

Table of Contents

<u>Objective 3: Predicting Match Outcome</u>	2
Data Understanding.....	2
Data Integration.....	4
Data Cleaning and Transformation	5
Renaming Attributes	7
Data Noise.....	7
Modelling	9
Descriptive Analysis.....	9
9.3.2 Predictive Analysis	12
Model Evaluation.....	19
Descriptive Analysis	19
Predictive Analysis	23
Model Comparison	27

attendance, goals scored, shots, possession and fouls and cards and corners and expected goals. The dataset contains pre-match performance data including betting odds and match results together with statistics that are measured prior to the match. The structured format of the data enables performance analysis of teams while also enabling predictions about outcomes and trend evaluations in the league.

Dataset 2

team	# wins	# losses	# goals	# total_yel_card	# total_red_card	# total_scoring_att	# ontarget_scoring_att	# ht_woodwork	# att_ht_goal	# att_pen_goal	# att_freekick_goal
1 Manchester United	25.0	5.0	63.0	68.0	1.0	698.0	258.0	21.0	12.0	5.0	1.0
2 Chelsea	24.0	5.0	64.0	62.0	4.0	636.0	216.0	14.0	16.0	3.0	6.0
3 Liverpool	20.0	10.0	57.0	44.0	0.0	668.0	214.0	15.0	8.0	6.0	1.0
4 Arsenal	19.0	8.0	62.0	70.0	3.0	638.0	220.0	19.0	10.0	10.0	3.0
5 Tottenham Hotspur	17.0	12.0	57.0	48.0	3.0	520.0	184.0	6.0	5.0	6.0	2.0
6 Bolton Wanderers	16.0	14.0	47.0	66.0	4.0	464.0	120.0	7.0	10.0	6.0	0.0
7 Reading	16.0	10.0	52.0	28.0	2.0	410.0	122.0	0.0	10.0	3.0	0.0
8 Blackburn Rovers	15.0	10.0	52.0	37.0	0.0	470.0	150.0	5.0	12.0	5.0	3.0
9 Everton	15.0	10.0	52.0	40.0	2.0	465.0	153.0	0.0	9.0	0.0	0.0
10 Portsmouth	14.0	12.0	45.0	46.0	1.0	525.0	146.0	0.0	12.0	2.0	0.0
11 Middlesbrough	12.0	10.0	44.0	54.0	2.0	454.0	146.0	7.0	7.0	4.0	1.0
12 West Ham United	12.0	21.0	35.0	60.0	2.0	461.0	134.0	0.0	3.0	1.0	2.0
13 Aston Villa	11.0	10.0	43.0	40.0	1.0	470.0	154.0	0.0	5.0	0.0	1.0
14 Manchester City	11.0	10.0	29.0	50.0	4.0	454.0	147.0	5.0	3.0	2.0	0.0
15 Newcastle United	11.0	17.0	30.0	60.0	1.0	454.0	140.0	11.0	0.0	5.0	0.0
16 Sheffield United	10.0	20.0	32.0	68.0	2.0	483.0	147.0	14.0	10.0	2.0	2.0
17 Mipon Athletic	10.0	20.0	37.0	73.0	3.0	474.0	147.0	0.0	0.0	3.0	1.0
18 Charlton Athletic	0.0	20.0	34.0	56.0	4.0	410.0	133.0	11.0	0.0	4.0	1.0
19 Fulham	0.0	10.0	30.0	50.0	3.0	401.0	142.0	0.0	0.0	3.0	1.0
20 Nottford	5.0	20.0	20.0	40.0	2.0	410.0	135.0	10.0	4.0	3.0	0.0
21 Manchester United	27.0	5.0	60.0	51.0	2.0	690.0	260.0	10.0	12.0	0.0	0.0
22 Chelsea	25.0	3.0	63.0	63.0	5.0	605.0	199.0	12.0	10.0	7.0	1.0
23 Arsenal	24.0	3.0	74.0	70.0	3.0	627.0	200.0	12.0	14.0	5.0	1.0
24 Liverpool	21.0	4.0	67.0	40.0	1.0	666.0	214.0	12.0	7.0	5.0	2.0
25 Everton	19.0	11.0	50.0	40.0	2.0	474.0	160.0	4.0	11.0	1.0	1.0
26 Aston Villa	16.0	10.0	71.0	54.0	4.0	511.0	162.0	11.0	10.0	0.0	5.0
27 Portsmouth	16.0	11.0	40.0	70.0	3.0	559.0	160.0	5.0	4.0	4.0	2.0
28 Blackburn Rovers	15.0	10.0	30.0	72.0	0.0	530.0	140.0	14.0	11.0	5.0	1.0
29 Manchester City	15.0	11.0	45.0	50.0	4.0	423.0	140.0	5.0	5.0	2.0	2.0
30 West Ham United	13.0	15.0	42.0	63.0	1.0	500.0	154.0	15.0	7.0	2.0	2.0
31 Newcastle United	11.0	17.0	40.0	60.0	1.0	401.0	153.0	10.0	10.0	4.0	2.0
32 Sunderland	11.0	21.0	30.0	60.0	4.0	403.0	137.0	7.0	10.0	1.0	0.0
33 Tottenham Hotspur	11.0	14.0	60.0	51.0	1.0	555.0	195.0	10.0	12.0	4.0	1.0

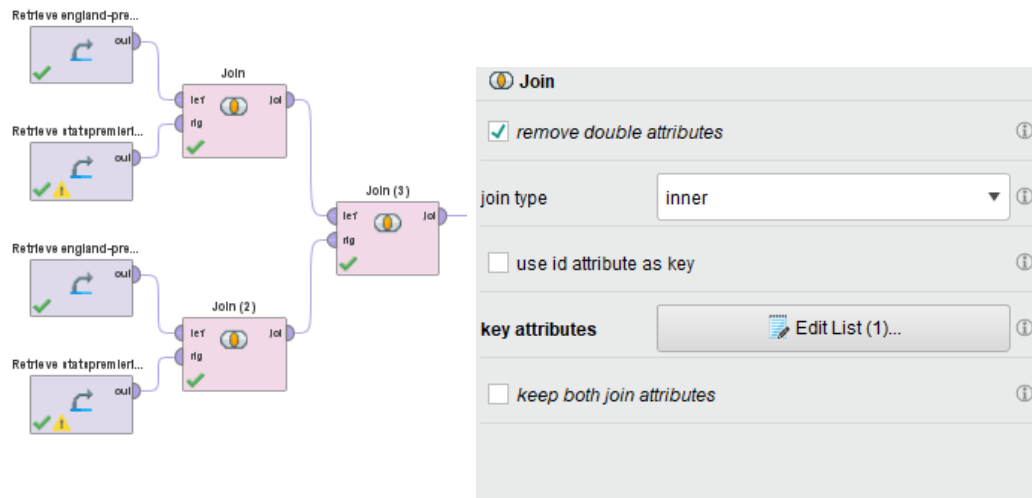
Figure 2: Stats for Premier League

Dataset about the stats of premier league matches:

<https://data.world/ericfruchi/premier-league/workspace/file?filename=statspremierleague.csv>

The dataset consists of detailed team performance statistics from years of Premier League. Key metrics there included wins, losses, goals scored and conceded, yellow and red cards, shooting accuracy, passing distribution, defensive actions and goalkeeping stats. The data has helped in analysing team strengths, playing styles, trends in different seasons, etc., which is helpful in performance evaluation and predictive analysis of matches.

Data Integration

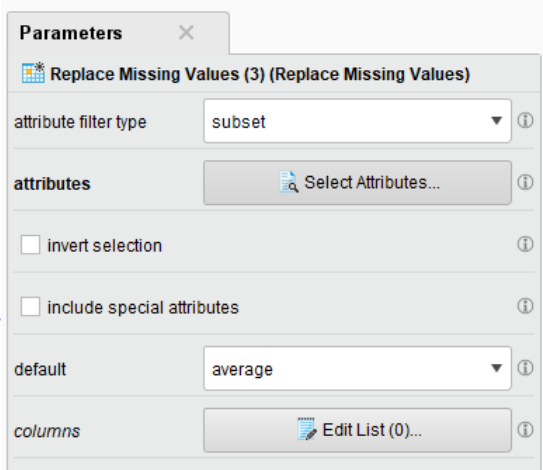
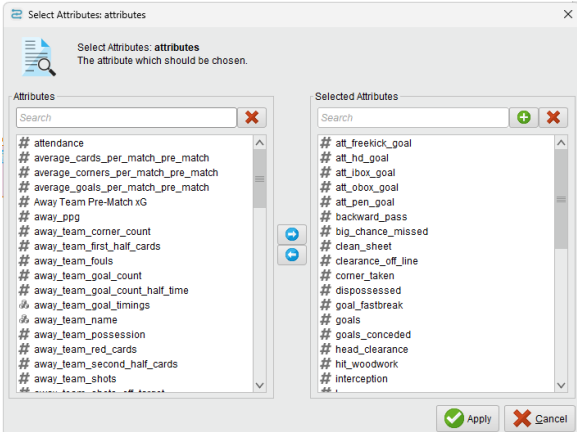


