FIFA 18 – Continued

Guided by: Dr. Carlo Lipizzi

Project by Rushabh Barbhaya

Stevens Institute of Technology

EM 623 – Data Science and Knowledge Discovery

Index

Table of Contents

[Data understanding 3](#_Toc512727678)

[Data Preparation 1](#_Toc512727679)

[Knime 1](#_Toc512727680)

[1. CSV Reader 2](#_Toc512727681)

[2. Missing Value Filter 3](#_Toc512727682)

[3. Outlier Removal 4](#_Toc512727683)

[4. Normalizer 5](#_Toc512727684)

[5. CSV Writer 6](#_Toc512727685)

[Rattle 7](#_Toc512727686)

[Recoding 8](#_Toc512727687)

[Correlation matix 9](#_Toc512727688)

[Regression model / Linear model 10](#_Toc512727689)

[Testing 16](#_Toc512727690)

[Conclusion 18](#_Toc512727691)

List of Figures

[*Figure 1 Knime Workflow* 1](#_Toc512727652)

[*Figure 2 CSV Reader Settings* 2](#_Toc512727653)

[*Figure 3 Missing Value Column FIlter settings* 3](#_Toc512727654)

[*Figure 4 Outlier Removal Settings* 4](#_Toc512727655)

[*Figure 5 Normalizer Settings* 5](#_Toc512727656)

[*Figure 6 CSV Writer Settings* 6](#_Toc512727657)

[*Figure 7 Data loading in Rattle* 7](#_Toc512727658)

[*Figure 8 Recoding Categorical Values* 8](#_Toc512727659)

[*Figure 9 Correlation plot* 9](#_Toc512727660)

[*Figure 10 Selecting target variable* 10](#_Toc512727661)

[*Figure 11 Rstudio crashes on execution* 11](#_Toc512727662)

[*Figure 12 Initial Run (Remove NA values)* 13](#_Toc512727663)

[*Figure 13 1st Iteration for p-values* 14](#_Toc512727664)

[*Figure 14 second run with filtered values.* 14](#_Toc512727665)

[*Figure 15 2nd Iteration for p-values* 15](#_Toc512727666)

[*Figure 16 Snapshot from training dataset* 16](#_Toc512727667)

[*Figure 17 Snapshot of testing dataset* 17](#_Toc512727668)

# Data understanding

We had made a model for FIFA 18 in our midsem exam. I wanted to make a predictive model for the same. I wanted to resign a model which could predict the overall performance of players. So I set out to design such a model. In my other class I learnt about liner modeling and rattle provides a function for this purpose.

The dataset contains all information about all the players in the game of FIFA 18 & they are ranked on the basis of overall performance score. The Dataset matrix is of [74 x 17,981] events i.e. 74 columns and 17,981 rows. It consists of the following rows:

1. Name
2. Age
3. Photo
4. Nationality
5. Flag
6. Overall [ Target ]
7. Potential
8. Club
9. Club Logo
10. Value
11. Wage
12. Special
13. Acceleration
14. Aggression
15. Agility
16. Balance
17. Ball control
18. Composure
19. Crossing
20. Curve
21. Dribbling
22. Finishing
23. Free kick accuracy
24. GK diving
25. GK handling
26. GK kicking
27. GK positioning
28. GK reflexes
29. Heading accuracy
30. Interceptions
31. Jumping
32. Long passing
33. Long shots
34. Marking
35. Penalties
36. Positioning
37. Reactions
38. Short passing
39. Shot power
40. Sliding tackle
41. Sprint speed
42. Stamina
43. Standing tackle
44. Strength
45. Vision
46. Volleys
47. CAM
48. CB
49. CDM
50. CF
51. CM
52. ID
53. LAM
54. LB
55. LCB
56. LCM
57. LDM
58. LF
59. LM
60. LS
61. LW
62. LWB
63. Preferred Positions
64. RAM
65. RB
66. RCB
67. RCM
68. RDM
69. RF
70. RM
71. RS
72. RW
73. RWB
74. ST

# Data Preparation

For designing a predictive model, I used the following softwares:

* Excel
* Knime
* Rattle

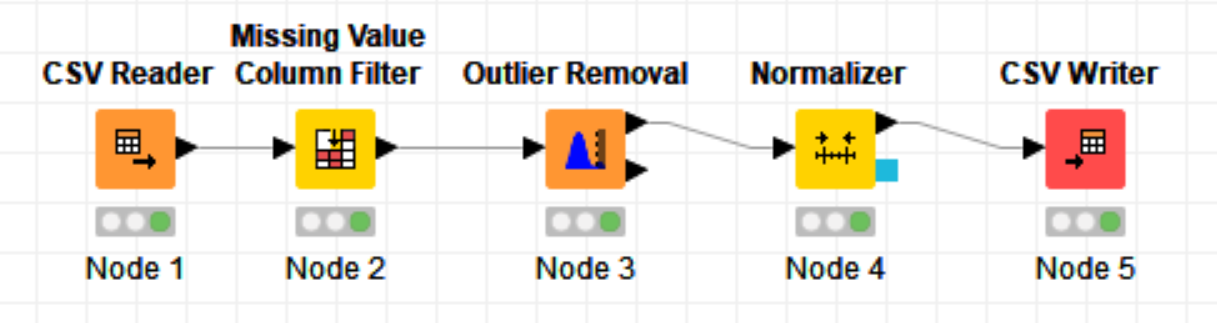
I cleaned some unnecessary data & ended up with the following columns:

1. Name
2. Age
3. Special
4. Acceleration
5. Aggression
6. Agility
7. Balance
8. Ball control
9. Composure
10. Crossing
11. Curve
12. Dribbling
13. Finishing
14. Free kick accuracy
15. GK diving
16. GK handling
17. GK kicking
18. GK positioning
19. GK reflexes
20. Heading accuracy
21. Interceptions
22. Jumping
23. Long passing
24. Long shots
25. Marking
26. Penalties
27. Positioning
28. Reactions
29. Short passing
30. Shot power
31. Sliding tackle
32. Sprint speed
33. Stamina
34. Standing tackle
35. Strength
36. Vision
37. Volleys
38. CAM
39. CB
40. CDM
41. CF
42. CM
43. LAM
44. LB
45. LCB
46. LCM
47. LDM
48. LF
49. LM
50. LS
51. LW
52. LWB
53. RAM
54. RB
55. RCB
56. RCM
57. RDM
58. RF
59. RM
60. RS
61. RW
62. RWB
63. ST
64. Overall [Target]

# Knime

I used Rattle for a small step before Knime. More explanation in Rattle section.

