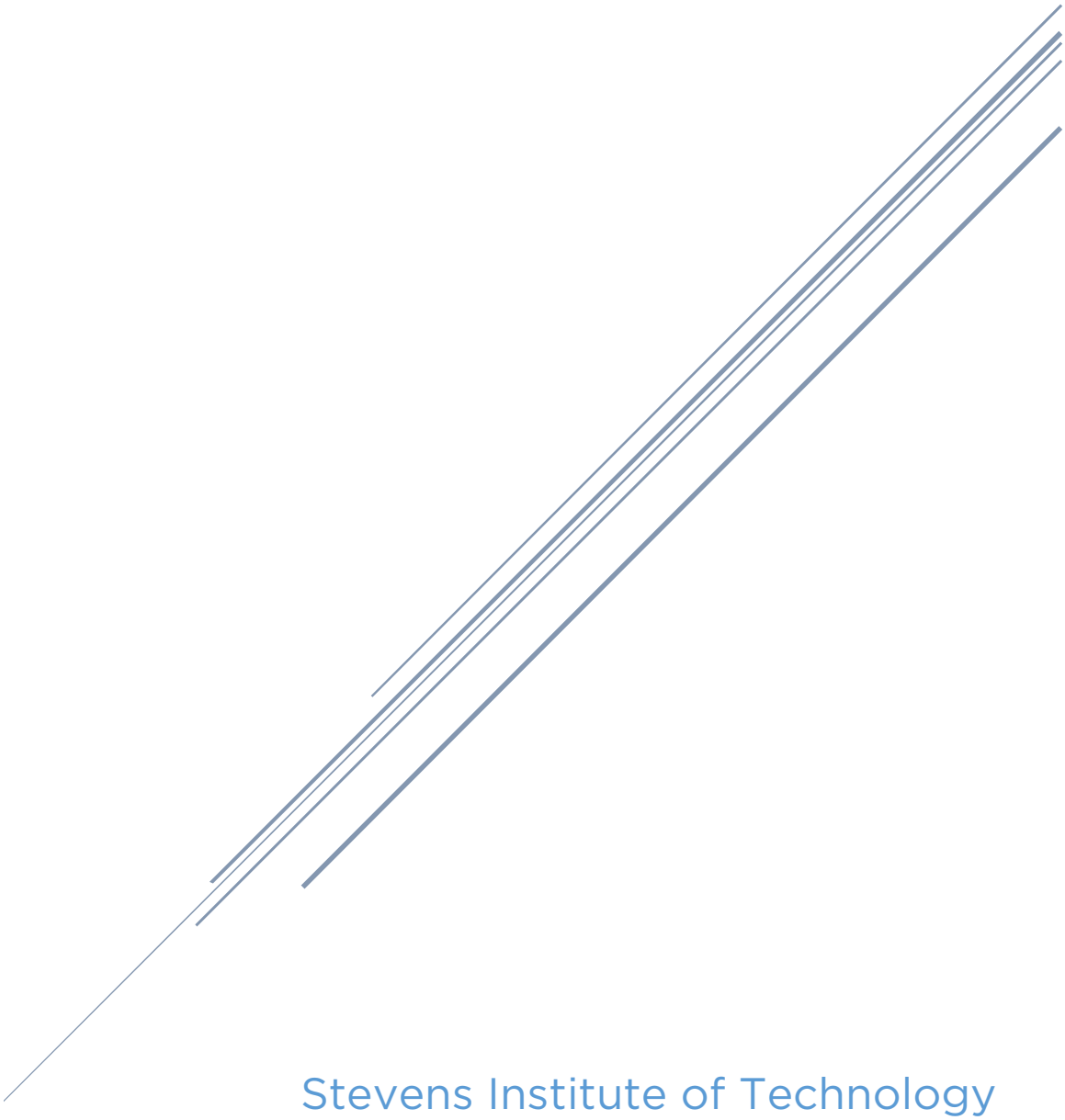
FIFA 18 – CONTINUED

Guided by: Dr. Carlo Lipizzi

Project by Rushabh Barbhaya



Stevens Institute of Technology

EM 623 – Data Science and Knowledge Discovery

INDEX

TABLE OF CONTENTS

LIST OF FIGURES

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.