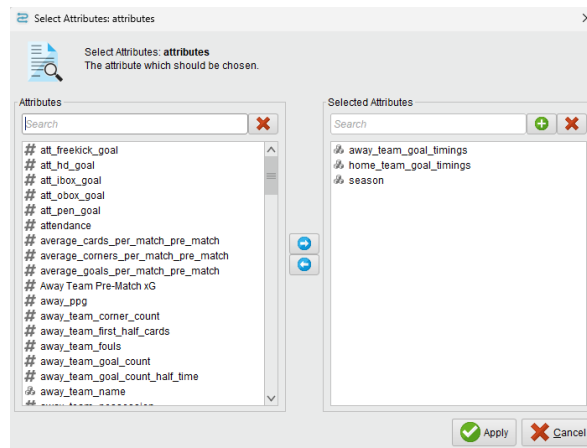
left key attributes	right key attributes
home_team_name	team

left key attributes	right key attributes
away_team_name	team

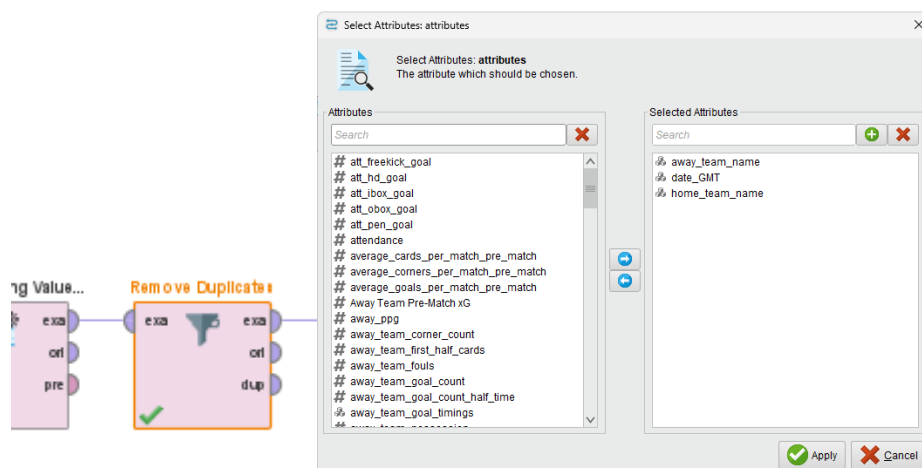
left key attributes	right key attributes
home_team_name	away_team_name

At the start, retrieve the two dataset and join them with “home_team_name” and “team”, “away_team_name” and “team”, this allows the team name to be the same, and final join is to add “home_team_name” into “away_team_name” with join as well and join type must be inner.



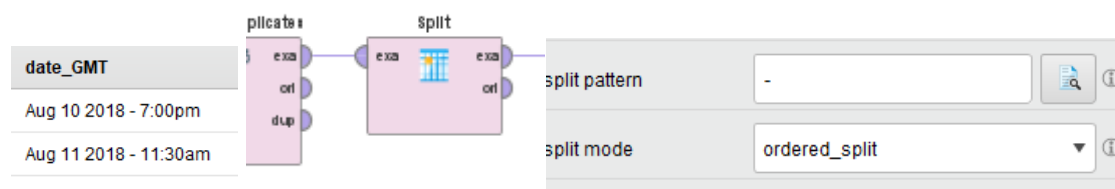


After replacing the missing value for numerical data type, add another replace missing value operator for nominal data types and set to average.



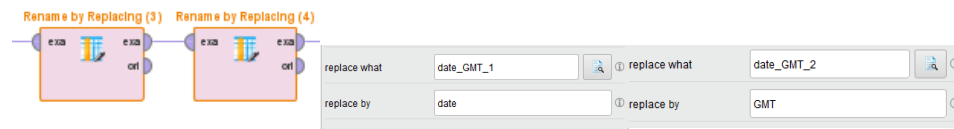
Filter (513,703 / 513,703 examples): Filter (380 / 380 examples): al

By using the remove duplicates operator and selecting the team names and dates as subsets that are duplicated from joining the datasets from the start, will change the rows from 500,000+ to just 380.

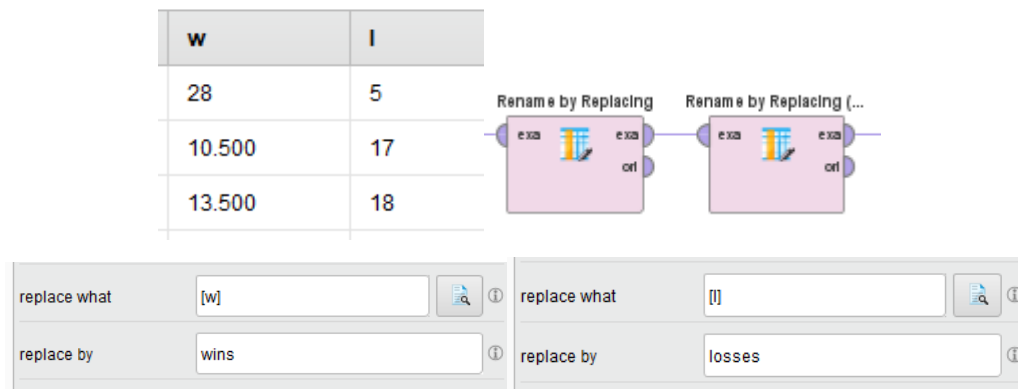


Next, using the split operator to split date and “Greenwich Mean Time” apart.

Renaming Attributes



After splitting the date and GMT, matching with the date value and GMT value respectively.



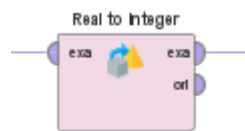
Same goes to renaming the “w” and “l”, to “wins” and “losses” respectively for clarity.

Data Noise

wins	Real	0	0	28	12.350
losses	Real	0	0	24	14.650
goals	Real	0	0	83	42.600
total_yel_card	Real	0	0	85	54.100
total_red_card	Real	0	0	4	2.200
total_scoring_att	Real	0	0	698	469.400
ontarget_scoring_att	Real	0	0	256	152.100
hit_woodwork	Real	0	0	21	10.550
att_hd_goal	Real	0	0	16	7.250
att_pen_goal	Real	0	0	10	3.850
att_freekick_goal	Real	0	0	6	1.150
att_ibox_goal	Real	0	0	72	35.300

As for checking the data types after, its founded that the data types for most of the attributes is set to real.

in	wins	losses	goals	total_yel_ca...	total_red_c...	total_scorin...	ontarget_sc...	hit_woodwo...	att_hd_goal	att_pen_goal	att_freekick...
2007	28	5	83	60	1	698	256	21	12	5	1
2007	10.500	17	38	65	1	454	140	11	8	5	0
2016	13.500	18	45	53	1	464	143	8	11	3	0
2007	7.500	15	38	56	3	451	142	8	8	3	1
2018	10.500	19	28	62	3	362	109	8	5	2	0
2007	5	20	29	45	2	418	135	10	4	3	0
2010	9.500	18	32	63	4	436	119	8	6	2	0
2007	20	10	57	44	0	668	214	15	8	6	1
2013	10	15	49	43	2	516	167	16	9	3	2
2007	19	8	63	59	3	638	226	19	10	10	3
2014	9.500	22	32	49	1	418	124	10	9	1	0
2007	15	10	52	65	2	465	153	9	9	8	2
2015	14.500	19	46	49	4	456	137	12	7	3	0
2007	17	12	57	48	3	520	184	6	5	6	2
2007	11.500	21	35	85	2	461	134	8	3	1	2
2007	23.500	3	64	62	4	636	216	14	16	3	6
2010	9	24	42	57	2	459	153	14	6	6	2
2007	10.500	18	29	59	4	454	147	5	3	2	0
2018	0	0	0	0	0	0	0	0	0	0	0
2014	19	19	33	58	2	414	143	9	6	5	1
2010	9.500	18	32	63	4	436	119	8	6	2	0
2016	13.500	18	45	53	1	464	143	8	11	3	0
2007	19	8	63	59	3	638	226	19	10	10	3
2018	10.500	19	28	62	3	362	109	8	5	2	0



The original values for wins is in decimal points, its impossible for the wins value to be in decimals, so by using the “real to integer” operator to transform the “wins” attribute’s data type into integers.