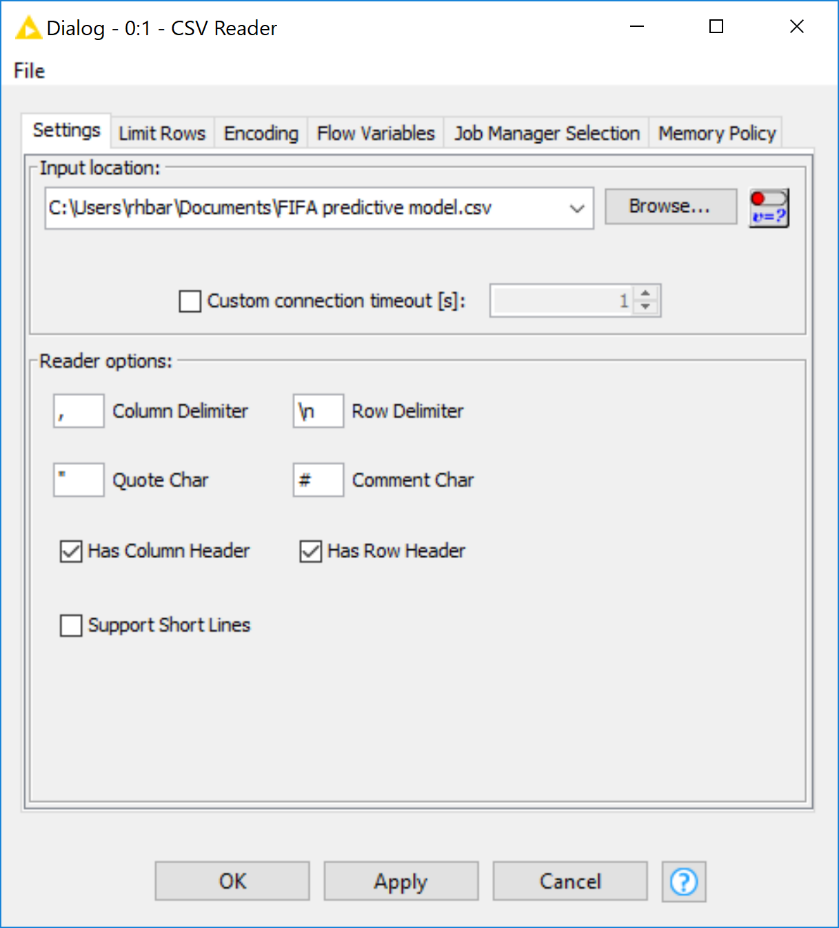
I mainly used Knime to clean the dataset. The workflow in Knime is shown in Figure 1 below.



*Figure 1 Knime Workflow*

## CSV Reader

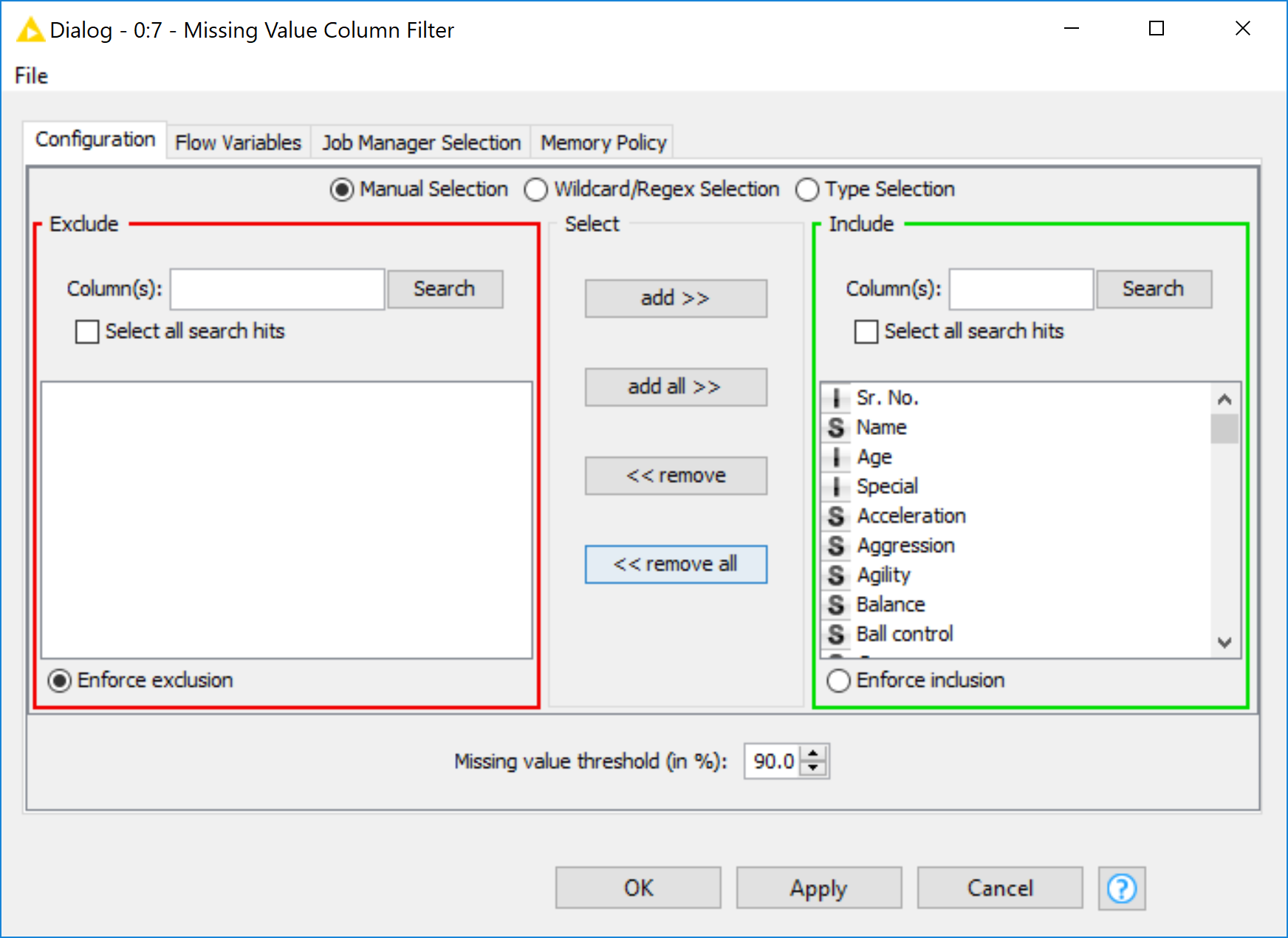
CSV reader was used to import the csv file to knime workflow. Image 2 shows the settings for this node.



*Figure 2 CSV Reader Settings*

## Missing Value Filter

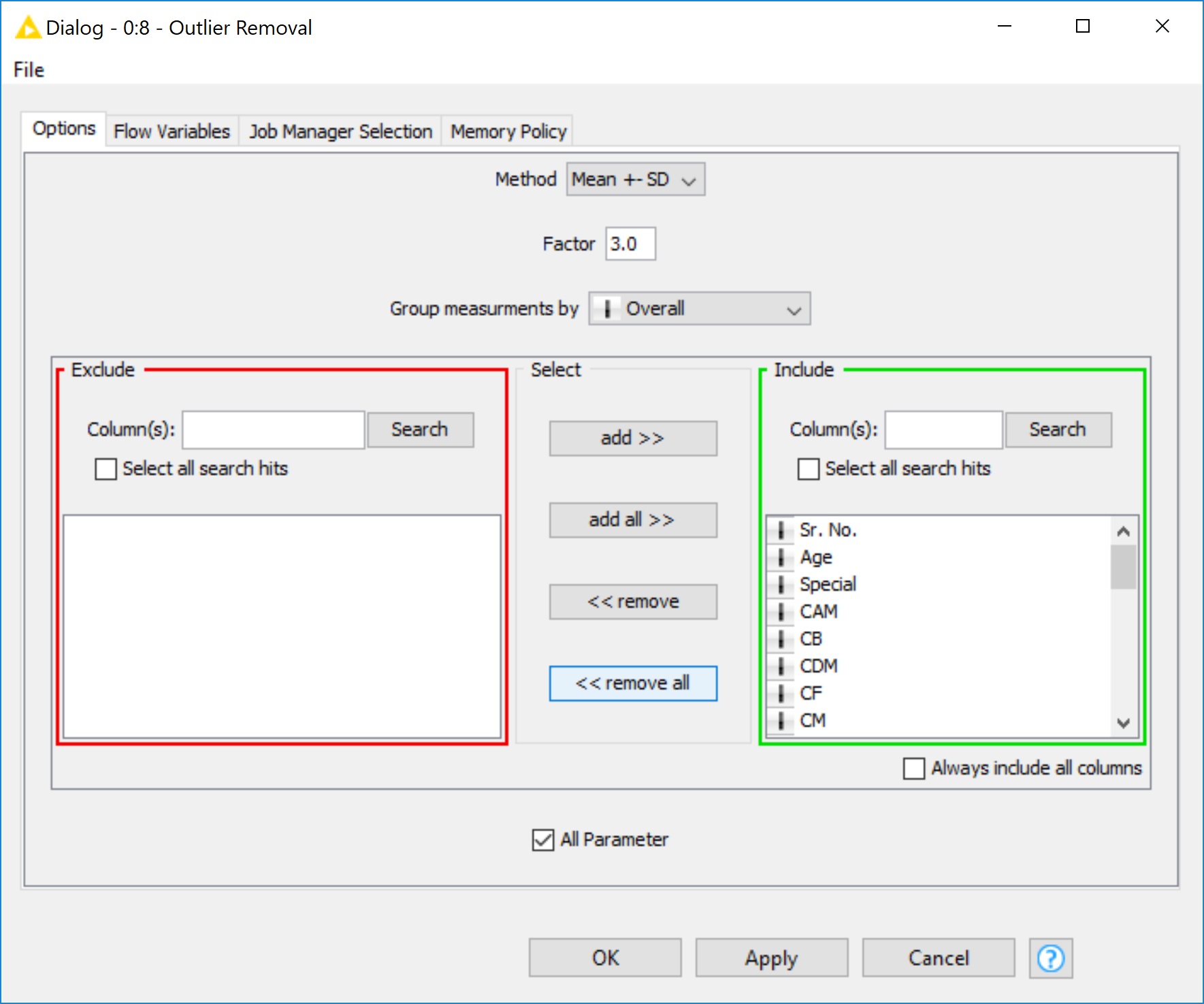
Missing value filter will remove any missing columns which has less than the threshold value set by the user. I had set the filter threshold value to 90%. The image below shows the settings for missing value column filter.