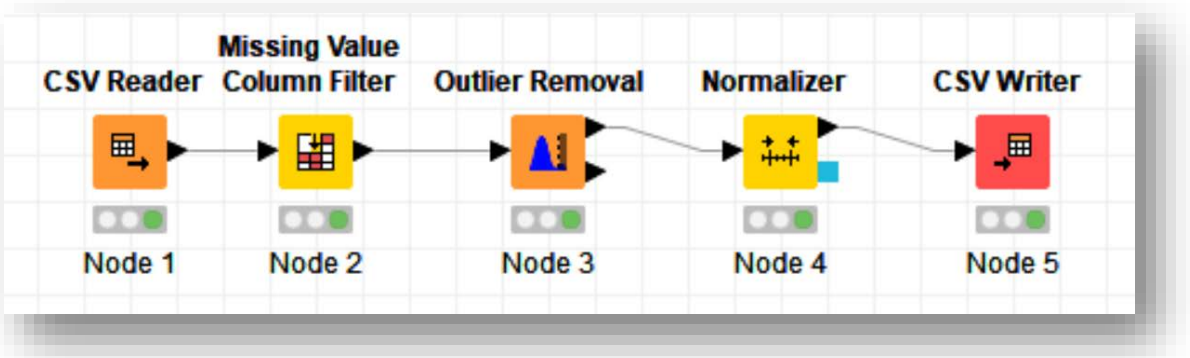


Figure 1 Knime Workflow

1. CSV READER

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

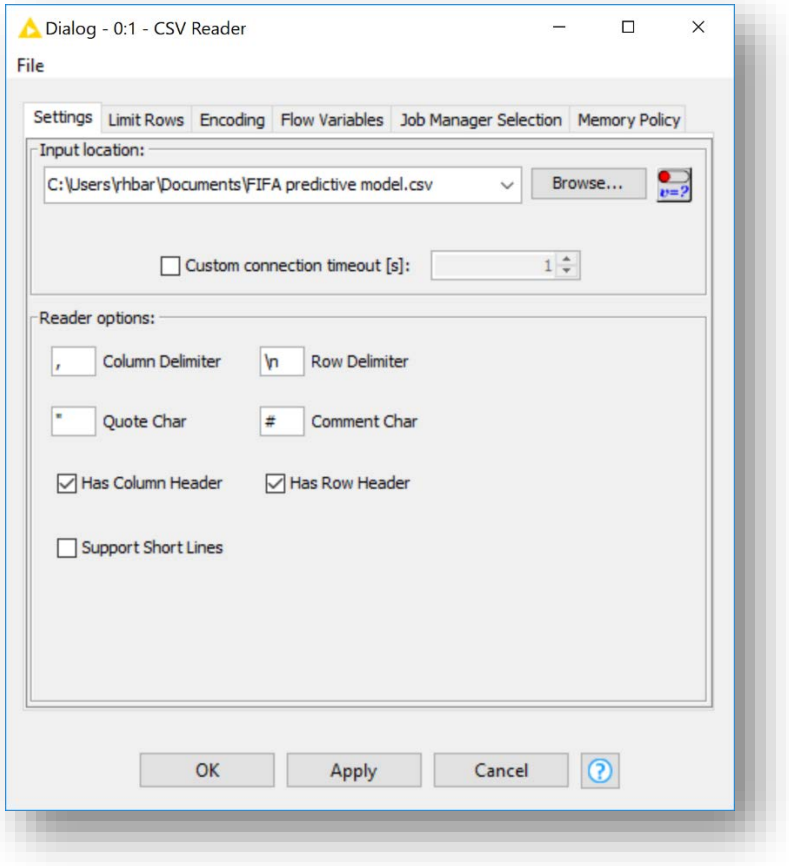


Figure 2 CSV Reader Settings

2. 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.

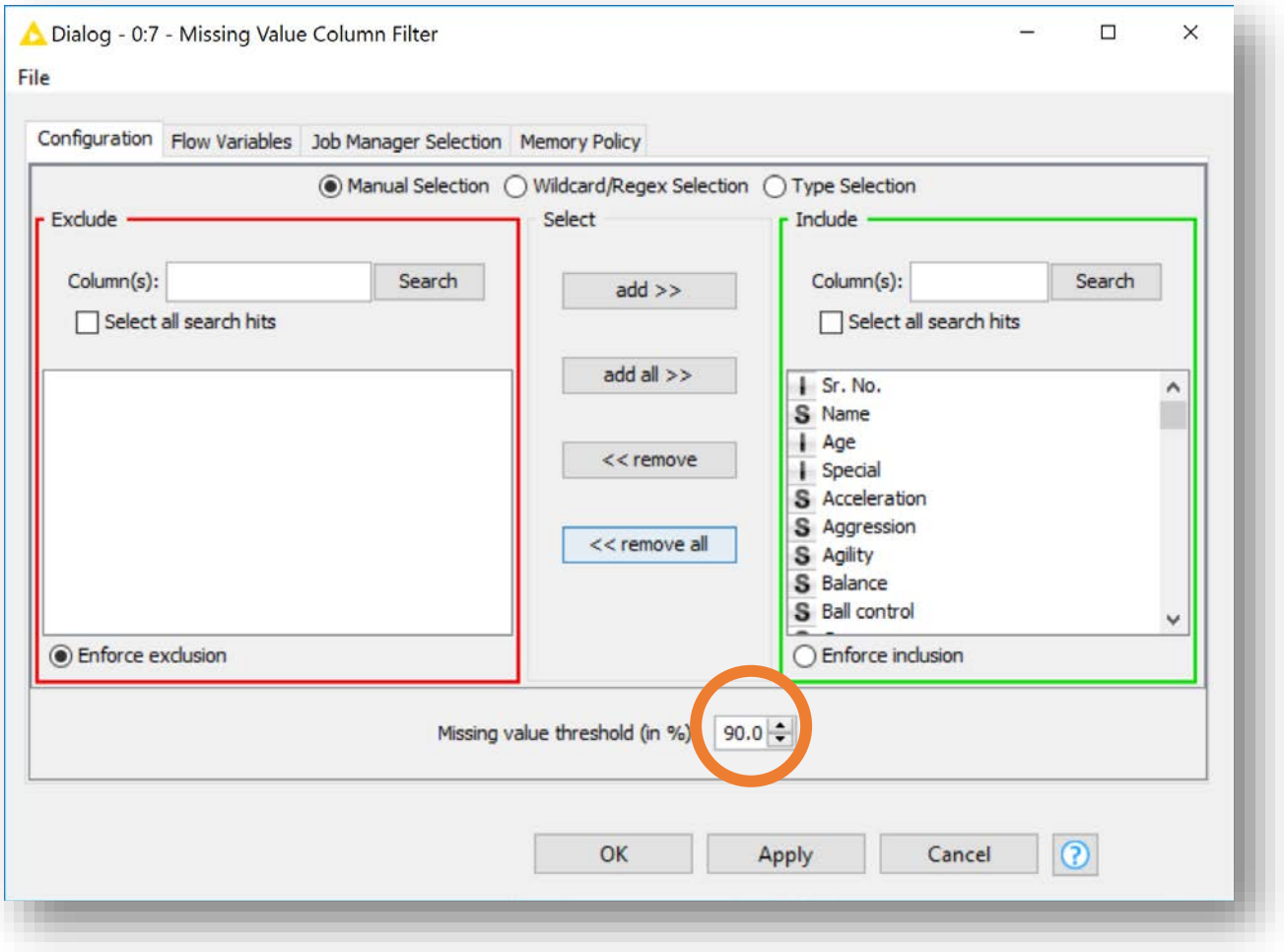


Figure 3 Missing Value Column FIlter settings

3. 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.