Row No.	wins	losses	goals	total_yel_ca...	total_red_ca...	total_scorin...	ontarget_sc...	clean_sheet	goals_conc...	saves	interception	total
1	28	5	83	60	1	698	256	16	27	2	254	890
2	10	17	38	65	1	454	140	7	47	5	361	965
3	13	18	45	53	1	464	143	7	67	90	640	709
4	7	15	38	56	3	451	142	7	60	12	198	901
5	10	19	28	62	3	362	109	10	58	103	519	744
6	5	20	29	45	2	418	135	9	59	17	221	744
7	9	18	32	63	4	436	119	8	56	0	537	705
8	20	10	57	44	0	668	214	20	27	1	246	969
9	10	15	49	43	2	516	167	7	60	0	779	819
10	19	8	63	59	3	638	226	12	35	6	214	998
11	9	22	32	49	1	418	124	7	74	157	491	619
12	15	10	52	65	2	465	153	14	36	5	303	860
13	14	19	46	49	4	456	137	10	55	124	750	791
14	17	12	57	48	3	520	184	6	54	11	276	995
15	11	21	35	85	2	461	134	9	59	4	233	860
16	23	3	64	62	4	636	216	22	24	4	262	982
17	9	24	42	57	2	459	153	3	82	0	661	686
18	10	18	29	59	4	454	147	14	44	0	234	986
19	0	0	0	0	0	0	0	0	0	0	0	0
20	19	19	33	58	2	414	143	12	48	119	695	845
21	9	18	32	63	4	436	119	8	56	0	537	705
22	13	18	45	53	1	464	143	7	67	90	640	709
23	19	8	63	59	3	638	226	12	35	6	214	998
24	10	19	28	62	3	362	109	10	58	103	519	744

The image shows a data workflow on the left and a 'Select Attributes: select subset' dialog box on the right.

Workflow: A 'Real to Integer' block (purple) is connected to a 'Select Attributes (2)' block (orange). Both blocks have 'exa' and 'ori' ports.

Select Attributes Dialog: The dialog has a title bar 'Select Attributes: select subset' and a subtitle 'Select Attributes: select subset Click to select the attribute subset.' It contains two lists of attributes:

- Attributes (Left List):**
 - ## att_freekick_goal
 - ## att_hd_goal
 - ## att_ibox_goal
 - ## att_oobx_goal
 - ## att_pen_goal
 - ## attendance
 - ## average_cards_per_match_pre_match
 - ## average_corners_per_match_pre_match
 - ## average_goals_per_match_pre_match
 - ## Away Team Pre-Match xG
 - ## away_team_corner_count
 - ## away_team_first_half_cards
 - ## away_team_fouls
 - ## away_team_goal_count
 - ## away_team_goal_count_half_time
 - ## away_team_goal_timings
 - ## away_team_name
 - ## away_team_possession
- Selected Attributes (Right List):**
 - ## away_ppg
 - ## clean_sheet
 - ## goals
 - ## goals_conceded
 - ## home_ppg
 - ## interception
 - ## losses
 - ## odds_ft_away_team_win
 - ## odds_ft_home_team_win
 - ## ontarget_scoring_att
 - ## saves
 - ## total_clearance
 - ## total_pass
 - ## total_red_card
 - ## total_scoring_att
 - ## total_tackle
 - ## total_yel_card
 - ## wins

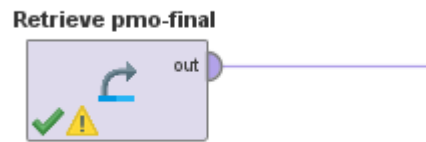
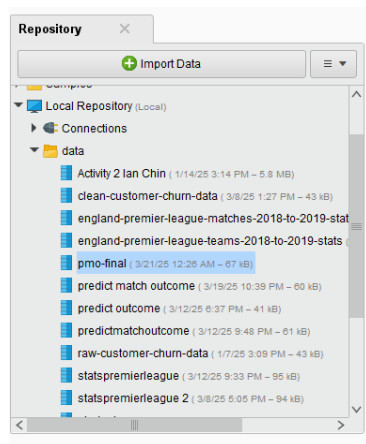
At the bottom of the dialog are 'Apply' and 'Cancel' buttons.

Modelling

Descriptive Analysis

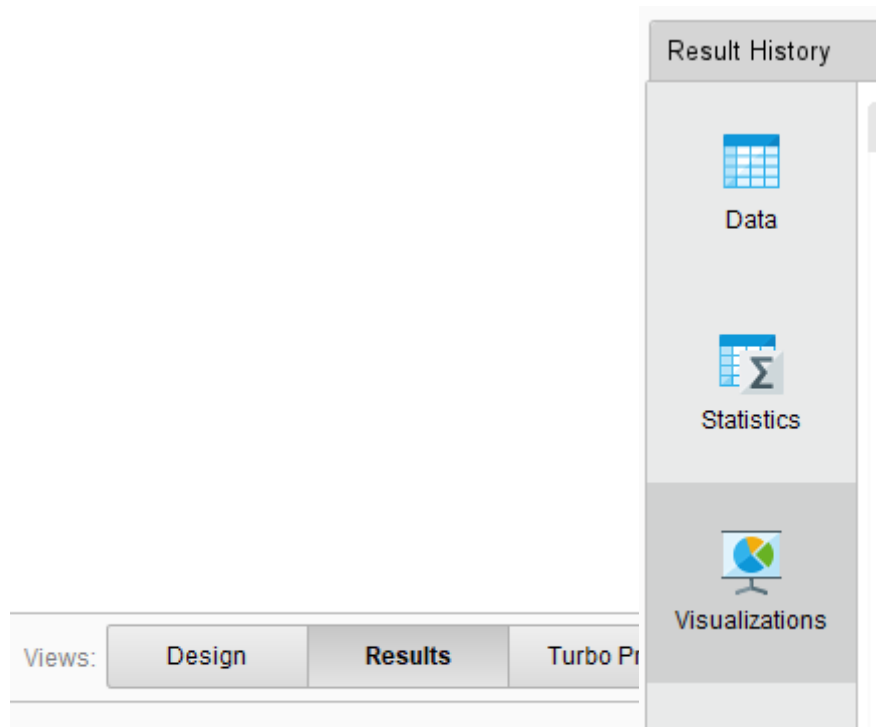
Data Mining Technique: **Data Visualization**

Data visualization helps to ease understanding of match outcome trends as a part of descriptive analysis. Can easily learn the patterns cause of home and away performances with the help of bar charts, pie charts and heatmaps. Line graphs are also useful to show trends over time, allowing the teams to base their decisions on the historical match data.

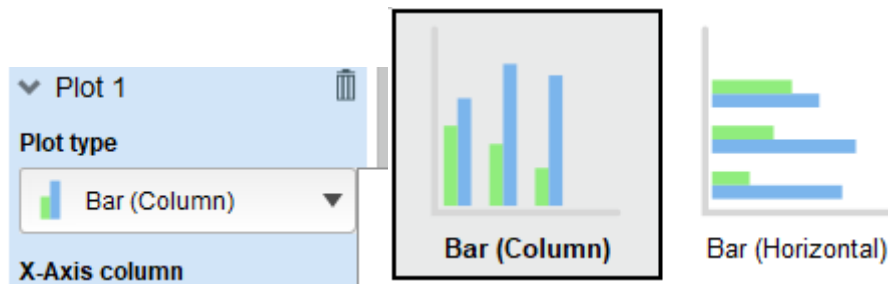


First and foremost, Retrieve Processed Data from the repository.

Here are the examples of what can possibly be filtered, this is great as it helps remove clustered of data.



To check the visualization result after filtering, click on results and then visualizations on the left and start visualizing.



For the plot type, the bar chart will be the best in slot for visualizing, either bar column or bar horizontal, both are similar options.

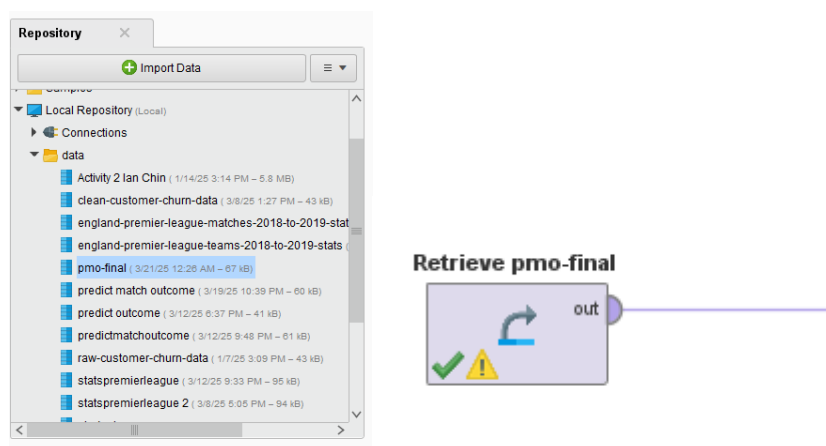
The image displays two side-by-side configuration panels for plots. Both panels have a 'Plot type' dropdown set to 'Bar (Column)'.
Plot 1 configuration:
 - **Value columns:** 'wins, losses'
 - **Aggregate data:** ☒
 - **Group by:** 'home_team_name'
 - **Aggregation Function:** 'Average'
 - **Sorting:** 'None'
 - **Color Group:** '-'
 - **Stacking:** 'No stacking'
 - **Plot style:** A button with a double arrow icon.
Plot 2 configuration:
 - **Value columns:** 'wins, losses'
 - **Aggregate data:** ☒
 - **Group by:** 'away_team_name'
 - **Aggregation Function:** 'Average'
 - **Sorting:** 'None'
 - **Color Group:** '-'
 - **Stacking:** 'No stacking'
 - **Plot style:** A button with a double arrow icon.