*Figure 3 Missing Value Column FIlter settings*

## Outlier Removal

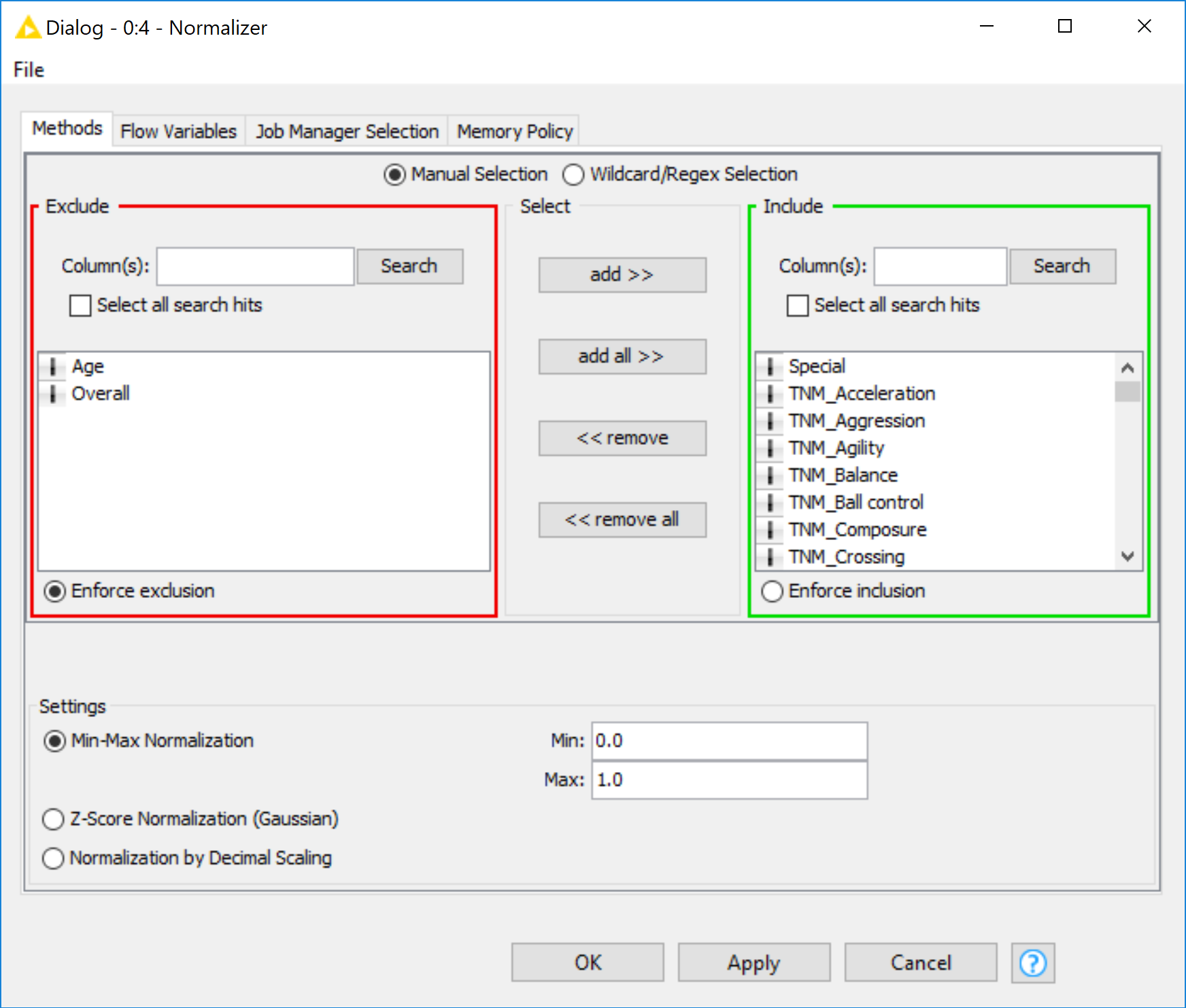
There are always some values in dataset which are either too small or too big. Those values affect our calculations and could give a misleading value, making the entire model less accurate. Therefore, I used an outlier remover. And set its value to ± 3 S.D. (standard deviations) apart. Meaning that the filter will remove any value which is 3 S.D. away from the median value. The image shows the settings for the outlier remover.