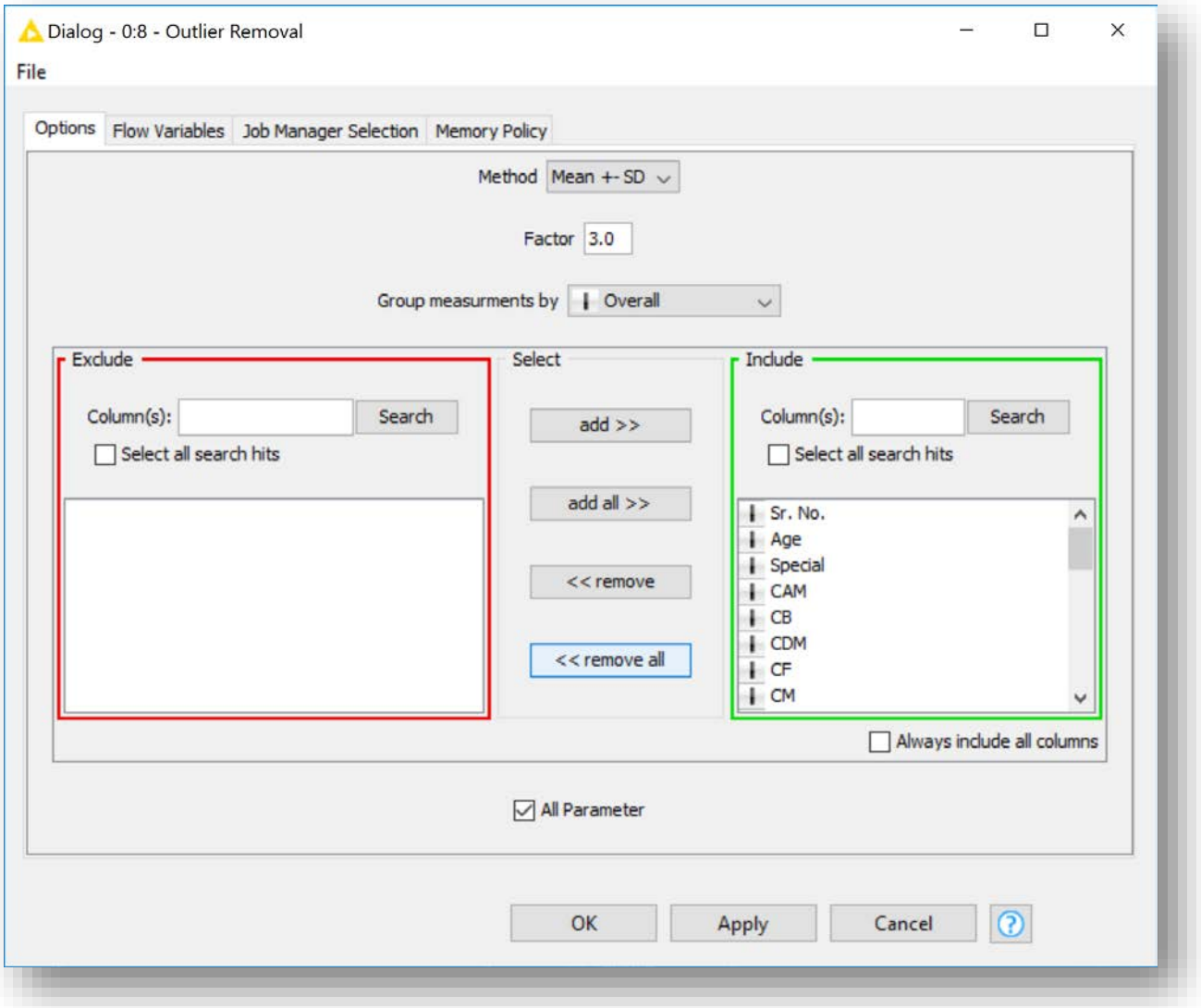


Figure 4 Outlier Removal Settings

4. 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.

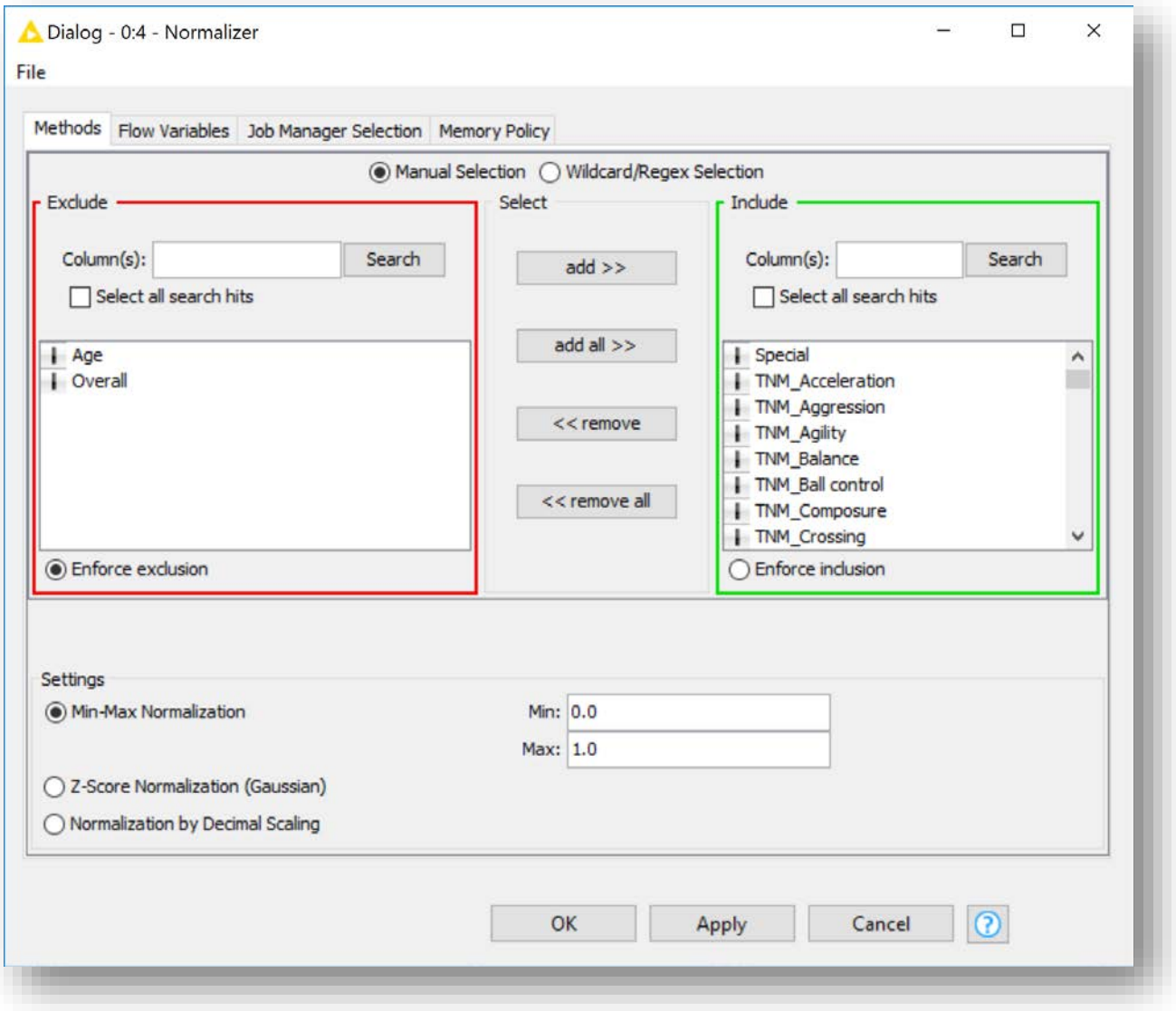


Figure 5 Normalizer Settings

5. 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.