Furthermore, this will be the final plot for the chart, it will show the **average number of wins and losses** when teams play at home and the **average number of wins and losses** when teams play away.

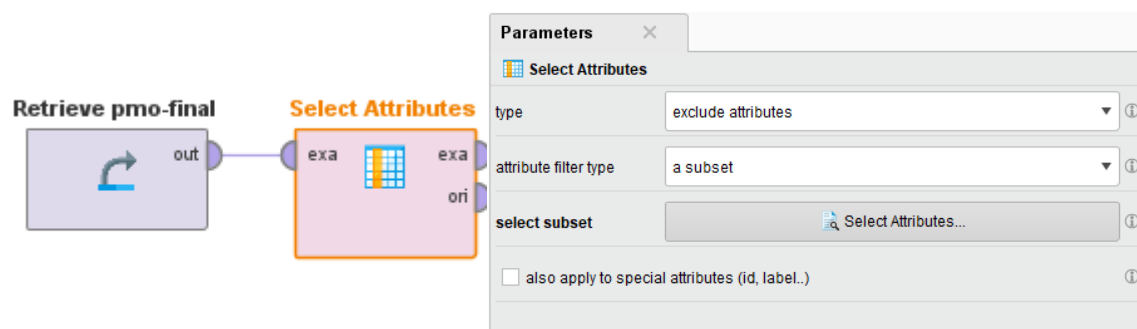
9.3.2 Predictive Analysis

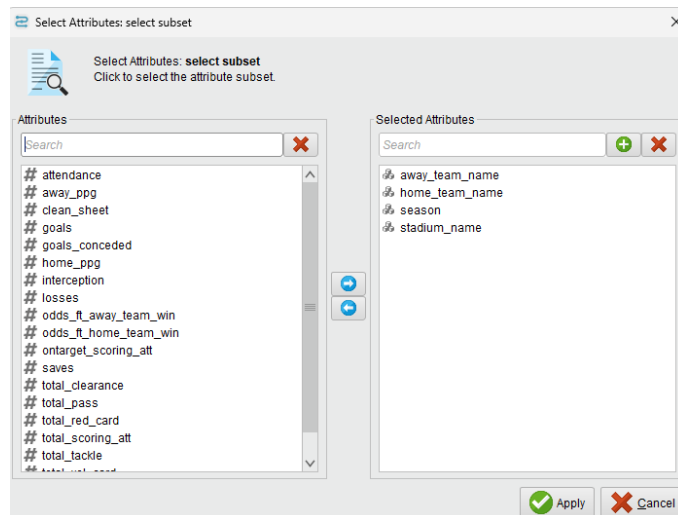
Data Mining Technique: **Linear Regression**

The supervised learning algorithm which we use for regression mostly and hence considered a fundamental algorithm is linear regression. Fitting the relationship between independent variables (predictors) and a dependent variable (outcome) to the data is done by this model using a linear equation. That algorithm aims to minimize the difference between the actual values and the predicted ones by minimizing the regression coefficients. Here, past performance metrics and other relevant factors are used to predict squad's winning percentage in the upcoming season by means of linear regression.

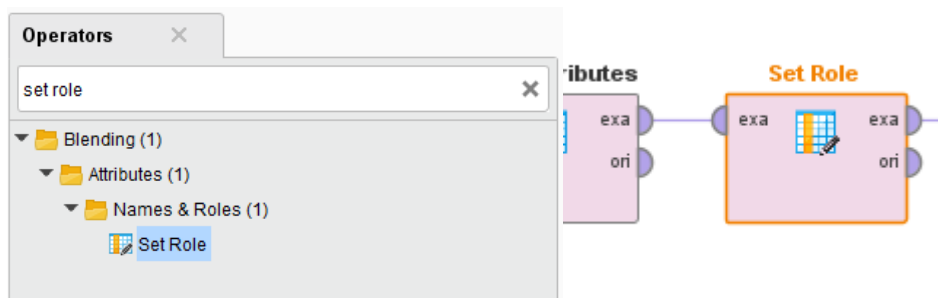


As always, retrieve the processed data from repository.

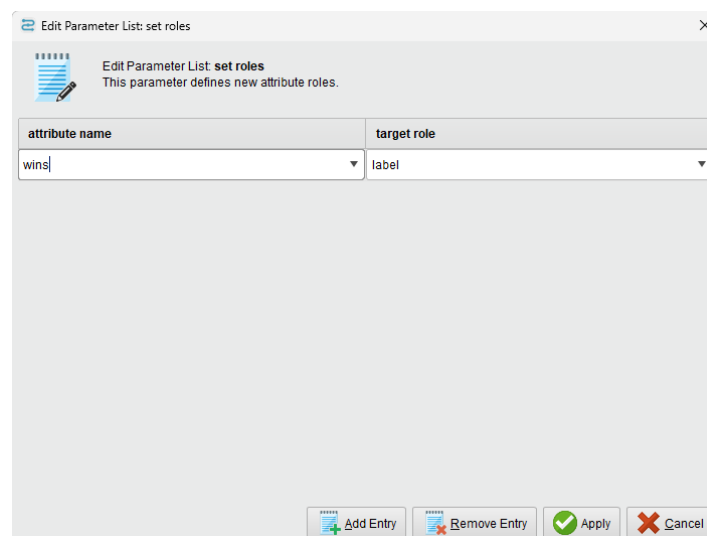




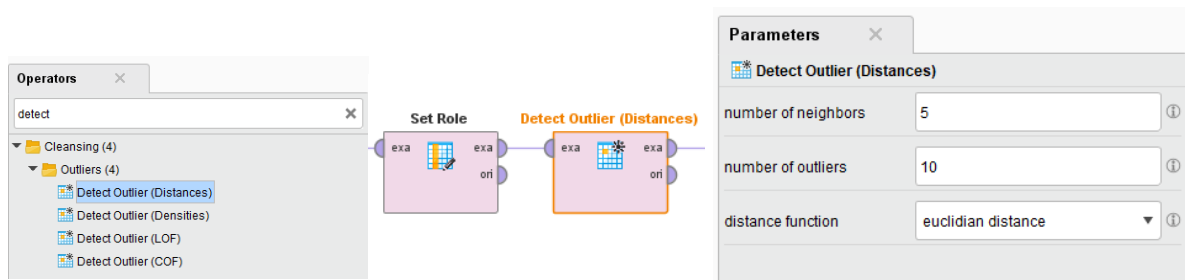
For the predictive model exclude all the attribute that is related to descriptive analysis, because don't be needing those in predictive.



Find the “Select Role” operator in the operator panel and drag into the process workflow and connect both together.



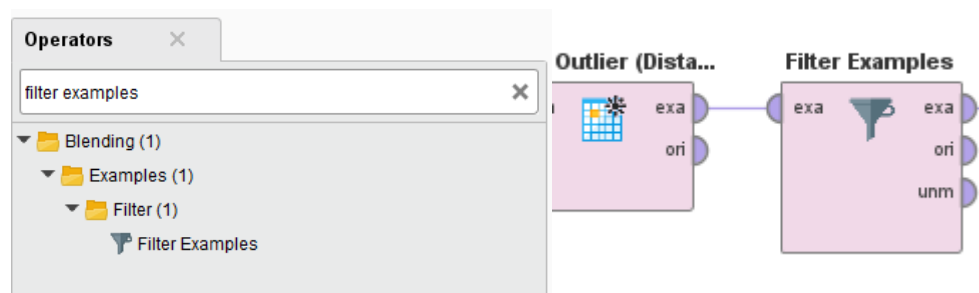
After, assigning the “wins” attribute name as a “label” for target role.

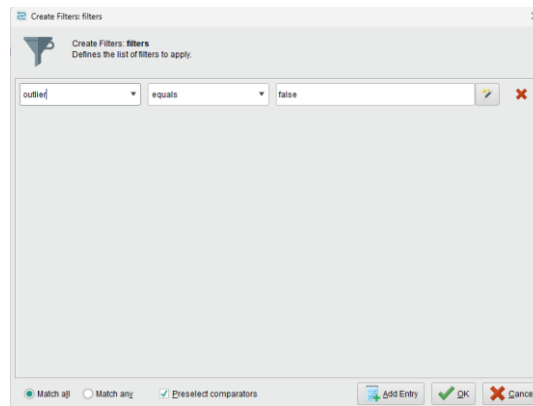


In order to find outliers, search detect outlier (distances) operators, drag and drop to the workflow and set the parameters of the number of neighbours to 5 as the dataset is not too much.

