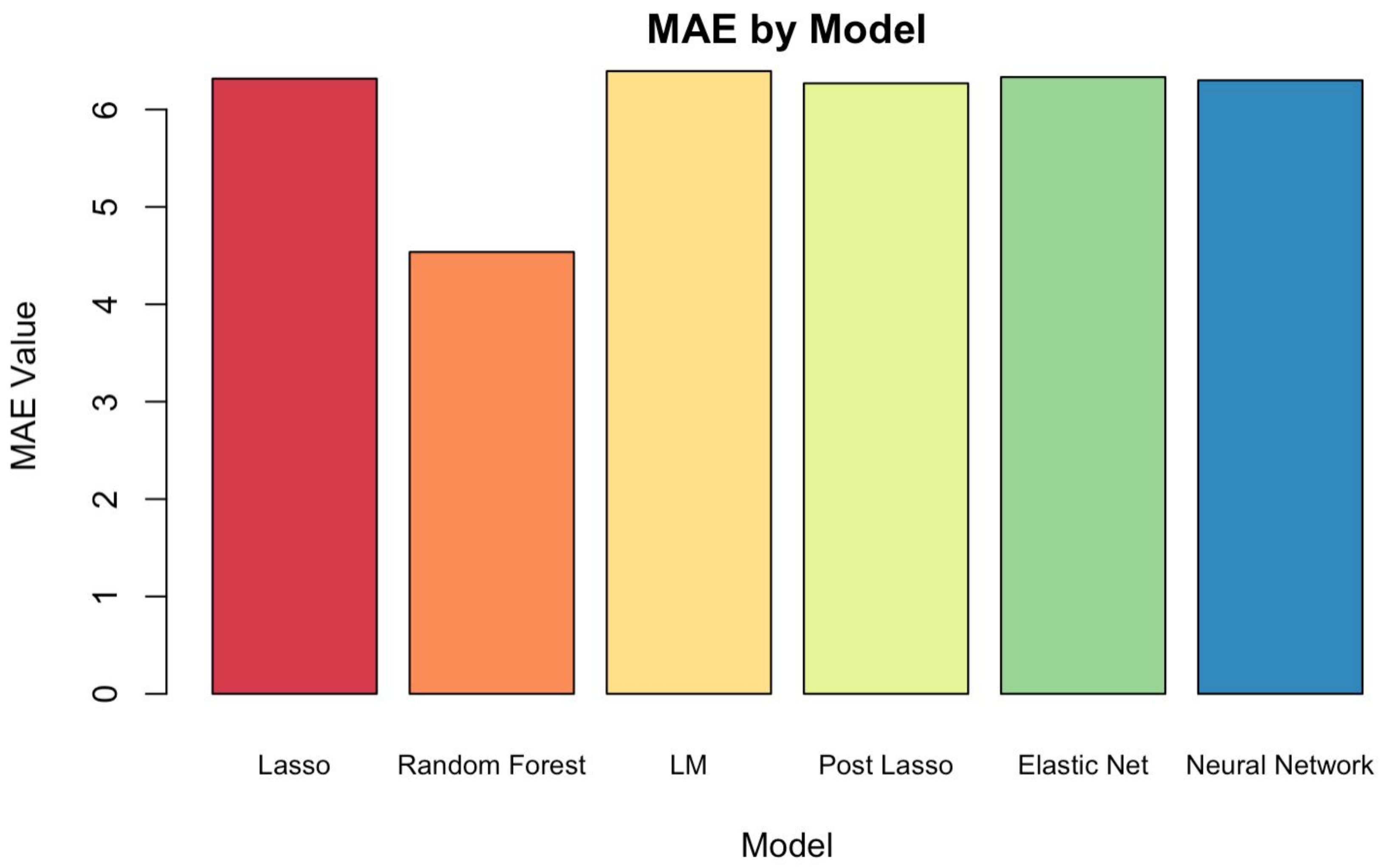
Results

Results

- Most important variables across all models:
  - Overall skill rating
  - Potential rating
  - Club
  - International Reputation
  - Reactions
  - Position


R2 by Model
R2 Value
0.0
0.2
0.4
0.6
Lasso
Random Forest
LM all
LM Lasso picks
Elastic Net
Neural Network
Model


RMSE by Model
RMSE Value
0
5
10
15
Lasso
Random Forest
LM all
LM Lasso picks
Elastic Net
Neural Network
Model


MAE by Model
MAE Value
0
1
2
3
4
5
6
Lasso
Random Forest
LM
Post Lasso
Elastic Net
Neural Network
Model

Conclusions

Conclusion

- We learned that a few variables matter a lot, and everything else doesn't really have an impact
- Management can maximize their market cap use by hiring someone highly skilled in a low-earning position that can help the team a lot but won't cost as much
- Teams with lower international reputations can hire highly skilled players just under a rating of 80

Thank you! Questions?