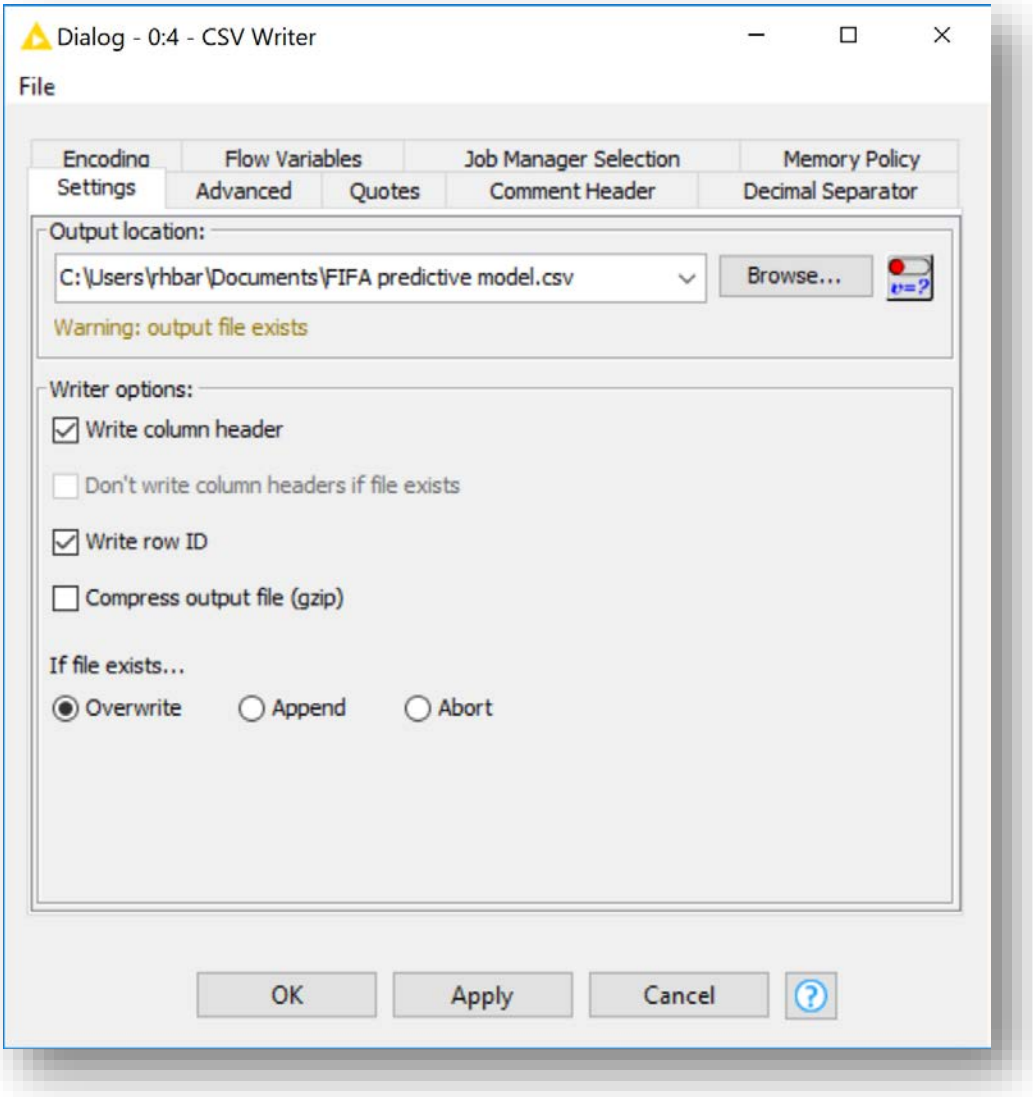


Figure 6 CSV Writer Settings

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.

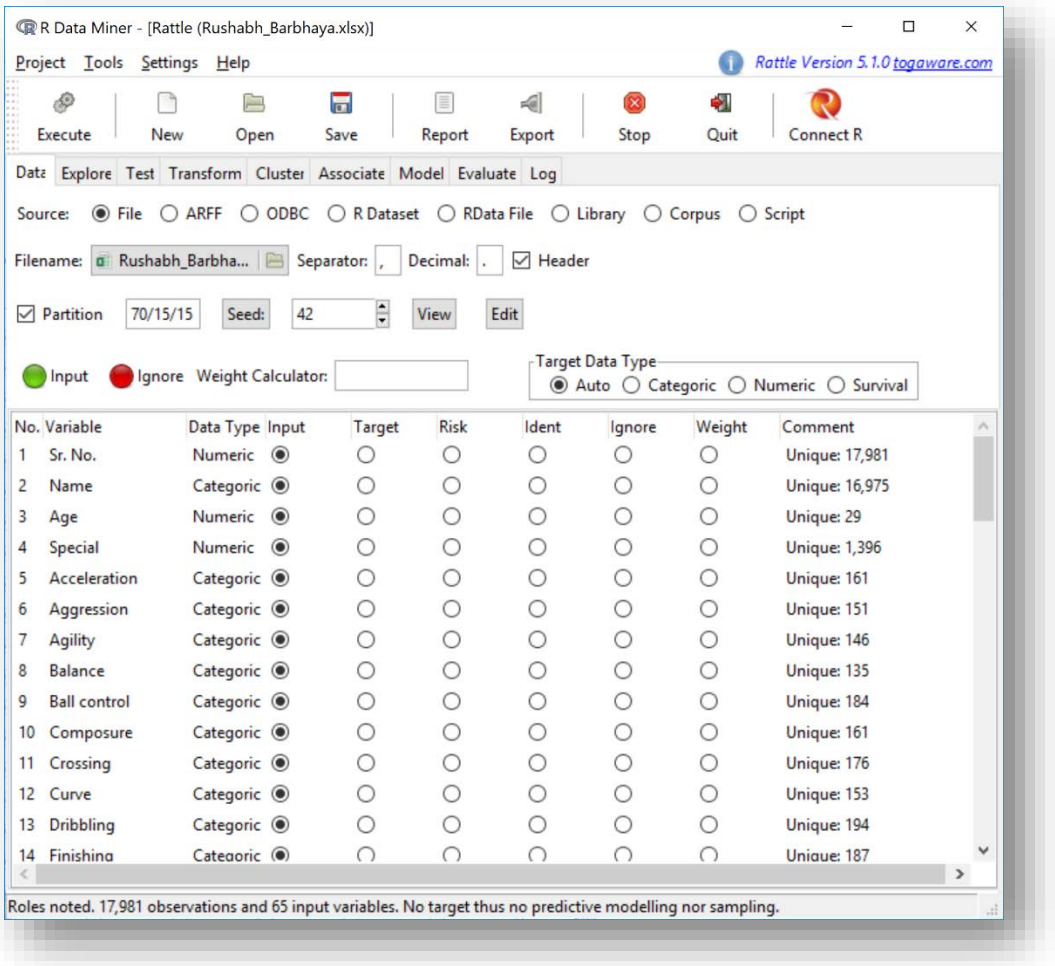


Figure 7 Data loading in Rattle

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.