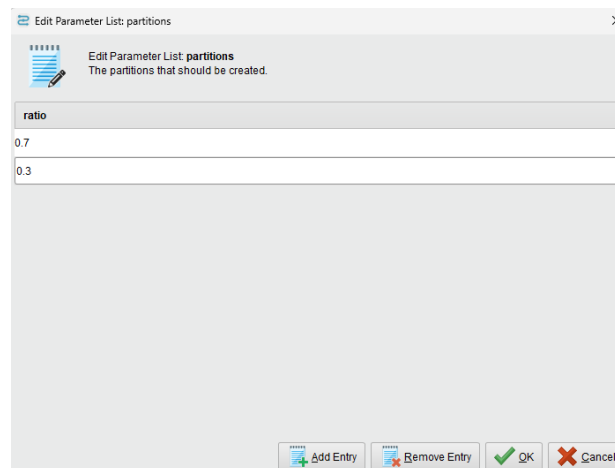
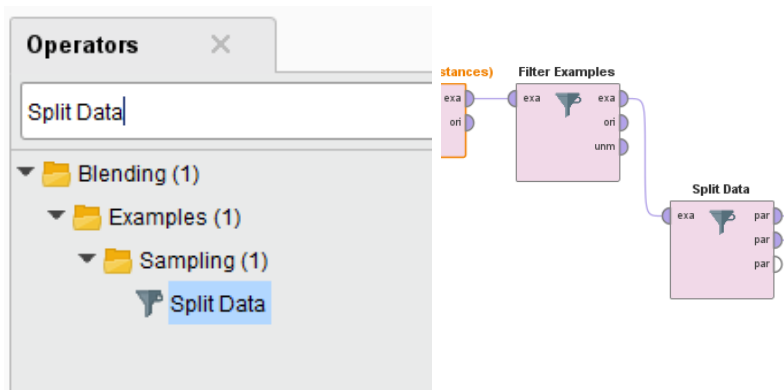
	wins	outlier	losses	goals	to
	10	false	15	49	43
	7	false	15	38	56
	9	false	24	42	57
	9	true	22	32	48
	19	false	19	33	58
	14	false	19	46	48
	20	false	10	57	44
	28	false	5	83	60

After running the process and check, there will be some outliers detected and now its needed to be filtered out.

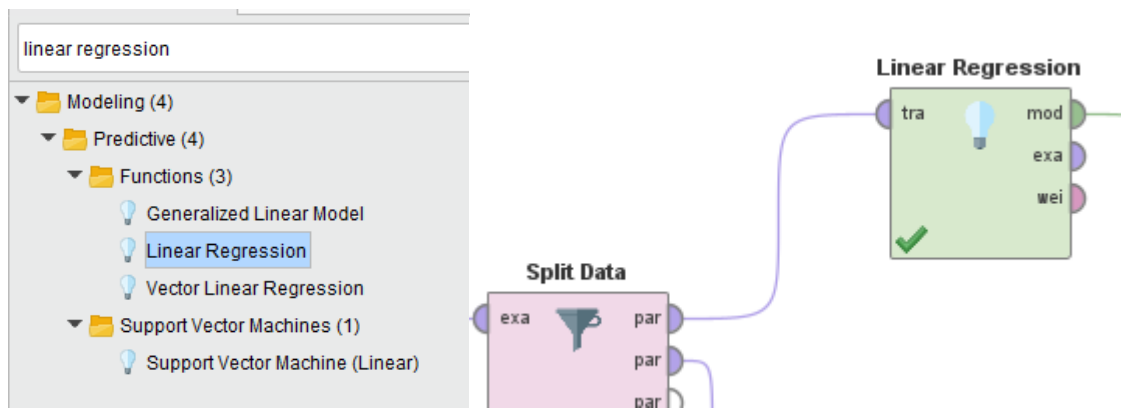




After detecting the outliers, find the filter examples in the operators panel and drag into the workflow, filter out the outlier by using “outliers, equals, false”, this will exclude all the outliers that are true.



Further continue by adding the “Split Data” operator and dragging it in the process workflow and set the ratio to 0.7 (70%) for training and 0.3 (30%) for testing then apply.

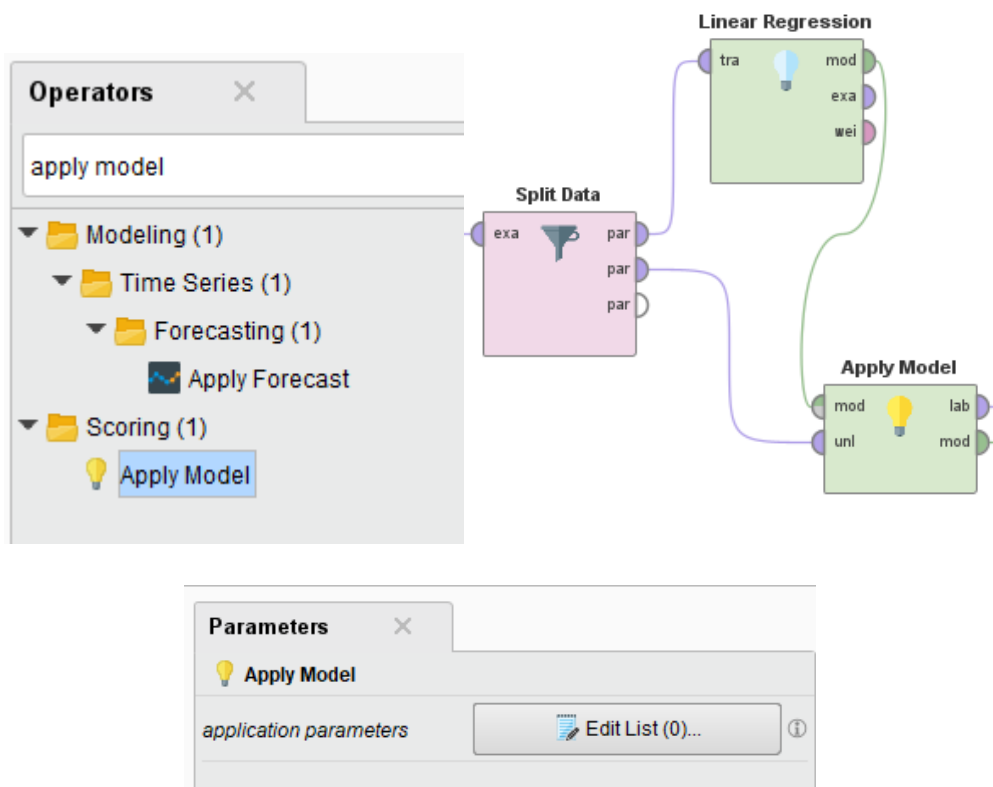


Next for the training side, find the linear regression operator in the operator panel, drag and drop into the process workflow.

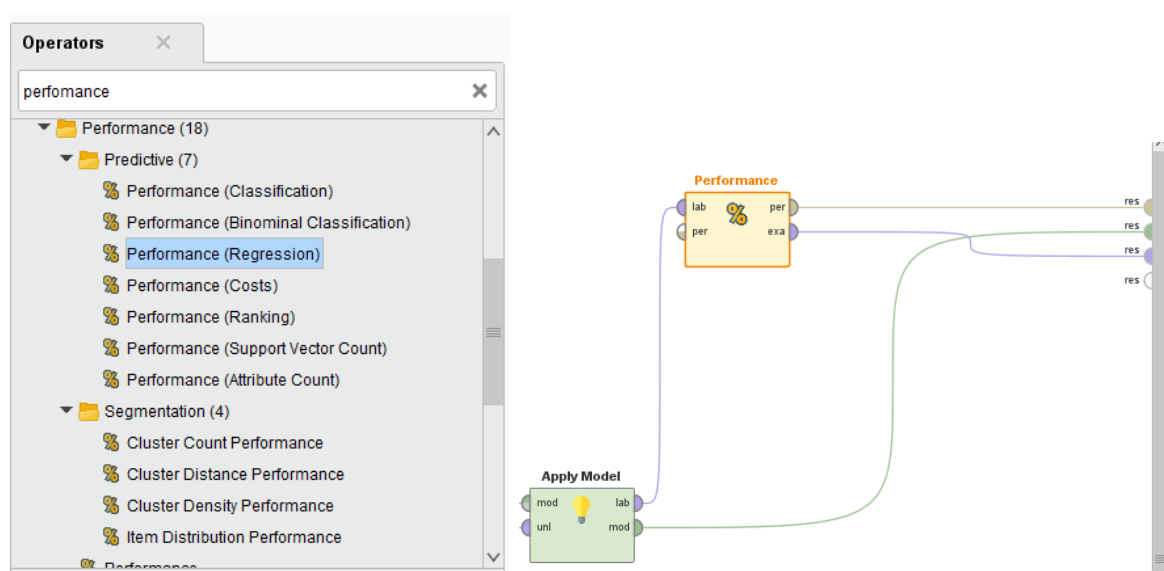
The image shows a 'Parameters' dialog box for the 'Linear Regression' operator. The dialog has a title bar with a close button. Below the title, there is a lightbulb icon and the text 'Linear Regression'. The parameters are listed in a table-like structure:

Parameter	Value	Info
feature selection	M5 prime	i
<input checked="" type="checkbox"/> eliminate colinear features		i
min tolerance	0.05	i
<input checked="" type="checkbox"/> use bias		i
ridge	1.0E-8	i

Select the M5 prime as feature selection and min tolerance set to 0.05.



As for applying the model, link it with the testing data of 20% and after linear regression, leave the parameters as it is.



For statistical performance evaluation of regression, find the “Performance(regression)” operator in the operators panel, drag and drop in the workflow and connect both performance set.