*Figure 4 Outlier Removal Settings*

## Normalizer

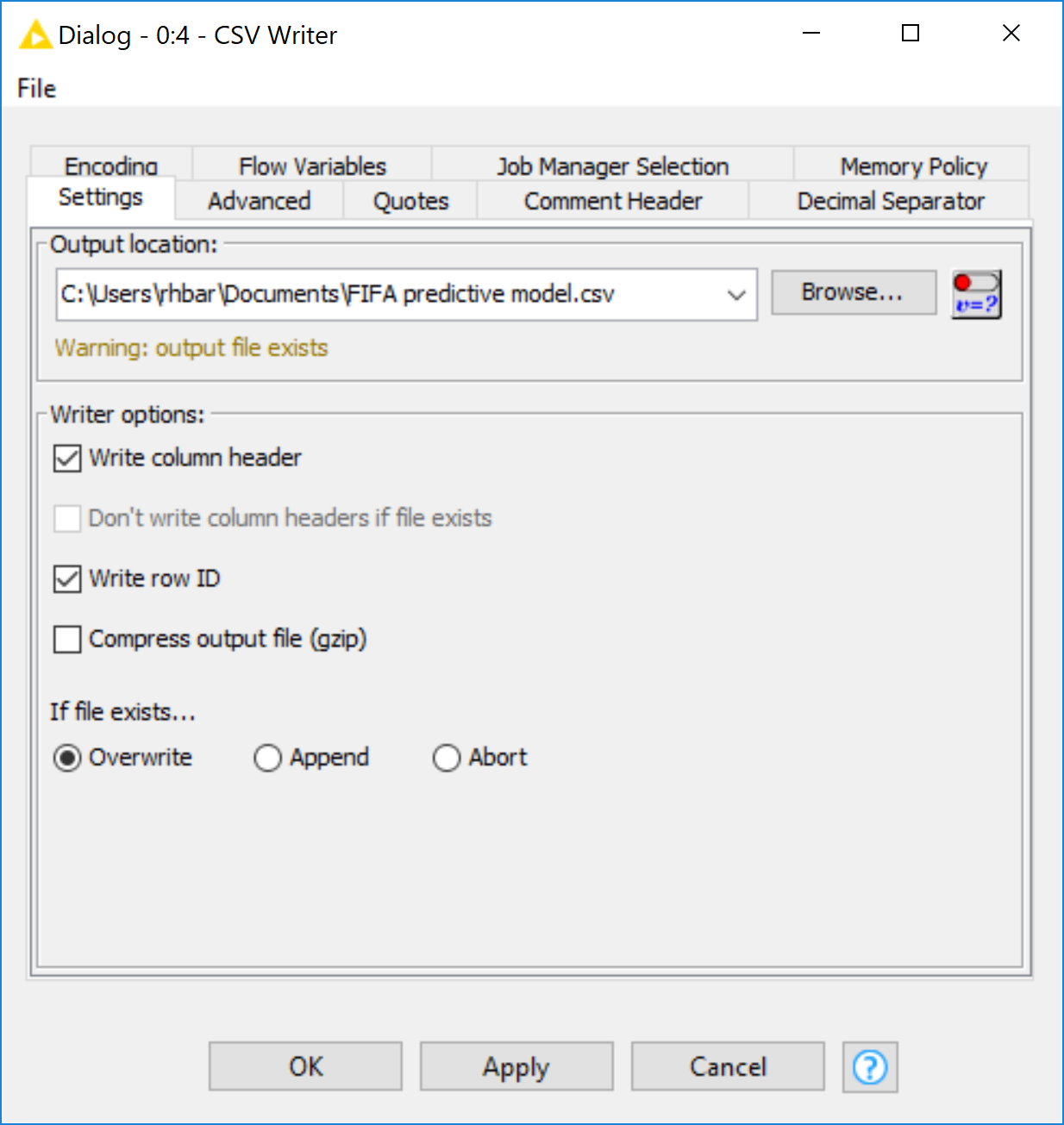
I used a normalizer because it sets the value of all the parameters between 1 & 0. This step really helps in calculation. Following image shows the settings for Normalizer. I excluded age and overall from this. Overall being the Output parameter on which the whole model was based. And age being the sorting parameter for this case. I used a Mix-Max Normalizer, Setting ranges to 0 & 1 respectively.



*Figure 5 Normalizer Settings*

## CSV Writer

Finally to export all the work. I added a csv writer node. It just saves the output in CSV format. Which I can use in Rattle. Following are the settings for CSV Writer.



*Figure 6 CSV Writer Settings*

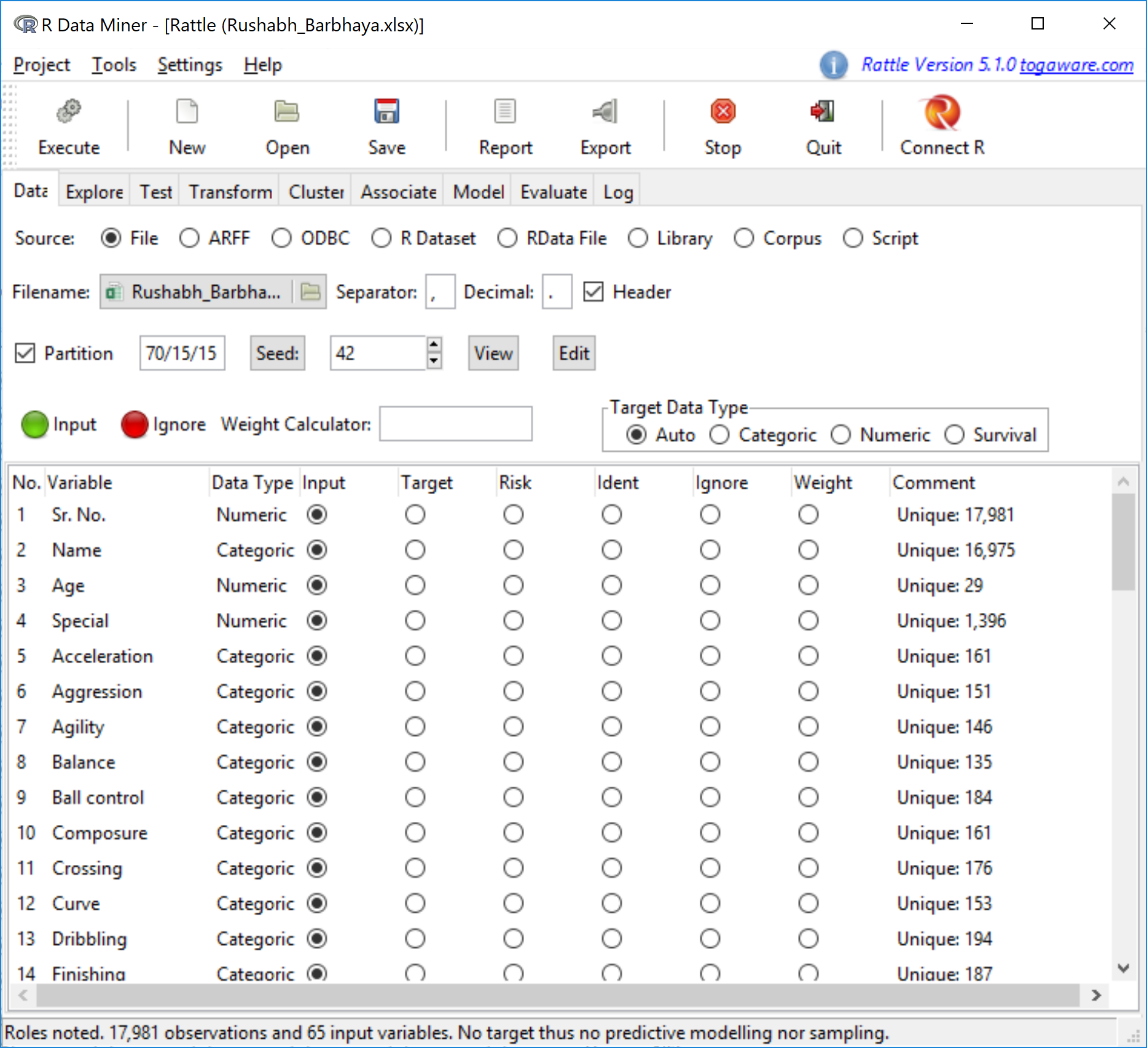
# Rattle

Explanation as mentioned before: I had to use rattle because for some reason some of the numeric values were being displayed as categorical values. It is not possible to design a regression model/linear model using categorical values.

To do that. I loaded the file in opened Rstudio and executed the following commands in the editor.

