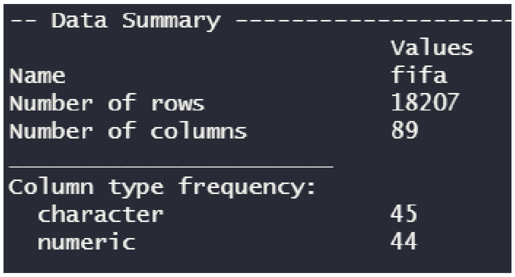
Brief description of the problem and goal of the project

Sports industry has realized the role analytics can play to help them make better decisions all around. Lately, clubs have ended up paying exorbitant fees to acquire the services of a player and thus end up with huge amount of debts and that's where predictive analytics come into play. A solid predictive model which can accurately predict the player's correct value can help the clubs to a large extent. Predicting the value of the players based on various parameters like, Age, Wage, validity of the player's contract, etc. This can help football clubs make better decisions in terms of buying players, giving them a better business model, economic confidence, and a better approach to deal with the negotiations.

Data Description

The dataset I have used to create a predictive model includes the latest edition of FIFA 2019. It consists of player attributes like Age, Nationality, Overall, Potential, Club, Value, Wage, Preferred Foot, International Reputation, Weak Foot, Skill Moves, Work Rate, Position, Jersey Number, Joined, Loaned From, Contract Valid Until, Height, Weight, Position, etc. This data is collected and updated regularly by SOFIFA. In total, there are 89 variables and 18,000 observations.



```
-- Data Summary --------------------
                           Values
Name                       fifa
Number of rows             18207
Number of columns          89
_______________________
Column type frequency:
  character                45
  numeric                  44
```

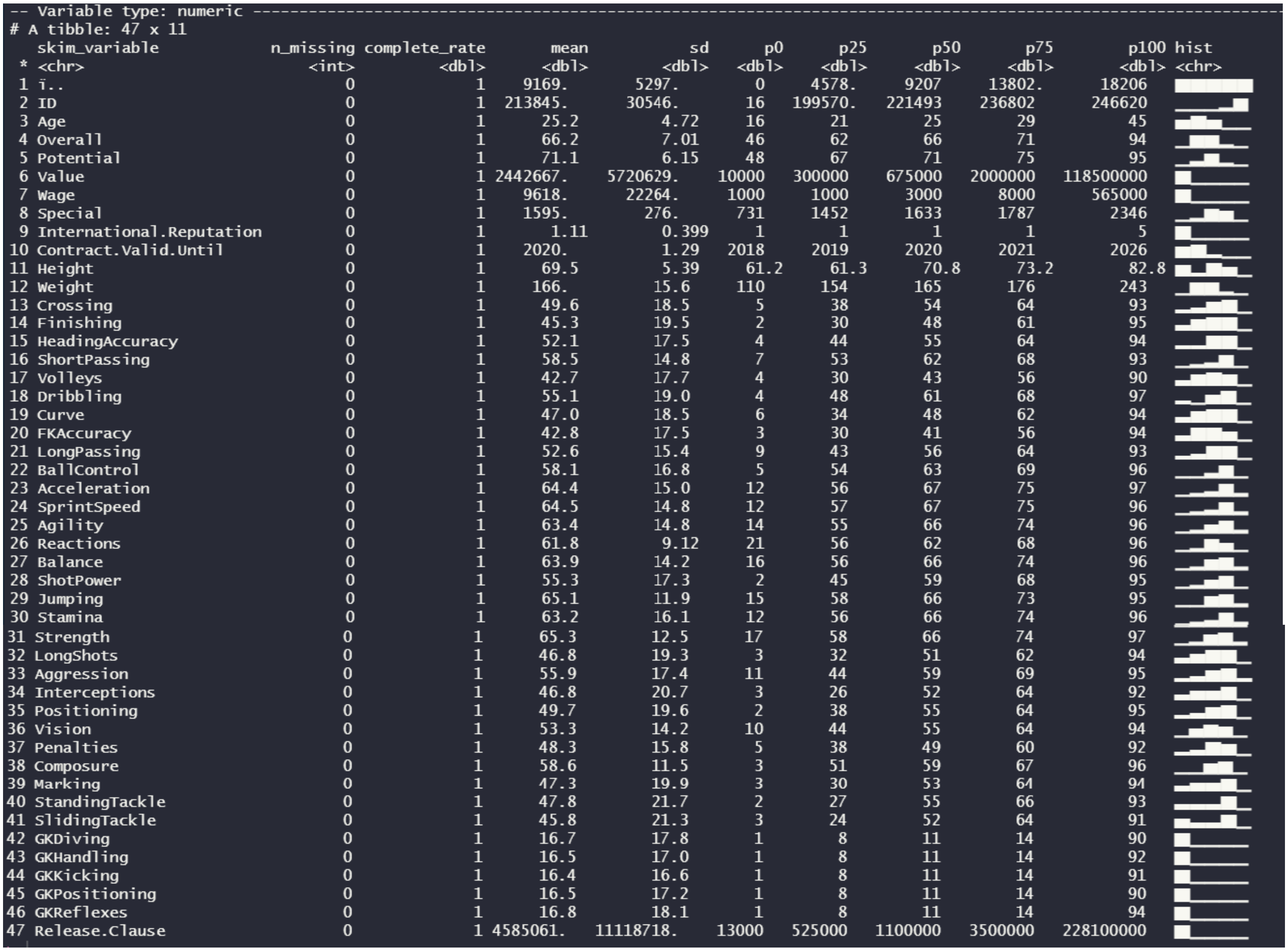
Source: https://www.kaggle.com/karangadiya/fifa19