Parameters

Performance (Performance (Regression))

main criterionfirst

☒ root mean squared error

☒ absolute error

☒ relative error

☐ relative error lenient

☐ relative error strict

☐ normalized absolute error

☒ root relative squared error

☐ squared error

☒ correlation

☒ squared correlation

[Hide advanced parameters](#)

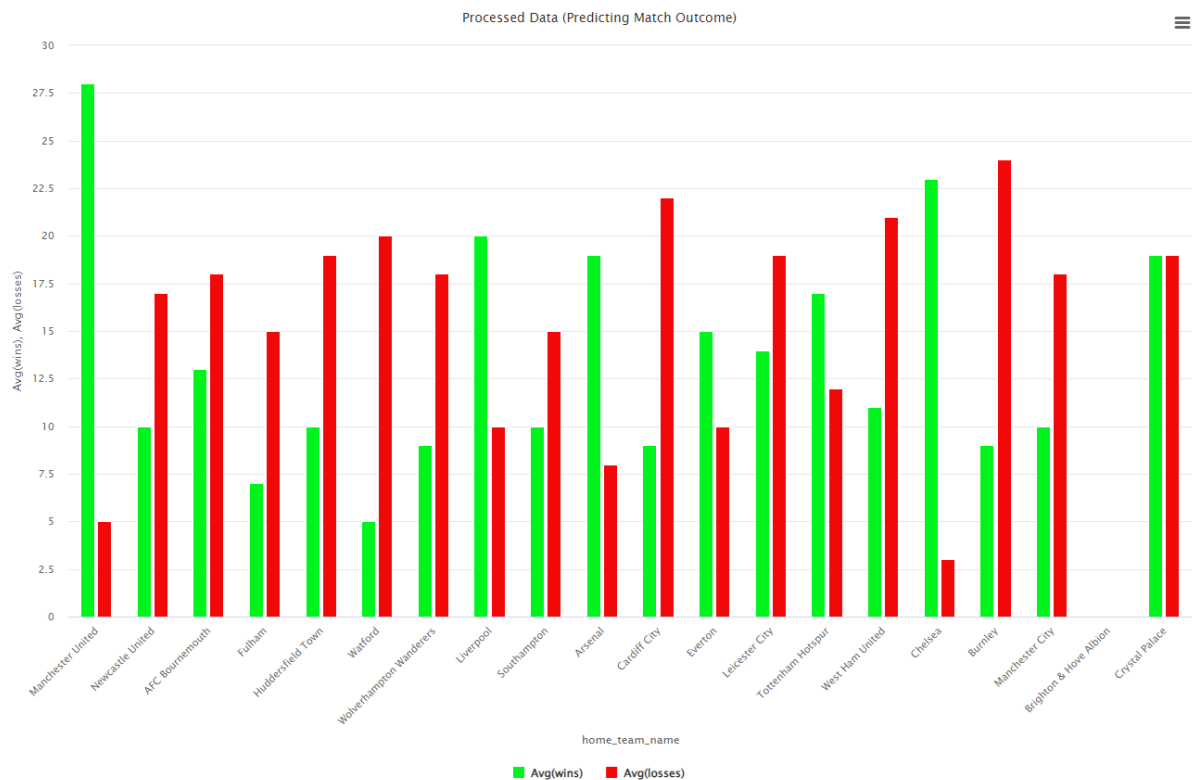
For the parameters, check absolute error, relative error, root relative squared error to have a clear understanding of how well the model performs. Hence, also check correlation and squared correlation to focus more on correlation-based metric

Model Evaluation

Descriptive Analysis

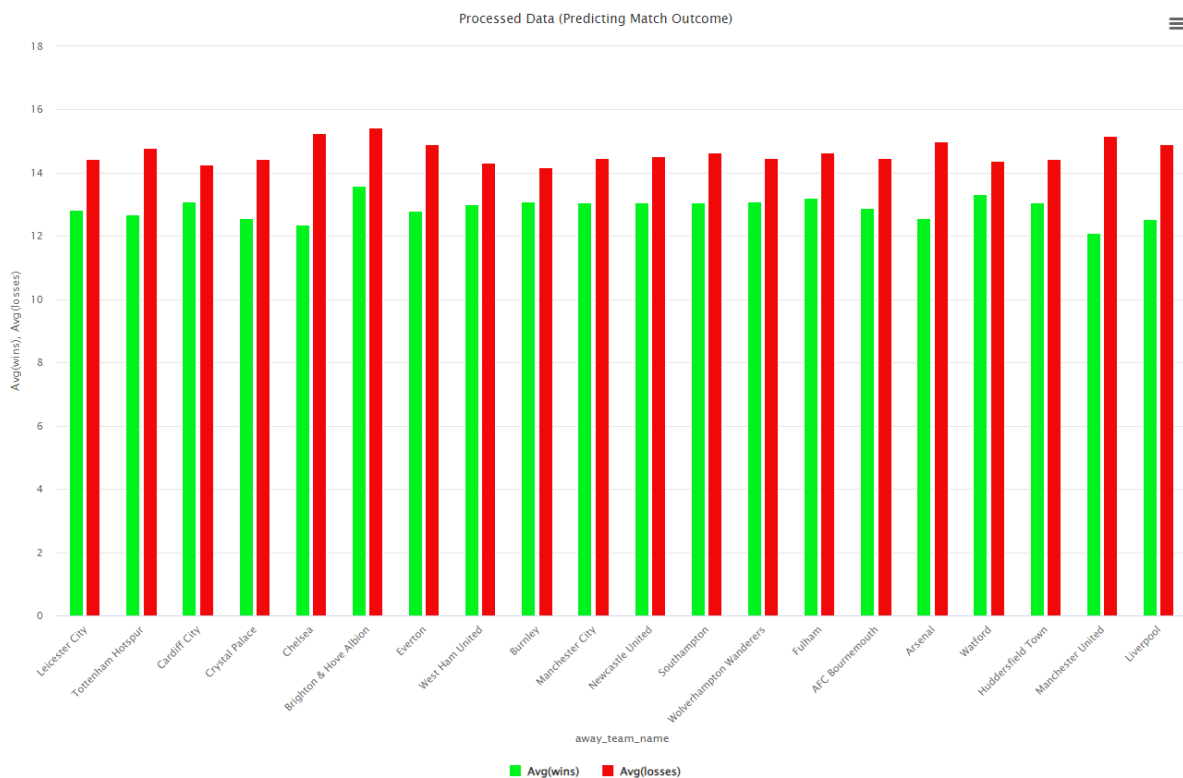
Comparing average wins and losses for teams based on home or away status

Home team match outcome

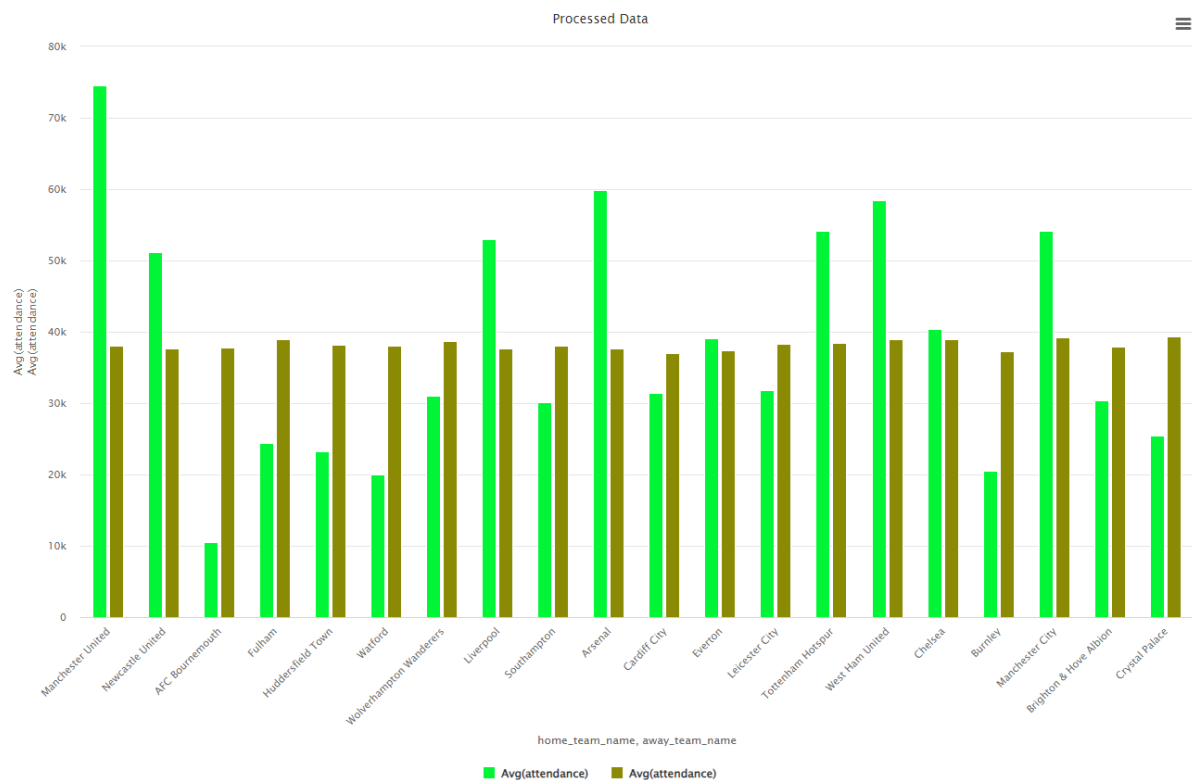


The figure above shows the average number of wins (Green bar) and losses (Red bar) for teams that are playing at home status. This bar shows that how teams will perform when being in their home stadium, generally the home team will have a more advantage over because of the familiar conditions and supportive of fans. For example, Manchester United, Arsenal and Chelsea etc, shows a higher average win comparing with the average losses. On the other hand, Cardiff City, Burnley and Huddersfield Town are struggling, by having more losses at home. Furthermore, Liverpool and Manchester city are really maintaining the consistency of high wins being in the home stadium, while the teams like Watford and Fulham got much of a high loss. The range of graphs indicates that some teams are able to use home field advantage while others aren't.