* library(rattle)
* rattle

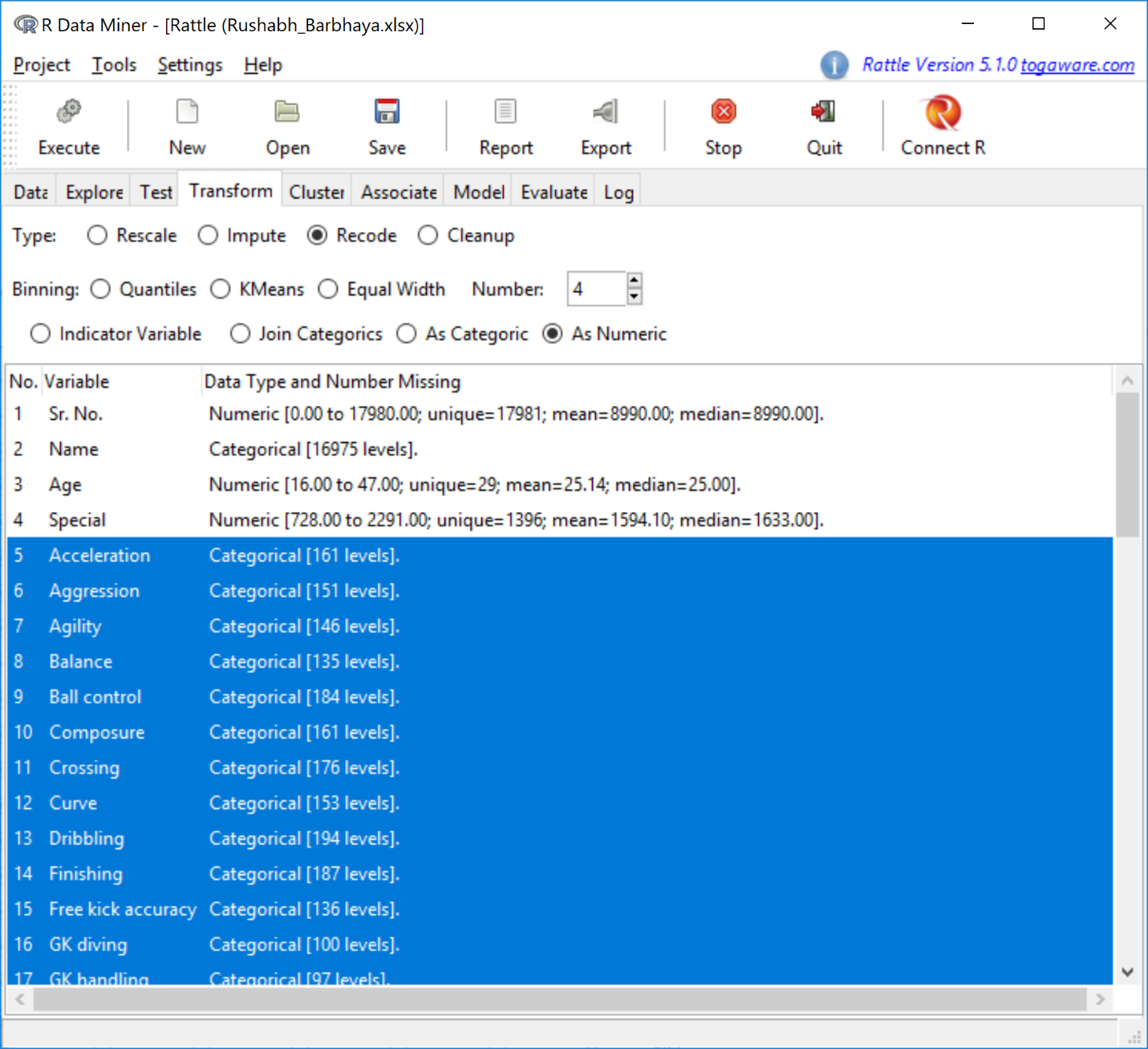
After executing it I loaded the output csv file from knime to rattle & made a 70/30 partition for training and testing the model. Then then press execute. After executing the screen should look like this.



*Figure 7 Data loading in Rattle*

## Recoding

As mentioned before, due to some reason the numeric value were shown as categorical values. To rectify this, I had to recode the variables as numeric. To do that, move to ‘Transform’ tab and select ‘recode’ option. The process is shown in the image below.

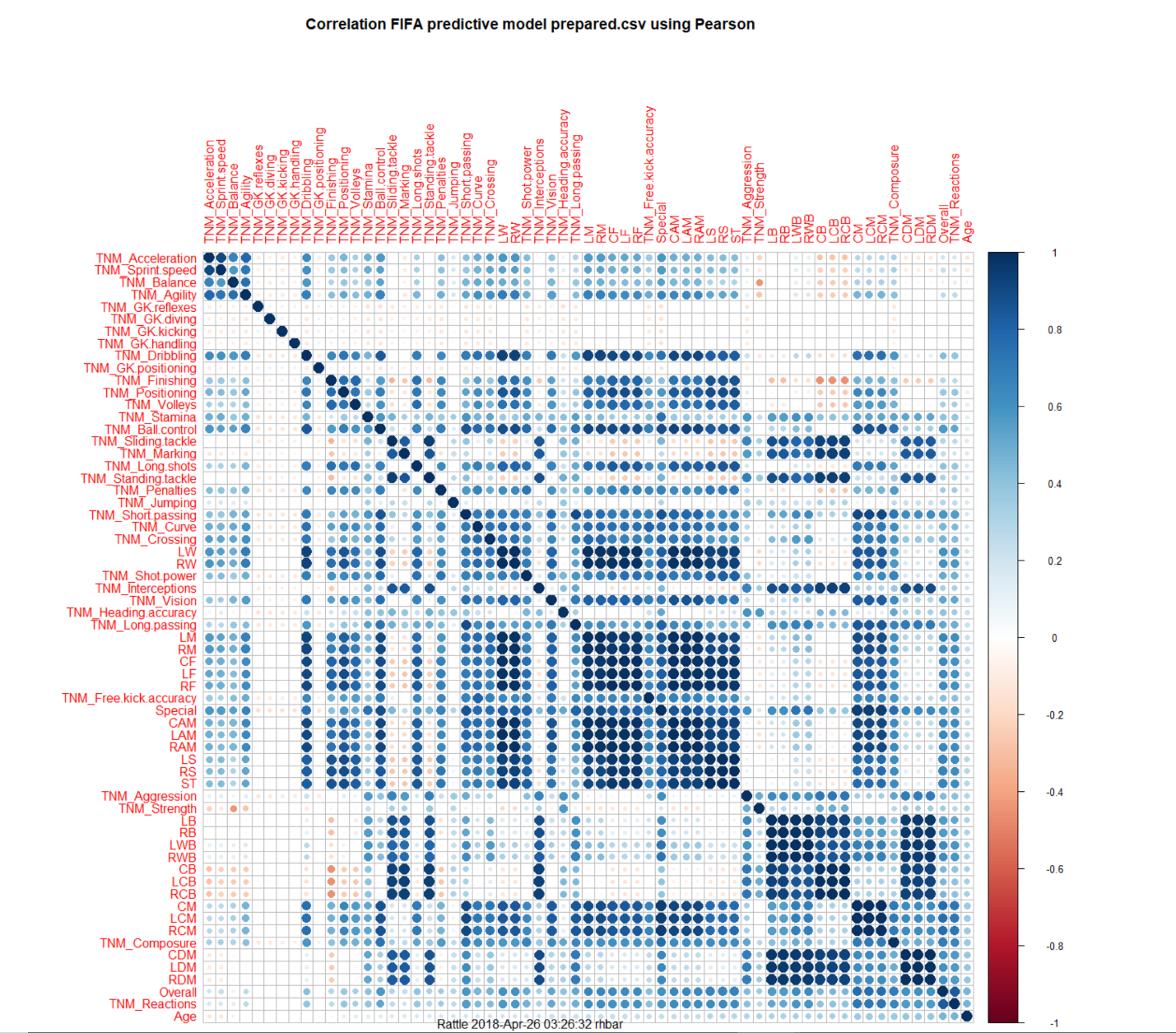


*Figure 8 Recoding Categorical Values*

After selecting the data (shown in figure 8 as highlighted text) press ‘execute’. The new values will appear in the ‘data’ tab and the old values will be automatically moved to ignore. After doing that I check for the correlation among data.

## Correlation matix

Correlation matrix shows the correlation between 2 variables. They are in the range of -1 to 1. It represents the data being directly proportional (value greater than 0 and towards 1) or inversely proportion (value less than 0 and towards -1) to each other. To get correlation plot. Select ‘test’ tab and then select correlation. I used the Pearson correlation, for this plot. The plot shows up in rattle. It look as follows.