The important variables which can help in accurately predicting a player's value are:

- Age: Player's age
- Wage: Weekly earnings of a player
- Overall: Player's footballing ability on a scale of 0 to 100
- Potential: Player's footballing potential on a scale of 0 to 100
- Contract.Valid.Until: Year till the player's contract is valid
- Release.Clause: Amount needed to buy that player
- Position: The position in which the player is deployed
- Height: Height of the player
- Weight: Weight of the player

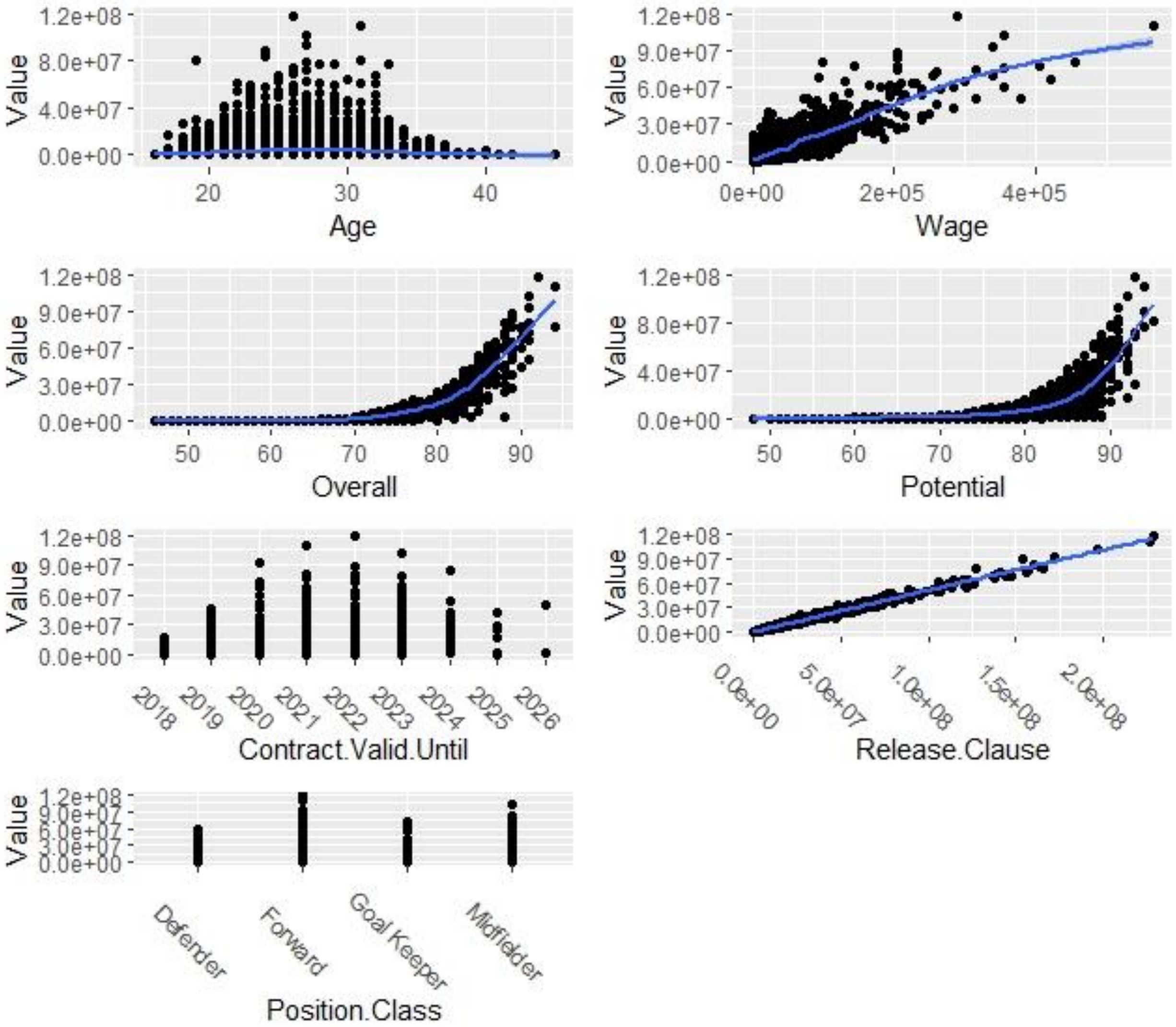
To obtain the best analysis, a thorough cleaning of the data was necessary. All the redundant variables, NA values, if any, were removed. Columns like 'Value', 'Wage', 'Height', 'Weight', and 'Release.Clause' were modified to make them more accessible to our analysis. Values in the 'Positions' variable were clubbed together to leave us with three class of positions, namely, Forward, Midfielder, Defender and Goalkeeper. A new variable was created to contain these class of positions.