Away team match outcome



In this visualization of the second bar chart displays the same as the first bar chart, but for teams that are playing at away stadium. The statistics results shows that the number of away matches result has more losses (Red bar) than wins (Green bar). So, this indicates that teams that are playing at away stadium averages loss more than wins due to the unfamiliar conditions in locations. The majority of teams including Arsenal, Manchester United and Liverpool suffer more losses while playing matches outside their home stadium. Most clubs find winning matches outside their stadium facilities to be challenging based on the general trend data.



In figure above shows the average attendance of football matches for home and away teams, this bar chart is sharing how fans company their football teams in different stadiums. First, the x axis has various football teams and the y axis is average number of spectators. The numbers of home attendance are shown in green bars, the numbers of away attendance in brown bars. Within the chart it is evident that Manchester United, Arsenal and Chelsea teams in particular do much more than draw crowd to their stadium due to fan support. On the other hand, teams like Huddersfield Town and AFC Bournemouth have lower attendance figures, which means smaller number of fans supporting them or small stadium capacity. What is particularly evident in the Relatively lower away attendance across teams which is indicative of the fact that fewer traveling fans than home supporters. This visualization efforts have successfully shown the effect of home stadiums on fan attendance, which was already an established phenomenon that teams experience greater benefit from stronger fans show when playing at home.

Overall Visualization



The overview of both visualization of bar chart, combining the home team and away team status data, provides a clearer and clean view of each team's overall performance. Furthermore, allowing an easy comparison between average wins and losses when having home or away status. This final chart of data visualization helps in understanding that majority of match outcome of teams that are playing at home stadiums and winning will be higher than teams that are playing at away stadiums when it comes to winning.

LinearRegression

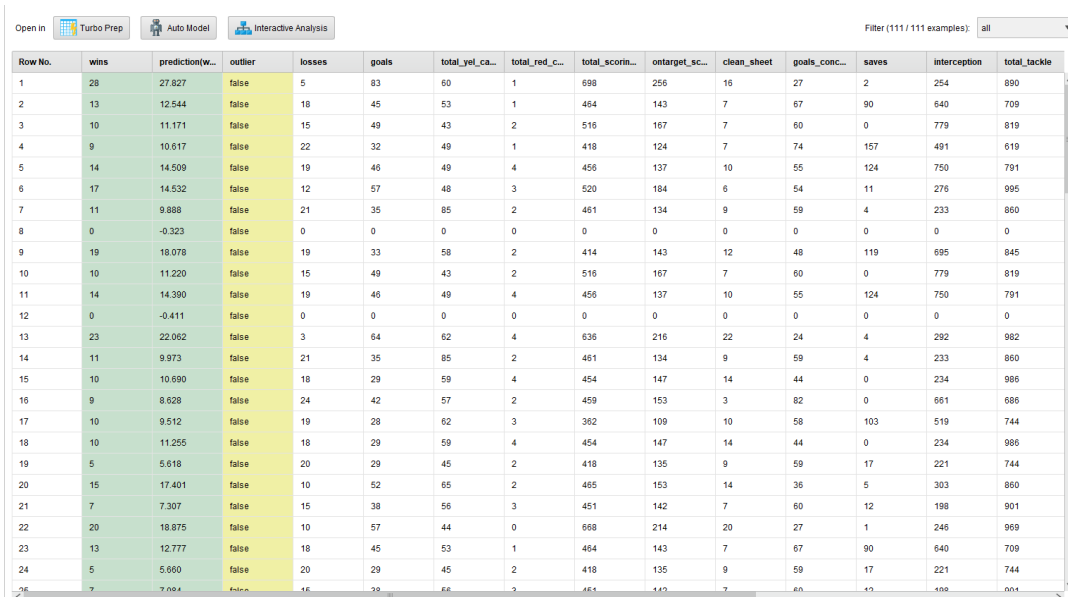
```
- 0.103 * losses
+ 0.120 * goals
+ 0.143 * total_yel_card
- 0.567 * total_red_card
- 0.034 * total_scoring_att
+ 0.135 * ontarget_scoring_att
- 0.146 * goals_conceded
+ 0.044 * saves
+ 0.006 * interception
- 0.002 * total_tackle
+ 0.003 * total_clearance
- 0.000 * total_pass
+ 1.521 * home_ppg
+ 0.356 * away_ppg
- 0.097 * odds_ft_home_team_win
+ 0.045 * odds_ft_away_team_win
- 2.296
```

The structure of this Linear Regression model is to predict football match outcome given various performance metrics. For instance, positive coefficients will be signs of positive impact on the predicted result (such as goals, goals per game sevocn, etc.: +0.120) and negative coefficients will indicate negative impact (such as total red cards, goals conceded, etc.: -0.567, -0.146). Significantly, red cards have the biggest impact in terms of negative results. Smaller effects are shown from defensive metrics like saves and interceptions, and on total passes (0.000) there is no influence. Overall, the models highlights that scoring goals, nailing a good home record and avoiding disciplinary issues are key to success.