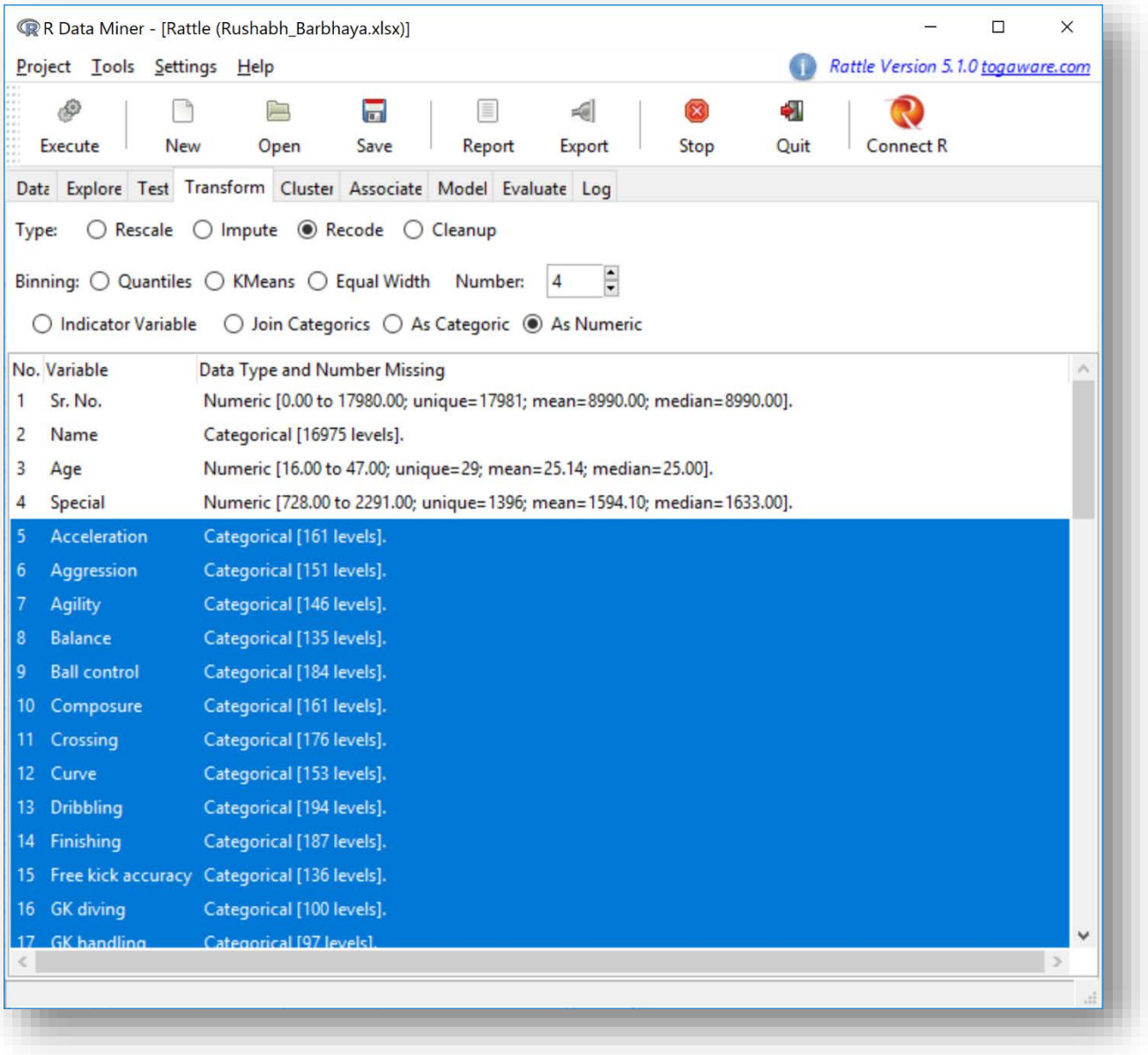


Figure 8 Recoding Categorical Values

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.

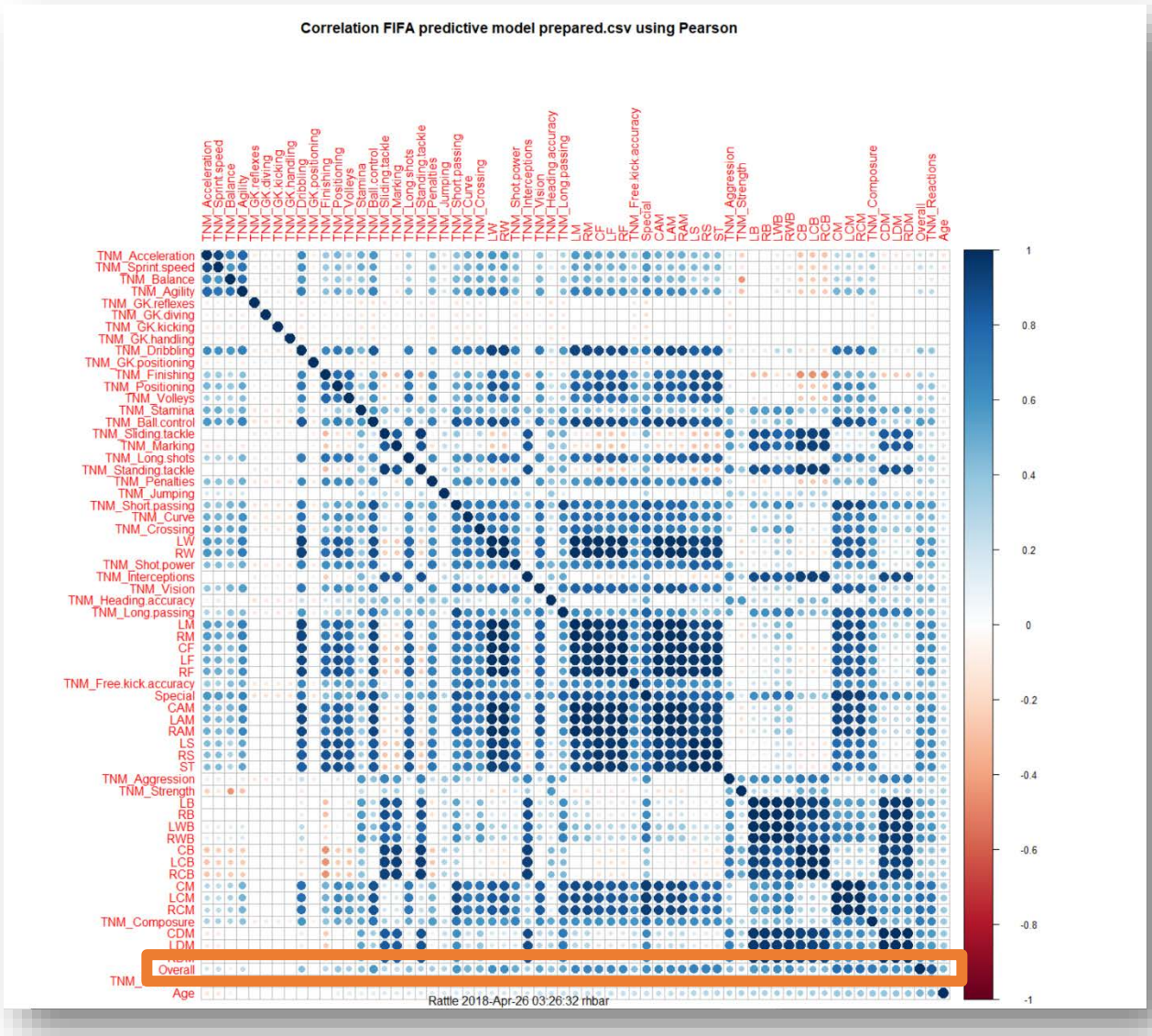


Figure 9 Correlation plot

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)