The final outcome of data cleaning was as follows:



```
-- Variable type: numeric ----------------------------------------------------------------------------------------------
# A tibble: 47 x 11
   skim_variable           n_missing complete_rate      mean         sd     p0     p25     p50     p75      p100 hist
 * <chr>                       <int>         <dbl>     <dbl>      <dbl>  <dbl>   <dbl>   <dbl>   <dbl>     <dbl> <chr>
 1 ï..                             0             1    9169.       5297.      0   4578.    9207  13802.     18206
 2 ID                              0             1  213845.      30546.     16 199570.  221493  236802    246620
 3 Age                             0             1      25.2       4.72     16      21      25      29        45
 4 Overall                         0             1      66.2       7.01     46      62      66      71        94
 5 Potential                       0             1      71.1       6.15     48      67      71      75        95
 6 Value                           0             1 2442667.    5720629.  10000  300000  675000 2000000 118500000
 7 Wage                            0             1    9618.      22264.   1000    1000    3000    8000    565000
 8 Special                         0             1    1595.        276.    731    1452    1633    1787      2346
 9 International.Reputation        0             1       1.11      0.399     1       1       1       1         5
10 Contract.Valid.Until            0             1    2020.        1.29   2018    2019    2020    2021      2026
11 Height                          0             1      69.5       5.39   61.2    61.3    70.8    73.2      82.8
12 Weight                          0             1     166.        15.6    110     154     165     176       243
13 Crossing                        0             1      49.6      18.5       5      38      54      64        93
14 Finishing                       0             1      45.3      19.5       2      30      48      61        95
15 HeadingAccuracy                 0             1      52.1      17.5       4      44      55      64        94
16 ShortPassing                    0             1      58.5      14.8       7      53      62      68        93
17 Volleys                         0             1      42.7      17.7       4      30      43      56        90
18 Dribbling                       0             1      55.1      19.0       4      48      61      68        97
19 Curve                           0             1      47.0      18.5       6      34      48      62        94
20 FKAccuracy                      0             1      42.8      17.5       3      30      41      56        94
21 LongPassing                     0             1      52.6      15.4       9      43      56      64        93
22 BallControl                     0             1      58.1      16.8       5      54      63      69        96
23 Acceleration                    0             1      64.4      15.0      12      56      67      75        97
24 SprintSpeed                     0             1      64.5      14.8      12      57      67      75        96
25 Agility                         0             1      63.4      14.8      14      55      66      74        96
26 Reactions                       0             1      61.8       9.12     21      56      62      68        96
27 Balance                         0             1      63.9      14.2      16      56      66      74        96
28 ShotPower                       0             1      55.3      17.3       2      45      59      68        95
29 Jumping                         0             1      65.1      11.9      15      58      66      73        95
30 Stamina                         0             1      63.2      16.1      12      56      66      74        96
31 Strength                        0             1      65.3      12.5      17      58      66      74        97
32 LongShots                       0             1      46.8      19.3       3      32      51      62        94
33 Aggression                      0             1      55.9      17.4      11      44      59      69        95
34 Interceptions                   0             1      46.8      20.7       3      26      52      64        92
35 Positioning                     0             1      49.7      19.6       2      38      55      64        95
36 Vision                          0             1      53.3      14.2      10      44      55      64        94
37 Penalties                       0             1      48.3      15.8       5      38      49      60        92
38 Composure                       0             1      58.6      11.5       3      51      59      67        96
39 Marking                         0             1      47.3      19.9       3      30      53      64        94
40 StandingTackle                  0             1      47.8      21.7       2      27      55      66        93
41 SlidingTackle                   0             1      45.8      21.3       3      24      52      64        91
42 GKDiving                        0             1      16.7      17.8       1       8      11      14        90
43 GKHandling                      0             1      16.5      17.0       1       8      11      14        92
44 GKKicking                       0             1      16.4      16.6       1       8      11      14        91
45 GKPositioning                   0             1      16.5      17.2       1       8      11      14        90
46 GKReflexes                      0             1      16.8      18.1       1       8      11      14        94
47 Release.Clause                  0             1 4585061.   11118718.  13000  525000 1100000 3500000 228100000
```

Post data cleaning, exploratory data analysis was done to identify pattern between predictor variable and response variables. For this, scatterplots were plotted for the **Value** variable with each of the important response variables.