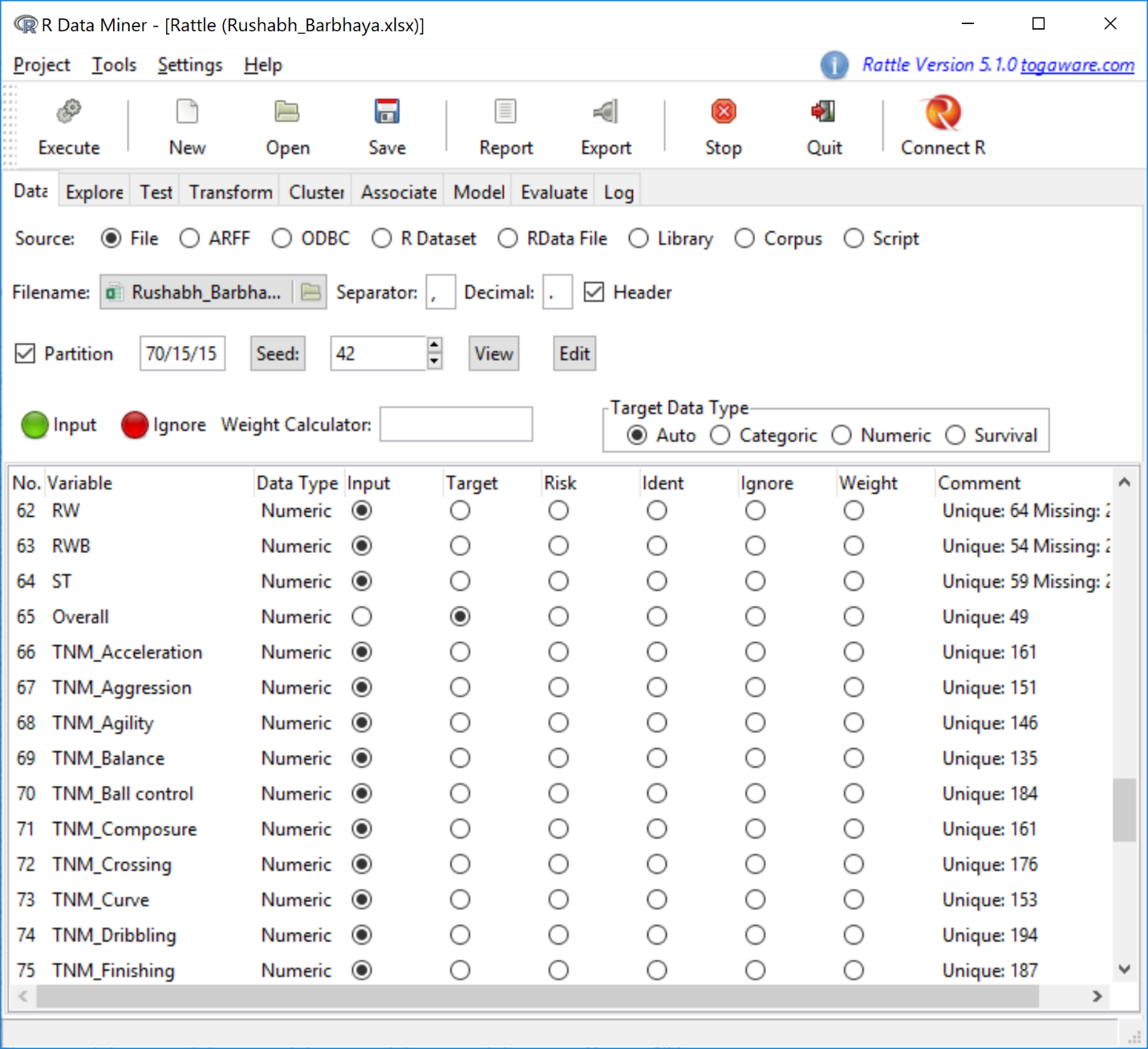


*Figure 9 Correlation plot*

From the plot it is evident that all the values are in direct correlation with overall parameter.

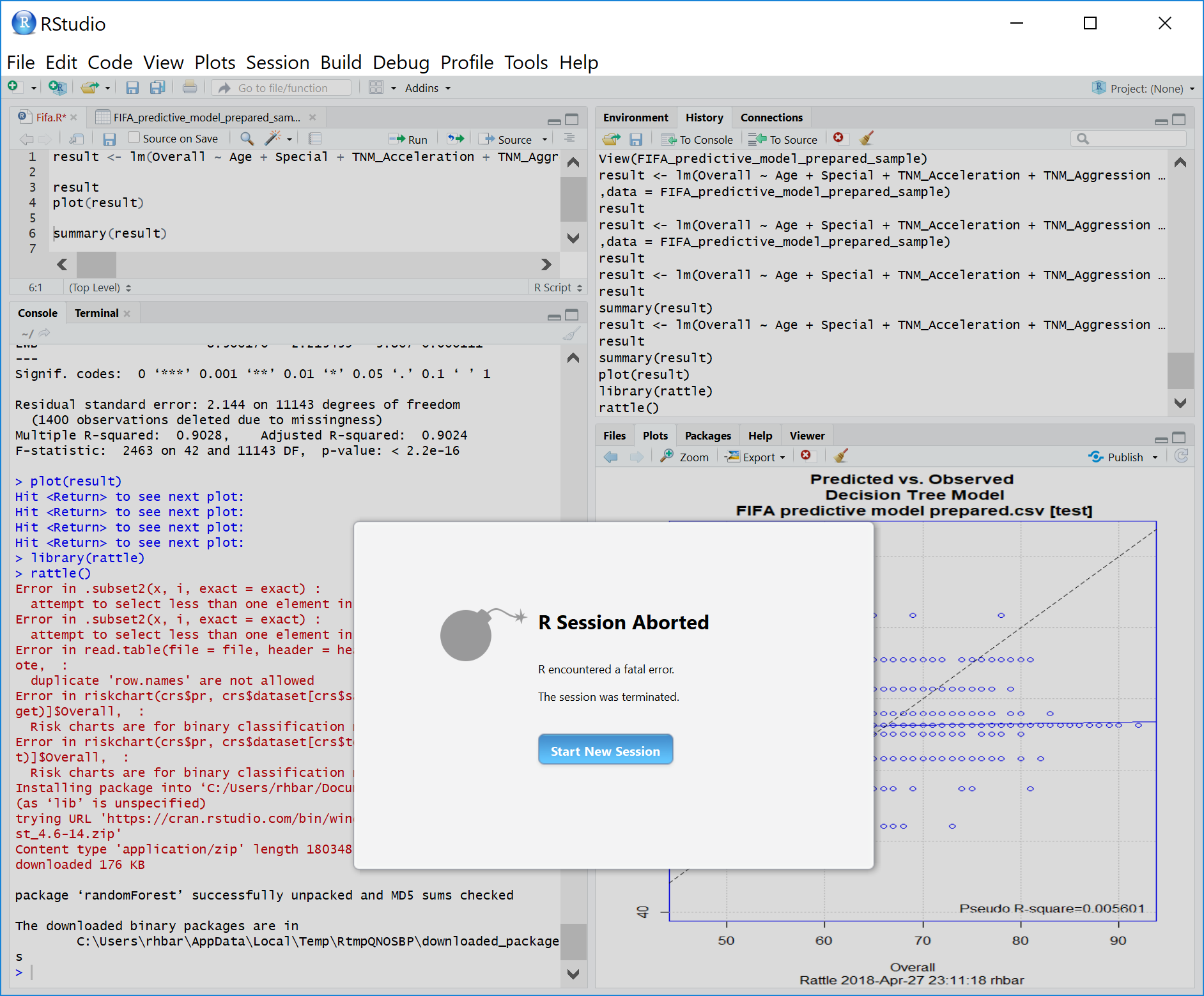
## Regression model / Linear model

Now with all the information gather I tried to run the linear model from ‘Model’ tab. But first we have to set overall as the target variable. To do that, open ‘data’ tab search for overall and set it to target (shown in image below)



*Figure 10 Selecting target variable*

Now that this is out of our way, run the linear test from model tab as mentioned before. But I faced some problems here. My rattle would crash every time I run that test. To solve this I searched for running regression model using R scripts. After some digging and testing I got tried to understand a small piece of code. Code indicated below.