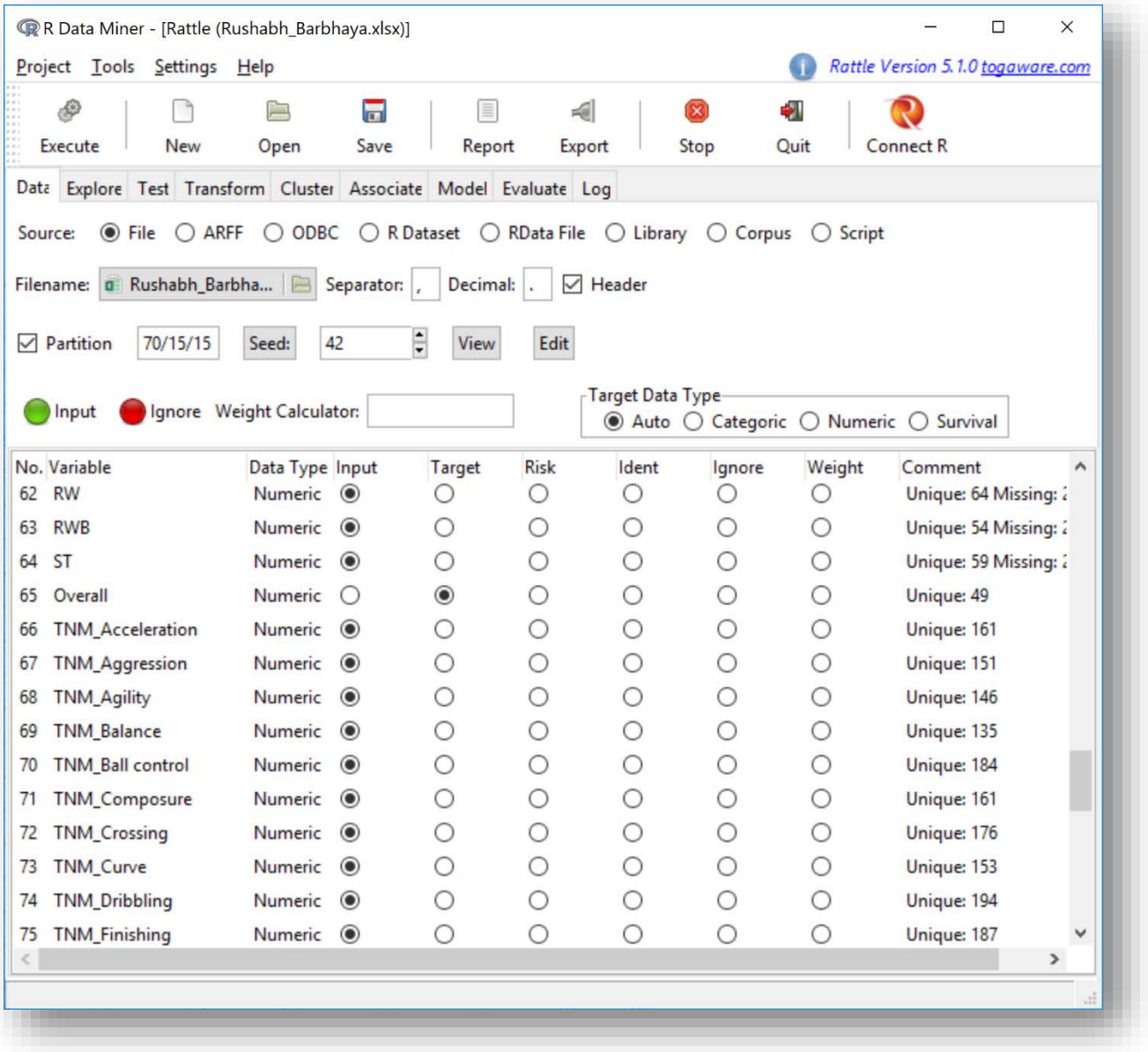


Figure 10 Selecting target variable

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.