These scatterplots and the histograms above provided enough information to identify what kind of analysis was required and why.

Pre-analysis plan

To go ahead with our analysis, the dataset needed to be split into test and train data. This was done using the 70:30 ratio and setting the seed to 50.
The models chosen for the analysis were:

- Multiple Linear Regression Model
- General Additive Model
- Decision Tree Model
- Random Forest Model
- Generalized Boosted Model

Multiple Linear Regression Model helps us to understand how much will the dependent variable change when we change the independent variables and to identify the significant variables so as to make a better regression in the later analysis.

General Additive Model was chosen because now that we understand the significant independent variables and their relationship with the dependent variable, we can tailor each independent variable to better fit with the dependent variable.

Decision Tree Model provides a comprehensive analysis of the consequences along each branch and identifies decision nodes that need further analysis. Due to its comprehensive analysis nature, this model was used to provide a better model to predict the outcome.

Random Forest Model was used to get an even better fit since it uses multiple decision trees. It combines the output of multiple decision trees to generate the final output.

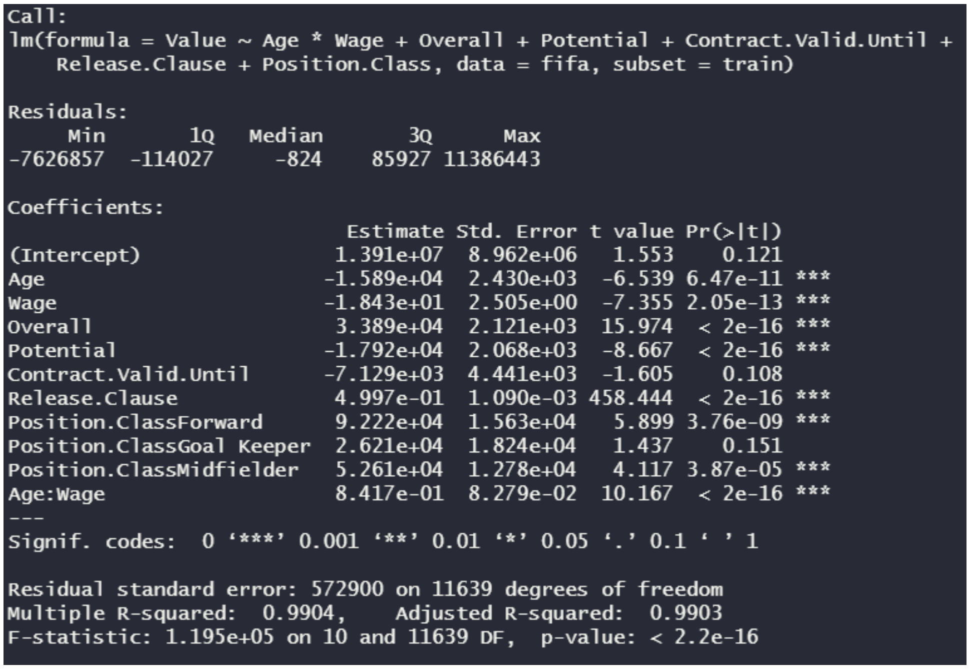
Generalized Boosted Model also use decision trees but unlike random forests, it repeatedly fits many decision trees to provide a better output.

The Cross-Validation approach used in our analysis is **Root Mean Squared Error (RMSE).** The model with the least RSME will be selected as the best model to predict the outcome.

Presentation of results

Explanation of models used and their results.