*Figure 11 Rstudio crashes on execution*

CODE:

result <- lm(Overall ~ Age + Special + TNM\_Acceleration + TNM\_Aggression + TNM\_Agility + TNM\_Balance + TNM\_Ball.control + TNM\_Composure + TNM\_Crossing + TNM\_Curve + TNM\_Dribbling + TNM\_Finishing + TNM\_Free.kick.accuracy + TNM\_GK.diving + TNM\_GK.handling + TNM\_GK.kicking + TNM\_GK.reflexes + TNM\_Heading.accuracy + TNM\_Interceptions + TNM\_Jumping + TNM\_Long.passing + TNM\_Long.shots + TNM\_Marking + TNM\_Penalties + TNM\_Positioning + TNM\_Reactions + TNM\_Short.passing + TNM\_Shot.power + TNM\_Sliding.tackle + TNM\_Sprint.speed + TNM\_Stamina + TNM\_Standing.tackle + TNM\_Strength + TNM\_Volleys + CAM + CB + CDM + CF + CM + LM + LW + LWB,data = FIFA\_predictive\_model\_prepared\_sample.csv)

result

plot(result)

summary(result)

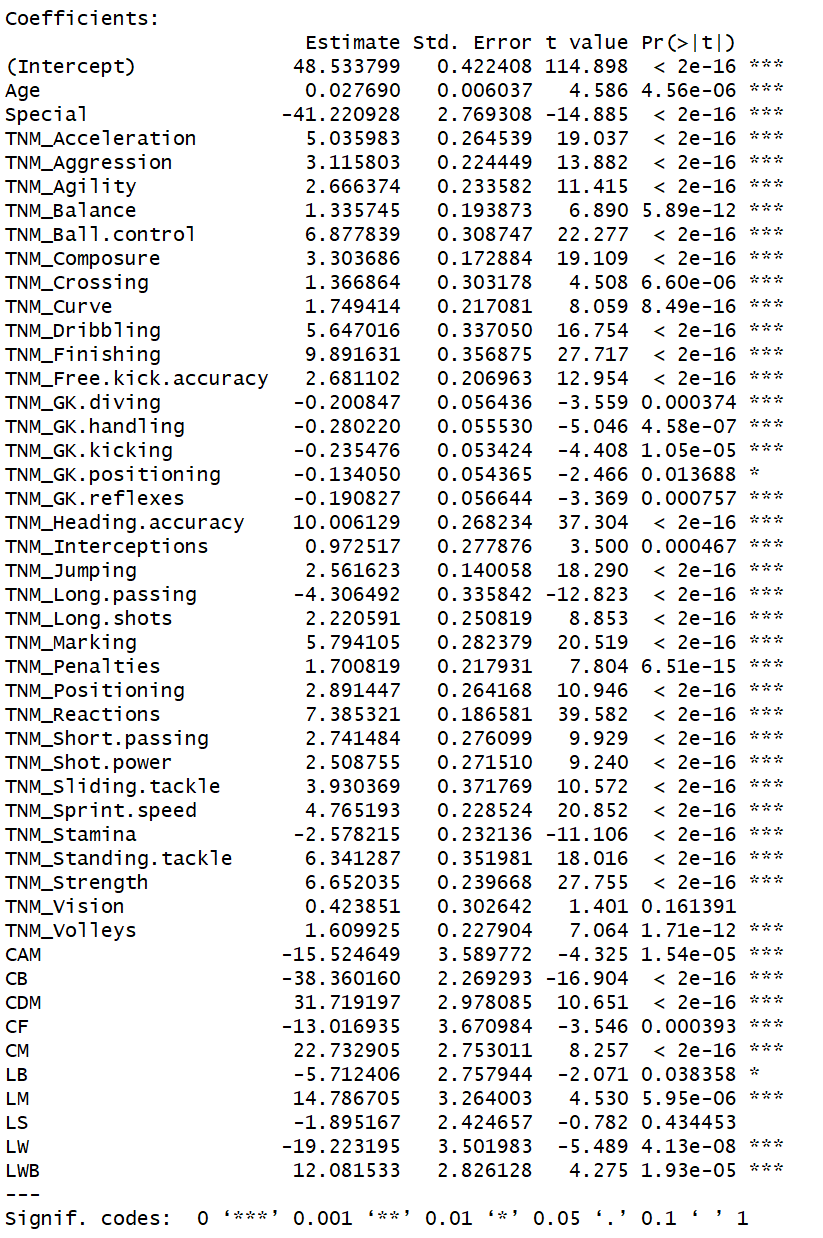
The above mentioned code is after 2 iterations. From first iteration I got the p-values/significance values from the code. After removing those which are least significant from the bunch. I ran the above code and got the final equation for ŷ

Ŷ = 48.819425 + (Age \* 0.027807) + (special \* -41.3307788) + (acceleration \* 4.99107) + (aggression \* 3.292963) + (agility \* 2.654985) + (balance \* 1.340296) + (ball control \* 6.835208) + (composure \* 3.337893) + (crossing \* 1.50318) + (curve \* 1.755898) + (dribbing \* 5.956804) + (finishing \* 9.808877) + (free kick accuracy \* 2.716243) + (gk diving \* -0.202428) + (gk handling \* -0.282977) + (gk kicking \* -0.23756) + (gk reflexes \* -0.196072) + (heading accuracy \* 9.761582) + (interceptions \* 1.021782) + (jumping \* 2.600327) + (long passing \* -4.42979) + (long shots \* 2.169266) + (marking \* 5.879911) + (penalties \* 1.715637) + (positioning \* 2.873695) + (reactions \* 7.337975) + (short passing \* 2.70014) + (shot power \* 2.394246) + (sliding tackle \* 3.663203) + (sprint speed \* 4.673068) + (stamina \* -2.644983) + (standing teckle \* 6.302579) + (strength \* 6.75879) + (volleys \* 1.574903) + (cam \* -12.756392) + (cb \* -39.755634) + (cdm \* 30.104203) + (cf \* -15.560082) + (cm \* 24.585474) + (lm \* 14.004634) + (lw \* -21.115013) + (lwb \* 8.56617)