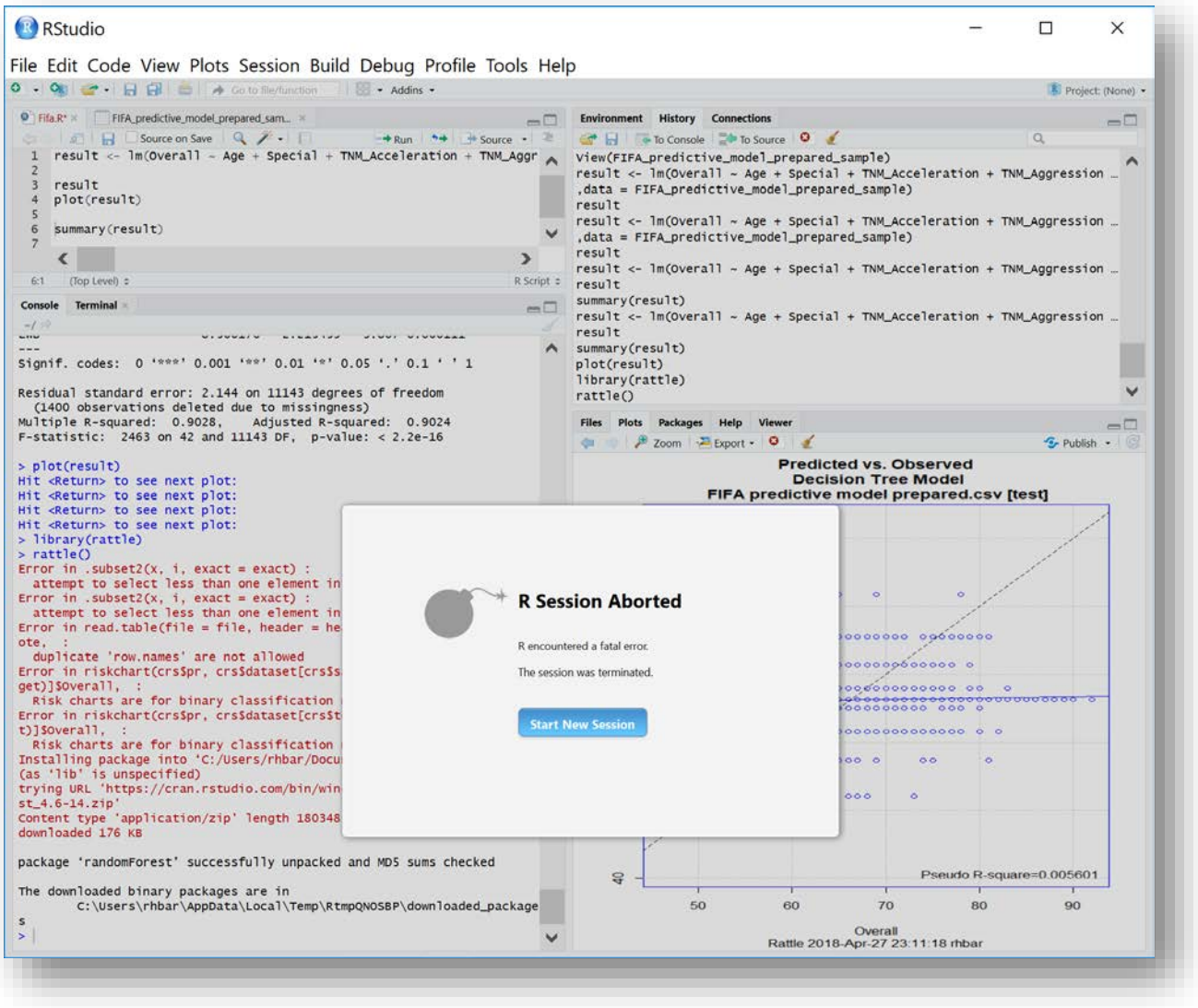


Figure 11 Rstudio crashes on execution

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 $\hat{y}$

$\hat{Y}$ = 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:

Coefficients:

(Intercept)	Age	Special	TNM_Acceleration	TNM_Aggression	TNM_Agility
48.53380	0.02769	-41.22093	5.03598	3.11580	2.66637
TNM_Balance	TNM_Ball.control	TNM_Composure	TNM_Crossing	TNM_Curve	TNM_Dribbling
1.33574	6.87784	3.30369	1.36686	1.74941	5.64702
TNM_Finishing	TNM_Free.kick.accuracy	TNM_GK.diving	TNM_GK.handling	TNM_GK.kicking	TNM_GK.positioning
9.89163	2.68110	-0.20085	-0.28022	-0.23548	-0.13405
TNM_GK.reflexes	TNM_Heading.accuracy	TNM_Interceptions	TNM_Jumping	TNM_Long.passing	TNM_Long.shots
-0.19083	10.00613	0.97252	2.56162	-4.30649	2.22059
TNM_Marking	TNM_Penalties	TNM_Positioning	TNM_Reactions	TNM_Short.passing	TNM_Shot.power
5.79411	1.70082	2.89145	7.38532	2.74148	2.50876
TNM_Sliding.tackle	TNM_Sprint.speed	TNM_Stamina	TNM_Standing.tackle	TNM_Strength	TNM_Vision
3.93037	4.76519	-2.57821	6.34129	6.65203	0.42385
TNM_Volleys	CAM	CB	CDM	CF	CM
1.60992	-15.52465	-38.36016	31.71920	-13.01693	22.73290
LAM	LB	LCB	LCM	LDM	LF
NA	-5.71241	NA	NA	NA	NA
LM	LS	LW	LWB	RAM	RB
14.78671	-1.89517	-19.22320	12.08153	NA	NA
RCB	RCM	RDM	RF	RM	RS
NA	NA	NA	NA	NA	NA
RW	RWB	ST			
NA	NA	NA			

Figure 12 Initial Run (Remove NA values)

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             48.533799   0.422408 114.898  < 2e-16 ***
Age                      0.027690   0.006037   4.586 4.56e-06 ***
Special                -41.220928   2.769308 -14.885  < 2e-16 ***
TNM_Acceleration         5.035983   0.264539  19.037  < 2e-16 ***
TNM_Aggression           3.115803   0.224449  13.882  < 2e-16 ***
TNM_Agility              2.666374   0.233582  11.415  < 2e-16 ***
TNM_Balance              1.335745   0.193873   6.890 5.89e-12 ***
TNM_Ball.control         6.877839   0.308747  22.277  < 2e-16 ***
TNM_Composure            3.303686   0.172884  19.109  < 2e-16 ***
TNM_Crossing             1.366864   0.303178   4.508 6.60e-06 ***
TNM_Curve                1.749414   0.217081   8.059 8.49e-16 ***
TNM_Dribbling            5.647016   0.337050  16.754  < 2e-16 ***
TNM_Finishing            9.891631   0.356875  27.717  < 2e-16 ***
TNM_Free.kick.accuracy   2.681102   0.206963  12.954  < 2e-16 ***
TNM_GK.diving           -0.200847   0.056436  -3.559 0.000374 ***
TNM_GK.handling         -0.280220   0.055530  -5.046 4.58e-07 ***
TNM_GK.kicking          -0.235476   0.053424  -4.408 1.05e-05 ***
TNM_GK.positioning      -0.134050   0.054365  -2.466 0.013688 *
TNM_GK.reflexes         -0.190827   0.056644  -3.369 0.000757 ***
TNM_Heading.accuracy    10.006129   0.268234  37.304  < 2e-16 ***
TNM_Interceptions        0.972517   0.277876   3.500 0.000467 ***
TNM_Jumping              2.561623   0.140058  18.290  < 2e-16 ***
TNM_Long.passing        -4.306492   0.335842 -12.823  < 2e-16 ***
TNM_Long.shots           2.220591   0.250819   8.853  < 2e-16 ***
TNM_Marking              5.794105   0.282379  20.519  < 2e-16 ***
TNM_Penalties            1.700819   0.217931   7.804 6.51e-15 ***
TNM_Positioning          2.891447   0.264168  10.946  < 2e-16 ***
TNM_Reactions            7.385321   0.186581  39.582  < 2e-16 ***
TNM_Short.passing        2.741484   0.276099   9.929  < 2e-16 ***
TNM_Shot.power           2.508755   0.271510   9.240  < 2e-16 ***
TNM_Sliding.tackle       3.930369   0.371769  10.572  < 2e-16 ***
TNM_Sprint.speed         4.765193   0.228524  20.852  < 2e-16 ***
TNM_Stamina             -2.578215   0.232136 -11.106  < 2e-16 ***
TNM_Standing.tackle      6.341287   0.351981  18.016  < 2e-16 ***
TNM_Strength             6.652035   0.239668  27.755  < 2e-16 ***
TNM_Vision               0.423851   0.302642   1.401 0.161391
TNM_Volleys              1.609925   0.227904   7.064 1.71e-12 ***
CAM                    -15.524649   3.589772  -4.325 1.54e-05 ***
CB                     -38.360160   2.269293 -16.904  < 2e-16 ***
CDM                     31.719197   2.978085  10.651  < 2e-16 ***
CF                     -13.016935   3.670984  -3.546 0.000393 ***
CM                      22.732905   2.753011   8.257  < 2e-16 ***
LB                      -5.712406   2.757944  -2.071 0.038358 *
LM                      14.786705   3.264003   4.530 5.95e-06 ***
LS                      -1.895167   2.424657  -0.782 0.434453
LW                     -19.223195   3.501983  -5.489 4.13e-08 ***
LWB                     12.081533   2.826128   4.275 1.93e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 13 1st Iteration for p-values

```
Coefficients:
     (Intercept)               Age           Special  TNM_Acceleration  TNM_Aggression     TNM_Agility
        48.81942           0.02781          -41.33079           4.99107         3.29296         2.65499
     TNM_Balance   TNM_Ball.control     TNM_Composure     TNM_Crossing     TNM_Curve     TNM_Dribbling
         1.34030           6.83521           3.33789           1.50318         1.75590         5.95680
  TNM_Finishing TNM_Free.kick.accuracy    TNM_GK.diving   TNM_GK.handling  TNM_GK.kicking  TNM_GK.reflexes
         9.80888           2.71624          -0.20243          -0.28298        -0.23756        -0.19607
TNM_Heading.accuracy  TNM_Interceptions     TNM_Jumping  TNM_Long.passing  TNM_Long.shots    TNM_Marking
         9.76158           1.02178           2.60033          -4.42979         2.16927         5.87991
   TNM_Penalties   TNM_Positioning     TNM_Reactions  TNM_Short.passing  TNM_Shot.power TNM_Sliding.tackle
         1.71564           2.87369           7.33797           2.70014         2.39425         3.66320
 TNM_Sprint.speed     TNM_Stamina TNM_Standing.tackle     TNM_Strength     TNM_Volleys             CAM
         4.67307          -2.64498           6.30258           6.75879         1.57490       -12.75639
              CB               CDM                CF                CM              LM              LW
       -39.75563          30.10420         -15.56008          24.58547        14.00463       -21.11501
             LWB
         8.56617
```