Model 1: Multiple Linear Regression:

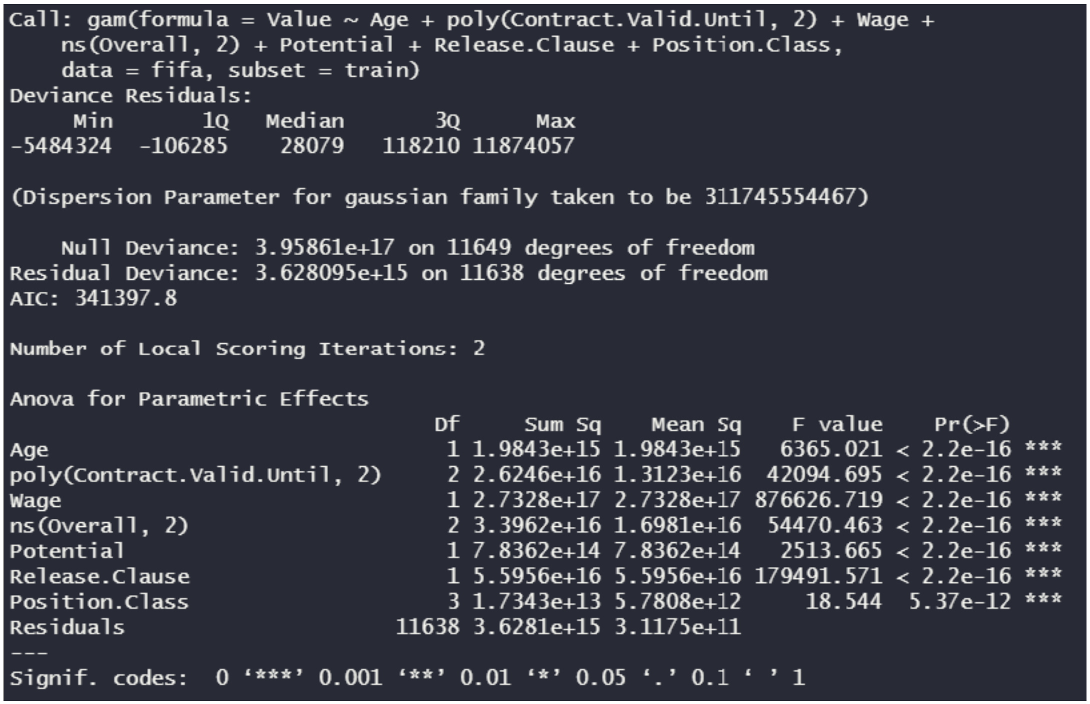


```
Call:
lm(formula = Value ~ Age * Wage + Overall + Potential + Contract.Valid.Until +
    Release.Clause + Position.Class, data = fifa, subset = train)

Residuals:
     Min       1Q   Median       3Q      Max
-7626857  -114027     -824    85927 11386443

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               1.391e+07  8.962e+06   1.553    0.121
Age                      -1.589e+04  2.430e+03  -6.539 6.47e-11 ***
Wage                     -1.843e+01  2.505e+00  -7.355 2.05e-13 ***
Overall                   3.389e+04  2.121e+03  15.974  < 2e-16 ***
Potential                -1.792e+04  2.068e+03  -8.667  < 2e-16 ***
Contract.Valid.Until     -7.129e+03  4.441e+03  -1.605    0.108
Release.Clause            4.997e-01  1.090e-03 458.444  < 2e-16 ***
Position.ClassForward     9.222e+04  1.563e+04   5.899 3.76e-09 ***
Position.ClassGoal Keeper 2.621e+04  1.824e+04   1.437    0.151
Position.ClassMidfielder  5.261e+04  1.278e+04   4.117 3.87e-05 ***
Age:Wage                  8.417e-01  8.279e-02  10.167  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 572900 on 11639 degrees of freedom
Multiple R-squared:  0.9904,    Adjusted R-squared:  0.9903
F-statistic: 1.195e+05 on 10 and 11639 DF,  p-value: < 2.2e-16
```

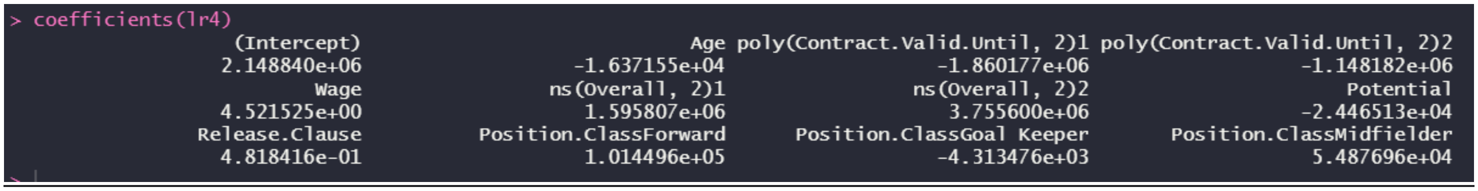
Initially, a simple multiple regression was run with all the important variables. All the insignificant variables were then dropped and then a polynomial regression with a polynomial function on 'Contract.Valid.Until' was run and we got a slightly better result. Finally an interaction variable was introduced, removing the polynomial function and this model produced even better result.