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 1.102 +/- 0.000
absolute_error: 0.923 +/- 0.602
relative_error: 7.88% +/- 4.83%
root_relative_squared_error: 0.171
squared_error: 1.214 +/- 1.579
correlation: 0.985
squared_correlation: 0.971
```

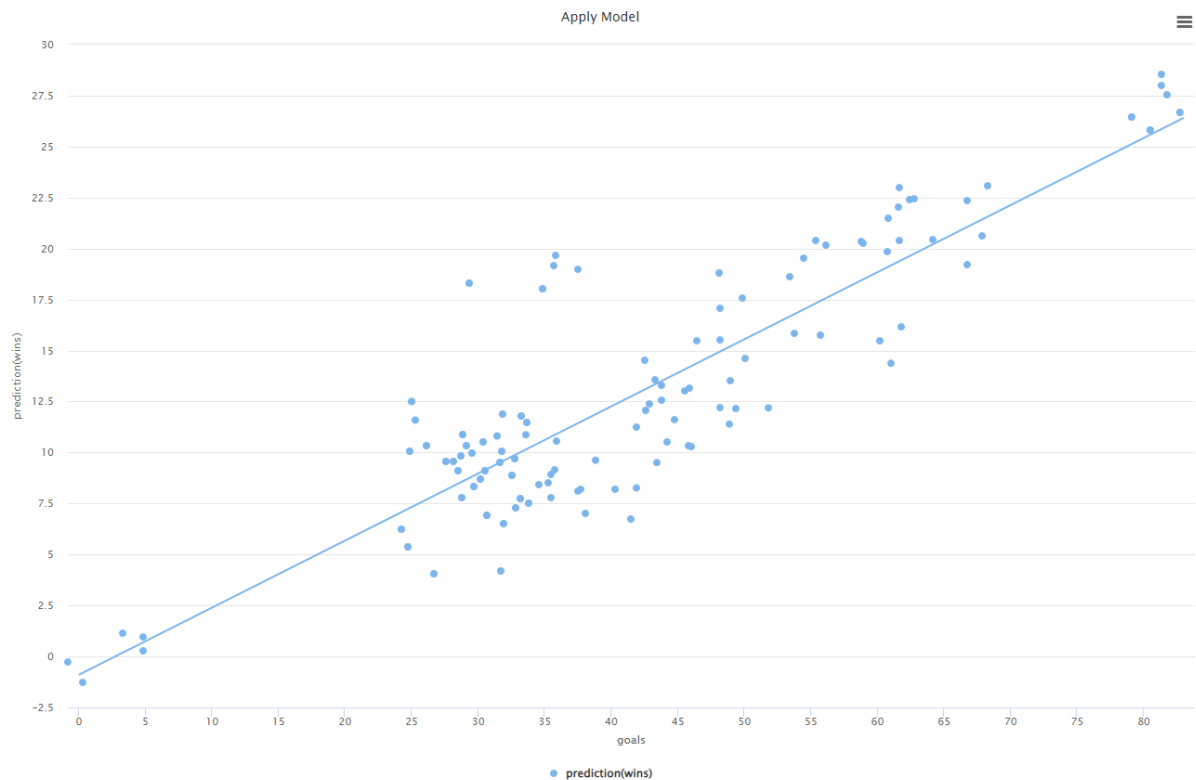
As seen from low RMSE (1.102) and absolute error (0.923 ± 0.602), the prediction are relatively close to actual values, and the linear regression model performs very well. It is demonstrated that the model significantly outperforms a simple mean-based predictor, with Relative error ($7.88\% \pm 4.83\%$) and Root relative squared error (0.171). Moreover, the robust predictive power of the model with as low as no error ($R^2 = 0.971$) is supported by the high correlation (0.985) and R^2 value of 0.971 for the target variable that accounts for a 97.1% variance. Together, these metrics indicate that the model is quite accurate and reliable.



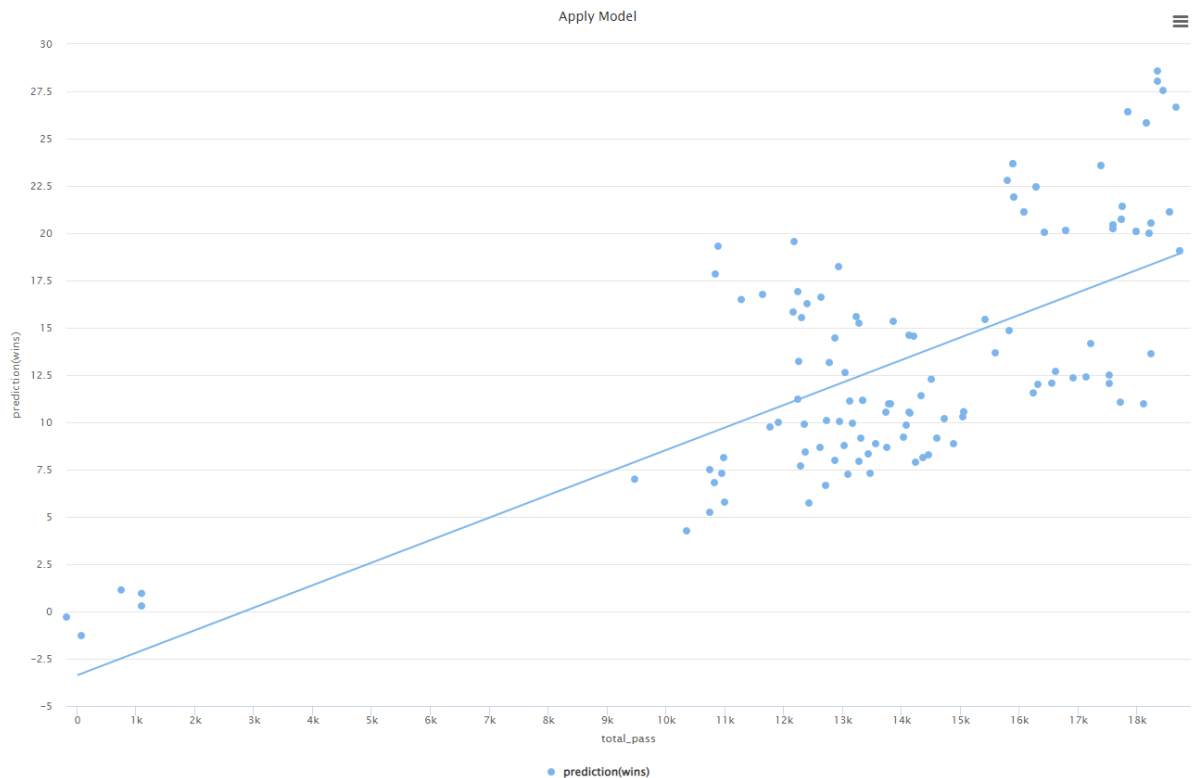
Row No.	wins	prediction(w...	outlier	losses	goals	total_yel_ca...	total_red_c...	total_scorin...	ontarget_sc...	clean_sheet	goals_conc...	saves	interception	total_tackle
1	28	27.827	false	5	83	60	1	698	256	16	27	2	254	890
2	13	12.544	false	18	45	53	1	464	143	7	67	90	640	709
3	10	11.171	false	15	49	43	2	516	167	7	60	0	779	819
4	9	10.617	false	22	32	49	1	418	124	7	74	157	491	619
5	14	14.509	false	19	46	49	4	456	137	10	55	124	750	791
6	17	14.532	false	12	57	48	3	520	184	6	54	11	276	995
7	11	9.888	false	21	35	85	2	461	134	9	59	4	233	860
8	0	-0.323	false	0	0	0	0	0	0	0	0	0	0	0
9	19	18.078	false	19	33	58	2	414	143	12	48	119	695	845
10	10	11.220	false	15	49	43	2	516	167	7	60	0	779	819
11	14	14.390	false	19	46	49	4	456	137	10	55	124	750	791
12	0	-0.411	false	0	0	0	0	0	0	0	0	0	0	0
13	23	22.062	false	3	64	62	4	636	216	22	24	4	292	982
14	11	9.973	false	21	35	85	2	461	134	9	59	4	233	860
15	10	10.690	false	18	29	59	4	454	147	14	44	0	234	986
16	9	8.628	false	24	42	57	2	459	153	3	82	0	661	886
17	10	9.512	false	19	28	62	3	362	109	10	58	103	519	744
18	10	11.255	false	18	29	59	4	454	147	14	44	0	234	986
19	5	5.618	false	20	29	45	2	418	135	9	59	17	221	744
20	15	17.401	false	10	52	65	2	465	153	14	36	5	303	860
21	7	7.307	false	15	38	56	3	451	142	7	60	12	198	901
22	20	18.875	false	10	57	44	0	668	214	20	27	1	246	969
23	13	12.777	false	18	45	53	1	464	143	7	67	90	640	709
24	5	5.660	false	20	29	45	2	418	135	9	59	17	221	744
25	7	7.094	false	16	39	66	3	464	143	7	60	12	198	901

The figure above shows the table data of the predictive model, with wins and predictions(wins) as labels and filtered out all the outliers.

Comparison between predictions wins and goals with predictions win and total pass



The figure above shows that plots Goals vs. Predicted Wins is strongly positively correlated, one with the number of goals scored and the other with the number of predicted wins. A trend line which is close and all of points are close to the trend line, means that the relationship has been consistent the more goals teams have scored, the more matches they will win. This shows that goals are a remarkably reliable predictor of success and further reinforces the study that the making of scoring opportunities directly effect a team chance of winning. Since variance is low the implication is that goal scoring forms a primary determinant for predictions of wins, therefore, it must be an area of focus for a team looking to get better.



The second graph, that is Total Passes vs Predicted Wins is also positively correlated however with a wider spread within the data points. While pass is a major point in game play, the direct impact of pass on winning is not as direct as goal score. Overall, the more passes a team makes, the more they win. However, the relationship is not as strong, suggesting that passing alone will not lead to success. It is likely that other factors such as passing accuracy taken, strategy and goal conversion rate all play a role in determining the effectiveness of passing. This analysis compared to first chart shows that although good passing strategies will help you to perform well, being able to score goals is the most important things to secure victory.

Model Comparison

Comparing Linear Regression with Deep Learning

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 1.102 +/- 0.000
absolute_error: 0.923 +/- 0.602
relative_error: 7.88% +/- 4.83%
root_relative_squared_error: 0.171
squared_error: 1.214 +/- 1.579
correlation: 0.985
squared_correlation: 0.971
```

PerformanceVector

```
PerformanceVector:
root_mean_squared_error: 0.818 +/- 0.000
absolute_error: 0.574 +/- 0.583
relative_error: 4.41% +/- 4.61%
root_relative_squared_error: 0.127
squared_error: 0.670 +/- 1.631
correlation: 0.992
squared_correlation: 0.985
```

Metrics	Linear Regression	Deep Learning	Comparison
Root Mean Squared Error (RSME)	1.102 ± 0.000	0.818 ± 0.000	RMSE on Deep Learning is lower meaning that it has better prediction accuracy.
Absolute Error	0.923 ± 0.602	0.574 ± 0.583	The absolute error value of Deep Learning is smaller, thus making a lesser deviance from the actual values.
Relative Error	7.88% ± 4.83%	4.41% ± 4.61%	This is because Deep Learning 's Relative Error is lower, being more precise overall.
Root Relative Squared Error	0.171	0.127	It means that B Deep Learning has a lower RRSE and hence better performance.
Squared Error	1.214 ± 1.579	0.670 ± 1.631	As demonstrated by Deep Learning, when less squared error is shown, then this means that better error minimization.
Correlation	0.985	0.992	Higher correlation in Deep Learning suggests a stronger relationship between predicted and actual values.
Squared Correlation (R ²)	0.971	0.985	Deep Learning has a slightly higher R ² value, indicating a better fit.

Conclusion:

Overall, Deep Learning obtains lower errors and has better correlation data than Linear Regression that was selected for objective 3, which renders it a more dependable predictive model.