Figure 14 second run with filtered values.

```
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           48.819425   0.329826 148.016  < 2e-16 ***
Age                    0.027807   0.006021   4.618 3.91e-06 ***
Special              -41.330788   2.558437 -16.155  < 2e-16 ***
TNM_Acceleration       4.991070   0.256737  19.440  < 2e-16 ***
TNM_Aggression         3.292963   0.199952  16.469  < 2e-16 ***
TNM_Agility            2.654985   0.229160  11.586  < 2e-16 ***
TNM_Balance            1.340296   0.189802   7.062 1.74e-12 ***
TNM_Ball.control       6.835208   0.306796  22.279  < 2e-16 ***
TNM_Composure          3.337893   0.172294  19.373  < 2e-16 ***
TNM_Crossing           1.503180   0.287341   5.231 1.71e-07 ***
TNM_Curve              1.755898   0.212817   8.251  < 2e-16 ***
TNM_Dribbling          5.956804   0.314500  18.941  < 2e-16 ***
TNM_Finishing          9.808877   0.280676  34.947  < 2e-16 ***
TNM_Free.kick.accuracy 2.716243   0.202098  13.440  < 2e-16 ***
TNM_GK.diving         -0.202428   0.056347  -3.593 0.000329 ***
TNM_GK.handling       -0.282977   0.055391  -5.109 3.30e-07 ***
TNM_GK.kicking        -0.237560   0.053288  -4.458 8.35e-06 ***
TNM_GK.reflexes       -0.196072   0.056570  -3.466 0.000530 ***
TNM_Heading.accuracy   9.761582   0.230434  42.362  < 2e-16 ***
TNM_Interceptions      1.021782   0.273308   3.739 0.000186 ***
TNM_Jumping            2.600327   0.131562  19.765  < 2e-16 ***
TNM_Long.passing      -4.429790   0.332515 -13.322  < 2e-16 ***
TNM_Long.shots         2.169266   0.246114   8.814  < 2e-16 ***
TNM_Marking            5.879911   0.277718  21.172  < 2e-16 ***
TNM_Penalties          1.715637   0.214070   8.014 1.22e-15 ***
TNM_Positioning        2.873695   0.259683  11.066  < 2e-16 ***
TNM_Reactions          7.337975   0.182583  40.190  < 2e-16 ***
TNM_Short.passing      2.700140   0.272689   9.902  < 2e-16 ***
TNM_Shot.power         2.394246   0.244918   9.776  < 2e-16 ***
TNM_Sliding.tackle     3.663203   0.353679  10.357  < 2e-16 ***
TNM_Sprint.speed       4.673068   0.221424  21.105  < 2e-16 ***
TNM_Stamina           -2.644983   0.225532 -11.728  < 2e-16 ***
TNM_Standing.tackle    6.302579   0.351750  17.918  < 2e-16 ***
TNM_Strength           6.758790   0.209720  32.228  < 2e-16 ***
TNM_Volleys            1.574903   0.226271   6.960 3.59e-12 ***
CAM                  -12.756392   3.305085  -3.860 0.000114 ***
CB                   -39.755634   2.051645 -19.377  < 2e-16 ***
CDM                   30.104203   2.889289  10.419  < 2e-16 ***
CF                   -15.560082   3.101561  -5.017 5.33e-07 ***
CM                    24.585474   2.661659   9.237  < 2e-16 ***
LM                    14.004634   3.236991   4.326 1.53e-05 ***
LW                   -21.115013   3.404589  -6.202 5.78e-10 ***
LWB                    8.566170   2.215459   3.867 0.000111 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\text{Number of values}}}$$

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

BL	BM	BN	BO	BP	BQ	BR
Overall	Model	Differen	Dif. Sq	Sum	Average	RMSE
50	74.02451	-24.0245	577.1769	352913.4	28.04016	5.295296
49	69.50033	-20.5003	420.2636			
49	65.92981	-16.9298	286.6185			
49	71.26818	-22.2682	495.8717			
50	66.99621	-16.9962	288.8712			
53	62.90815	-9.90815	98.17139			
50	73.53377	-23.5338	553.8385			
52	71.54256	-19.5426	381.9118			
50	69.80054	-19.8005	392.0614			
51	62.00817	-11.0082	121.1799			
51	77.37389	-26.3739	695.5819			
53	77.93516	-24.9352	621.7623			
59	73.66872	-14.6687	215.1714			
49	74.63216	-25.6322	657.0074			
56	68.196	-12.196	148.7425			

Figure 16 Snapshot from training dataset

And the RMSE value for testing dataset was ≈ 5.30

BL	BM	BN	BO	BP	BQ	BR
Overall	Model	Difference	Dif. Squ	Sum	Average	RMSE
46	66.8756	-20.8756	435.7908	154694.8	28.10589	5.301499
51	69.85851	-18.8585	355.6435			
54	71.20889	-17.2089	296.1459			
53	72.45209	-19.4521	378.3839			
49	66.87841	-17.8784	319.6377			
51	73.91545	-22.9155	525.1181			
52	71.25672	-19.2567	370.8211			
50	67.24432	-17.2443	297.3666			
53	74.74543	-21.7454	472.8636			
52	67.74397	-15.744	247.8725			
52	70.9267	-18.9267	358.2201			
53	73.53326	-20.5333	421.6148			

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.