Comparing the three models, the interaction variable model turned out to be the best with **RMSE = 598511.6**

Model 2: General Additive Model



```
Call: gam(formula = Value ~ Age + poly(Contract.Valid.Until, 2) + Wage +
    ns(Overall, 2) + Potential + Release.Clause + Position.Class,
    data = fifa, subset = train)
Deviance Residuals:
     Min       1Q   Median       3Q      Max
-5484324  -106285    28079   118210 11874057

(Dispersion Parameter for gaussian family taken to be 311745554467)

    Null Deviance: 3.95861e+17 on 11649 degrees of freedom
Residual Deviance: 3.628095e+15 on 11638 degrees of freedom
AIC: 341397.8

Number of Local Scoring Iterations: 2

Anova for Parametric Effects
                                 Df     Sum Sq    Mean Sq    F value    Pr(>F)
Age                               1 1.9843e+15 1.9843e+15   6365.021 < 2.2e-16 ***
poly(Contract.Valid.Until, 2)     2 2.6246e+16 1.3123e+16  42094.695 < 2.2e-16 ***
Wage                              1 2.7328e+17 2.7328e+17 876626.719 < 2.2e-16 ***
ns(Overall, 2)                    2 3.3962e+16 1.6981e+16  54470.463 < 2.2e-16 ***
Potential                         1 7.8362e+14 7.8362e+14   2513.665 < 2.2e-16 ***
Release.Clause                    1 5.5956e+16 5.5956e+16 179491.571 < 2.2e-16 ***
Position.Class                    3 1.7343e+13 5.7808e+12     18.544  5.37e-12 ***
Residuals                     11638 3.6281e+15 3.1175e+11
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



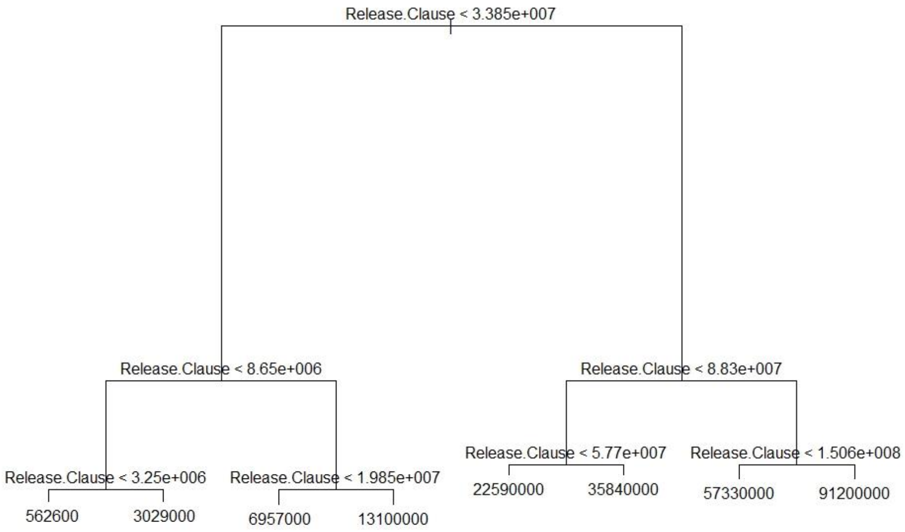
```
> coefficients(lr4)
                (Intercept)                         Age poly(Contract.Valid.Until, 2)1 poly(Contract.Valid.Until, 2)2
               2.148840e+06               -1.637155e+04                  -1.860177e+06                  -1.148182e+06
                       Wage              ns(Overall, 2)1                ns(Overall, 2)2                      Potential
               4.521525e+00                1.595807e+06                   3.755600e+06                  -2.446513e+04
             Release.Clause      Position.ClassForward        Position.ClassGoal Keeper      Position.ClassMidfielder
               4.818416e-01                1.014496e+05                  -4.313476e+03                   5.487696e+04
```

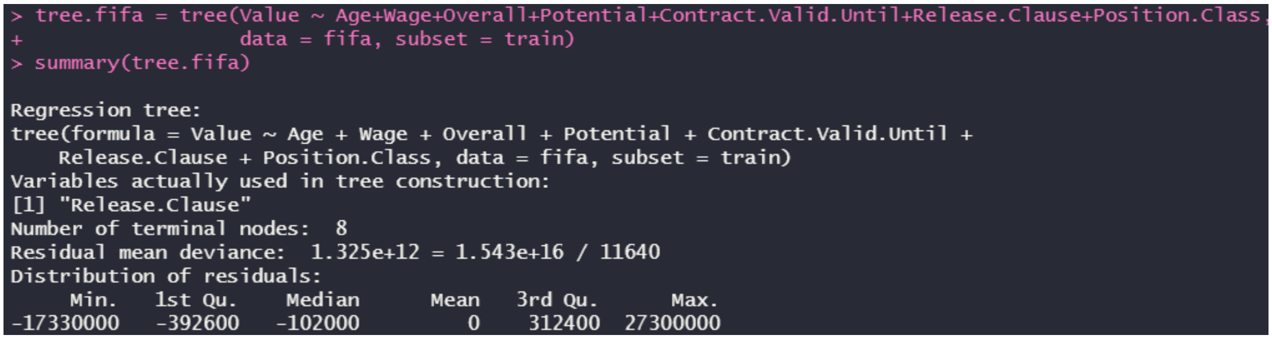
In this model, a polynomial function of 2 was added to 'Contracts.Valid.Until' and a natural cubic spline with degrees of freedom of 2 to 'Overall'**.** To get the accuracy of this model, we ran a test MSE.

The RSME score was 594049.3

This model performed better than the previous model but clearly there was room for improvement.

Model 3: Decision Trees


Release.Clause < 3.385e+007
Release.Clause < 8.65e+006
Release.Clause < 8.83e+007
Release.Clause < 3.25e+006
Release.Clause < 1.985e+007
Release.Clause < 5.77e+007
Release.Clause < 1.506e+008
562600
3029000
6957000
13100000
22590000
35840000
57330000
91200000

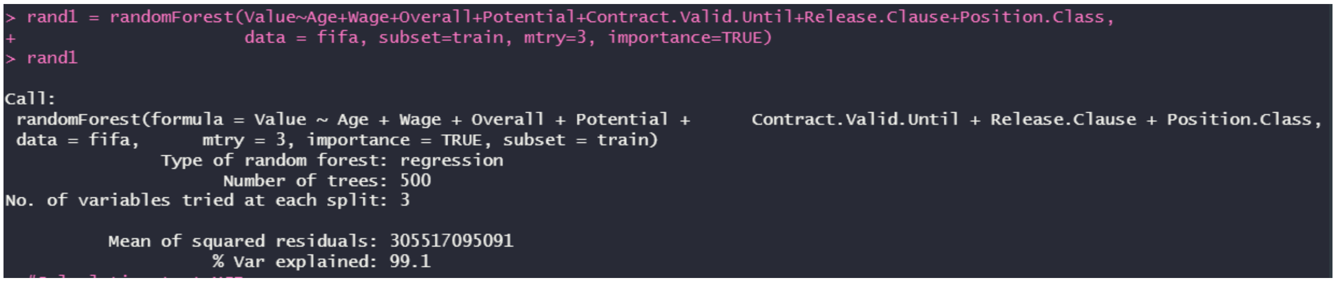