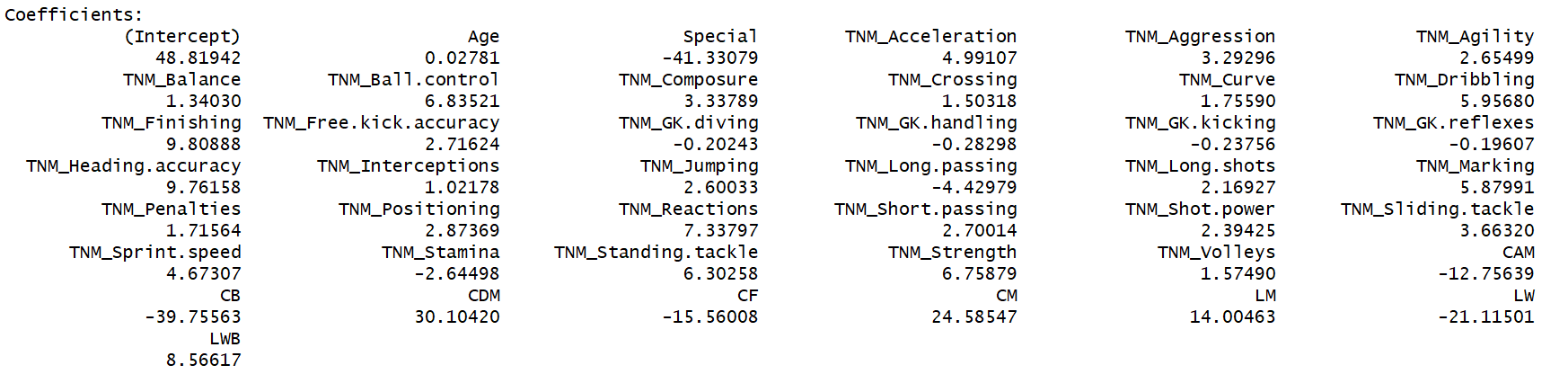
Following are the images of iterations:



*Figure 12 Initial Run (Remove NA values)*



*Figure 13 1st Iteration for p-values*



*Figure 14 second run with filtered values.*



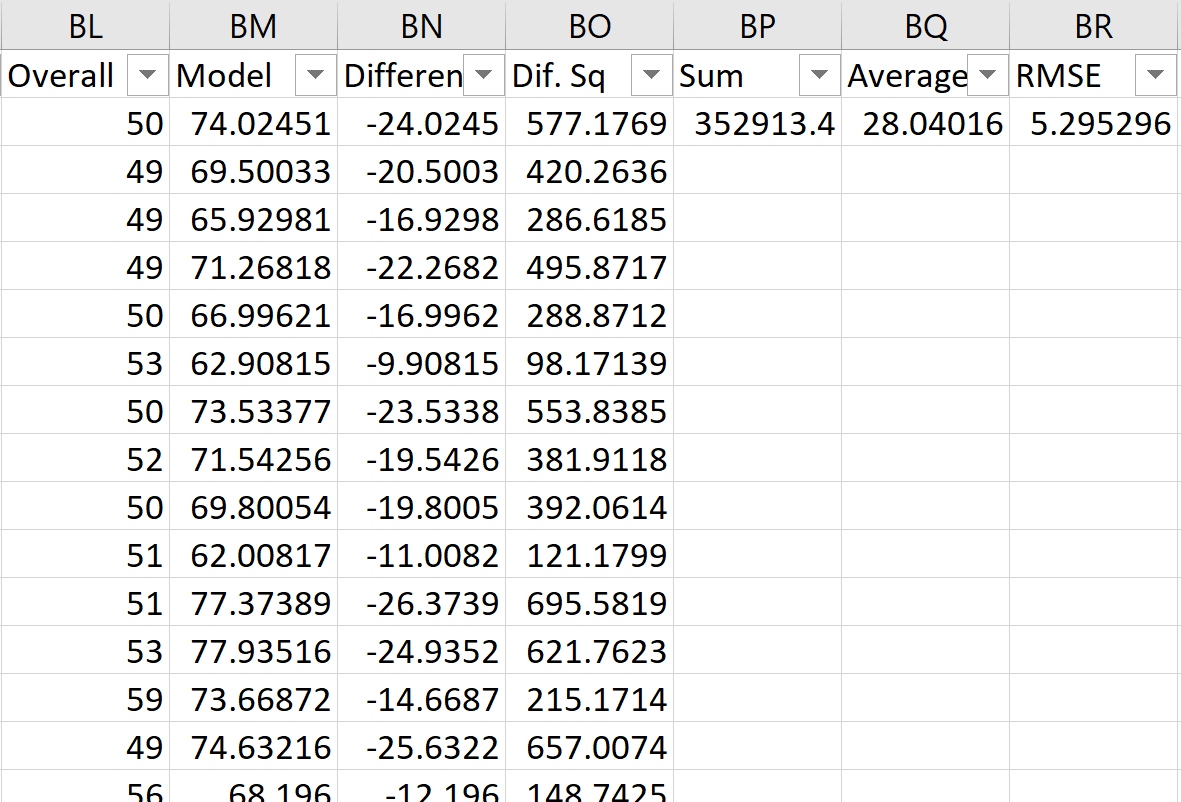
*Figure 15 2nd Iteration for p-values*

After this I faced a different problem. As I was not able to run linear model on rattle, I didn’t get testing csv file. I could obtain training file but not the testing file. Therefore, I had to manually check the training file and the whole dataset & make a new excel sheet for testing model.

# Testing

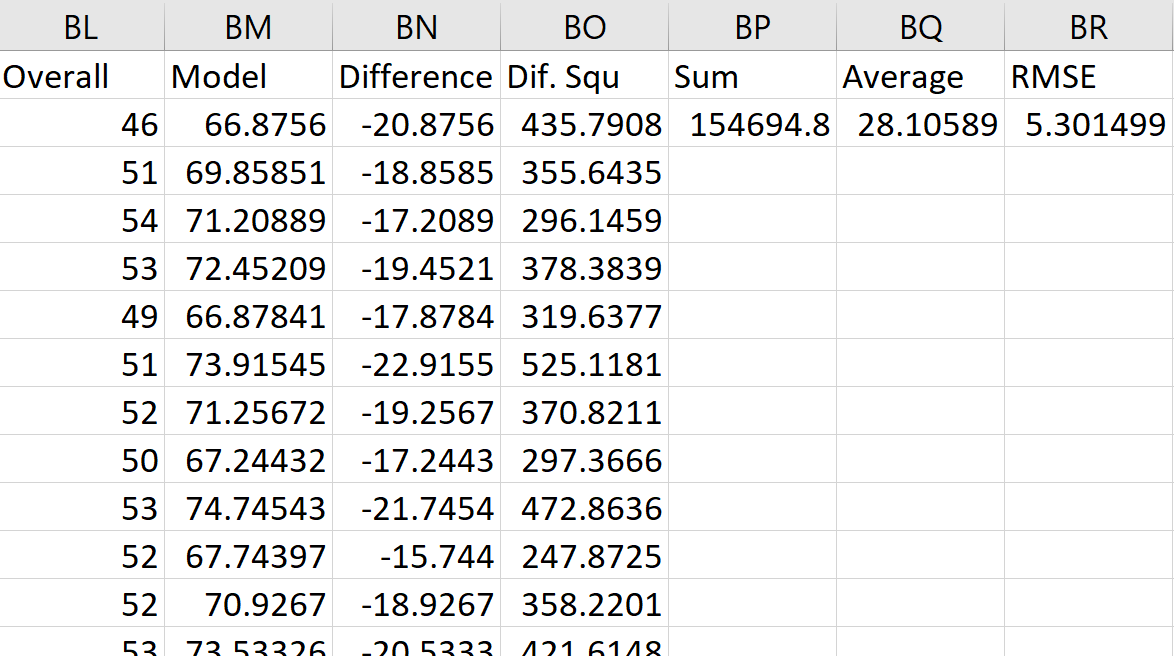
Now, after developing the model equation it was time to test the model. But again, it wasn’t possible in rattle so I had to perform the root mean square error for training and testing model using the book formulas.

After applying the formula to training dataset we get RMSE value as ≈ 5.29



*Figure 16 Snapshot from training dataset*

And the RMSE value for testing dataset was ≈ 5.30



*Figure 17 Snapshot of testing dataset*

# Conclusion

Even though I expected a lower score of RMSE value considering the size of the dataset. An overall RMSE value of 5.3 looks decent as the number of dependent variables are a lot and can affect the result a lot. I could have lowered the number of variables after matching the data from the correlation matrix. I trusted p-values more as compared to correlation matrix. Overall it can be justified as a decent model.