```
> tree.fifa = tree(Value ~ Age+Wage+Overall+Potential+Contract.Valid.Until+Release.Clause+Position.Class,
+                  data = fifa, subset = train)
> summary(tree.fifa)

Regression tree:
tree(formula = Value ~ Age + Wage + Overall + Potential + Contract.Valid.Until +
    Release.Clause + Position.Class, data = fifa, subset = train)
Variables actually used in tree construction:
[1] "Release.Clause"
Number of terminal nodes:  8
Residual mean deviance:  1.325e+12 = 1.543e+16 / 11640
Distribution of residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-17330000   -392600   -102000         0    312400  27300000
```

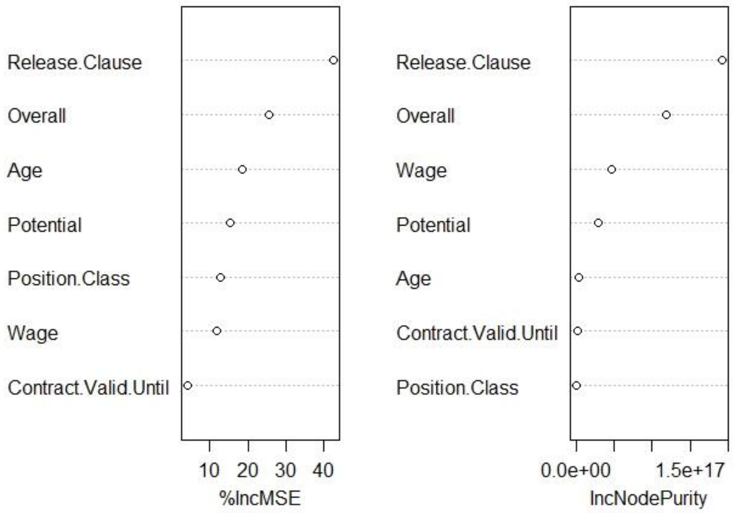
A decision tree was added with all the significant variables. The model used 'Release.Clause' as the only predictor variable to generate a tree.

The **RSME** obtained from this model was **1189180** which was considerably higher than the previous model and thus it was clearly not a good model for our analysis.

Model 4: Random Forest Model



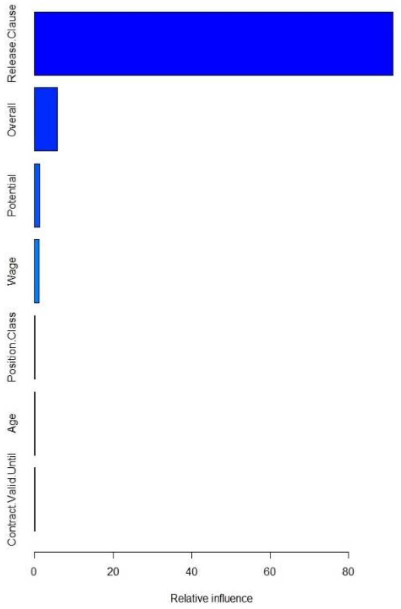
```
> rand1 = randomForest(Value~Age+Wage+Overall+Potential+Contract.Valid.Until+Release.Clause+Position.Class,
+                      data = fifa, subset=train, mtry=3, importance=TRUE)
> rand1

Call:
 randomForest(formula = Value ~ Age + Wage + Overall + Potential +      Contract.Valid.Until + Release.Clause + Position.Class,
 data = fifa,      mtry = 3, importance = TRUE, subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 305517095091
                    % Var explained: 99.1
```

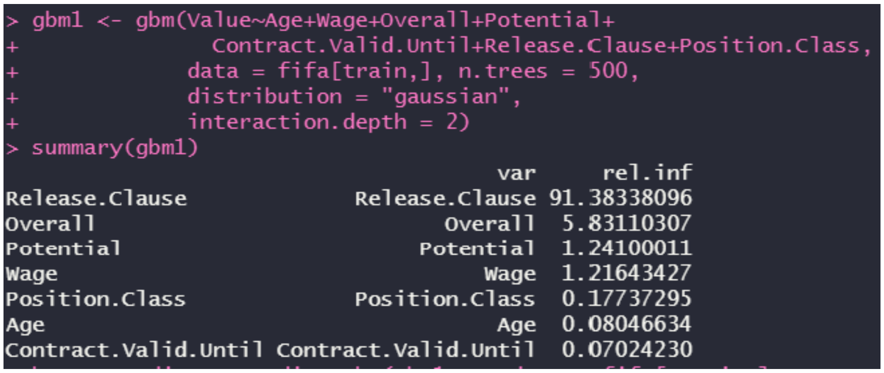


A random forest model was run with all the important variables, three predictor variables as candidates at each split and 500 number of trees. With this model, we were able to infer that Release.Clause, Overall and Wage were the most important variables.

The RSME for this model was **511268.2** which was far better than any of the previous models.

Model 5: Generalized Boosted Model





```
> gbm1 <- gbm(Value~Age+Wage+Overall+Potential+
+              Contract.Valid.Until+Release.Clause+Position.Class,
+            data = fifa[train,], n.trees = 500,
+            distribution = "gaussian",
+            interaction.depth = 2)
> summary(gbm1)
                                      var     rel.inf
Release.Clause             Release.Clause 91.38338096
Overall                           Overall  5.83110307
Potential                       Potential  1.24100011
Wage                                 Wage  1.21643427
Position.Class             Position.Class  0.17737295
Age                                   Age  0.08046634
Contract.Valid.Until Contract.Valid.Until  0.07024230
```

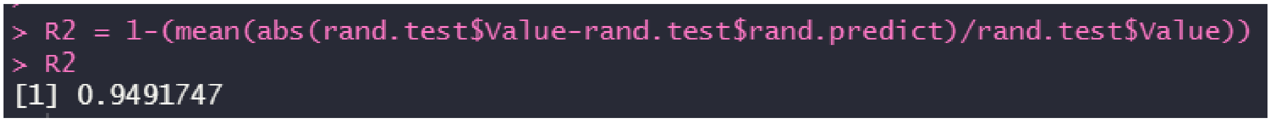
Using all the important variables, with number of trees = 500, gaussian distribution and interaction depth of 2, that is, two way interactions were fitted between the predictor variables.

The RSME for this model was **722773.1**

Summary

Comparing all the models above, it is safe to infer that the **Random Forest Model** stood out as the best model to predict value of players with RSME = 511268.2, and ‘Release.Clause’, ‘Overall’, and ‘Wage’ as the most important predictor variables.

From the R^2 value, which was 0.949, it can be concluded that the model is 99% accurate in predicting the outcome.



```
> R2 = 1-(mean(abs(rand.test$Value-rand.test$rand.predict)/rand.test$Value))
> R2
[1] 0.9491747
```