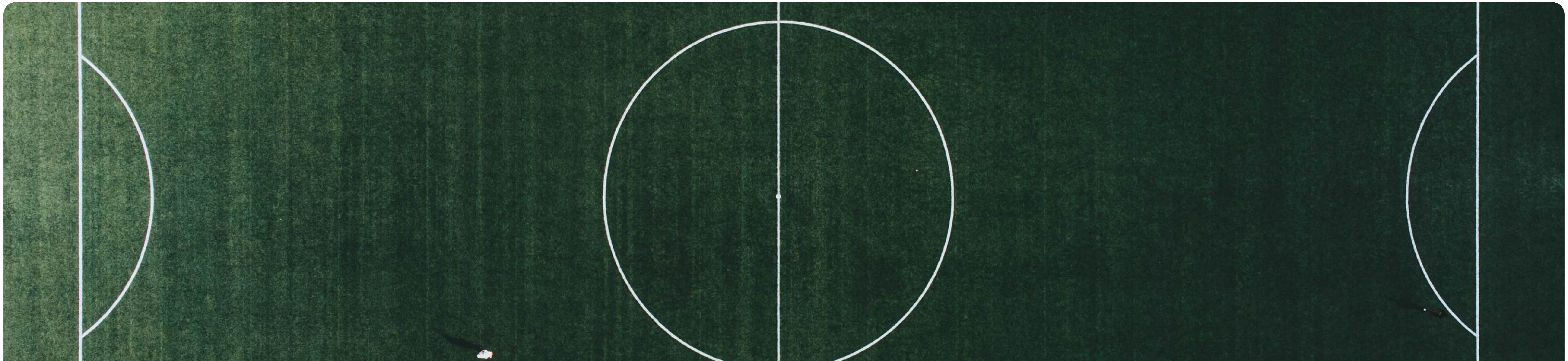
PROJECT REPORT

PREMIER LEAGUE MATCH WINNER PREDICTIONS

Part 1 is talking about the dataset and appllied operations on from cleaning, munging and applied feature engineering methods

Part 2 showing the applied analytics on the dataset and representing the gained insights using sas analytics and building ML Model to predict the winner team



Used Tools

ABOUT DATASET

This dataset is consists of 1389 recoeds where each record represent single match and 27 features. this data starting date of collection was in 2020-09-12 and ended in 2022-04-25 which stored features about team`s 'venue', 'result', 'gf', 'ga', 'opponent', 'xg', 'xga', 'poss', 'attendance', 'captain', 'formation', 'referee', 'match report', 'notes', 'sh', and 'sot'

Top Six Team IN Epl

Operations Index

Project Goals

- Showing the impact of playing matches at home versus playing away and that effects on teams' results.
- Checking if there is bias from referees toward specific teams in the English Premier League.
- Analyzing the performance of English teams from 2020-09-12 to 2022-04-25, including studying results, GF, GA, opponent, xG, xGA, possession, attendance, captain, formation, referee, match report, notes, shots, and shots on target for each team.
- Building machine learning models to predict the winning team in the English Premier League based on the collected data.

Uploading Dataset With sas

SAS CODE

*EdaAndFeatureEngineering.sas

CODE LOG RESULTS OUTPUT DATA

```
proc import datafile="/home/u63511609/BigDataFinalProject/matches.csv"
    out=permier_league
    dbms=csv
    replace;
    guessingrows=32767;
run;
```

Uploading Dataset With sas

OUTPUT

*EdaAndFeatureEngineering.sas ×

CODE LOG RESULTS OUTPUT DATA

Table: WORK.PERMIER_LEAGUE View: Column names Filter: (none)

Columns

- [x] Select all
- [x] VAR1
- [x] date
- [x] time
- [x] comp
- [x] round
- [x] day
- [x] venue
- [x] result
- [x] gf
- [x] ga

Total rows: 1389 Total columns: 28 Rows 1-100

	VAR1	date	time	comp	round	day	venue	result	gf	ga	opponent
1	1	2021-08-15	16:30:00.0	Premier League	Matchweek 1	Sun	Away	L	0	1	Tottenhan
2	2	2021-08-21	15:00:00.0	Premier League	Matchweek 2	Sat	Home	W	5	0	Norwich C
3	3	2021-08-28	12:30:00.0	Premier League	Matchweek 3	Sat	Home	W	5	0	Arsenal
4	4	2021-09-11	15:00:00.0	Premier League	Matchweek 4	Sat	Away	W	1	0	Leicester C
5	6	2021-09-18	15:00:00.0	Premier League	Matchweek 5	Sat	Home	D	0	0	Southamp
6	8	2021-09-25	12:30:00.0	Premier League	Matchweek 6	Sat	Away	W	1	0	Chelsea
7	10	2021-10-03	16:30:00.0	Premier League	Matchweek 7	Sun	Away	D	2	2	Liverpool
8	11	2021-10-16	15:00:00.0	Premier League	Matchweek 8	Sat	Home	W	2	0	Burnley
9	13	2021-10-23	17:30:00.0	Premier League	Matchweek 9	Sat	Away	W	4	1	Brighton
10	15	2021-10-30	15:00:00.0	Premier League	Matchweek 10	Sat	Home	L	0	2	Crystal Pa
11	17	2021-11-06	12:30:00.0	Premier League	Matchweek 11	Sat	Away	W	2	0	Manchest

EDA : checking if there are null values

OUTPUT

The MEANS Procedure

Variable	N Miss
VAR1	0
date	0
time	0
gf	0
ga	0
xg	0
xga	0
poss	0
attendance	696
sh	0
sot	0
dist	1
fk	0
pk	0
pkatt	0
season	0

SAS CODE

```
proc means data=permier_league nmiss n;
    var _numeric_;
    output out=numeric_missing_summary
        nmiss=Num_Missing
        n=Num_Total;
run;
```

EDA : checking if there are duplicated values

SAS CODE

```
proc sort data=permier_league out=sorted_permier_league nodupkey dupout=duplicates;
    by _all_;
run;

proc sql;
    select count(*) as duplicate_count
    from duplicates;
quit;
```

OUT PUT

duplicate_count
0

Dataset Has **No** Duplicated Values

EDA : checking dtype of each feature

SAS CODE

```
proc contents data=permier_league;
run;
```

OUTPUT

Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
1	VAR1	Num	8	BEST12.	BEST32.
15	attendance	Num	8	BEST12.	BEST32.
16	captain	Char	25	$25.	$25.
4	comp	Char	14	$14.	$14.
2	date	Num	8	YYMMDD10.	YYMMDD10.
6	day	Char	3	$3.	$3.
23	dist	Num	8	BEST12.	BEST32.
24	fk	Num	8	BEST12.	BEST32.
17	formation	Char	10	$10.	$10.
10	ga	Num	8	BEST12.	BEST32.
9	gf	Num	8	BEST12.	BEST32.
31	hour	Num	8		
19	match report	Char	12	$12.	$12.
20	notes	Char	1	$1.	$1.
30	opp_code	Char	2		
11	opponent	Char	15	$15.	$15.
25	pk	Num	8	BEST12.	BEST32.
26	pkatt	Num	8	BEST12.	BEST32.
14	poss	Num	8	BEST12.	BEST32.

OUTPUT

18	referee	Char	17	$17.	$17.
8	result	Char	1	$1.	$1.
5	round	Char	12	$12.	$12.
27	season	Num	8	BEST12.	BEST32.
21	sh	Num	8	BEST12.	BEST32.
22	sot	Num	8	BEST12.	BEST32.
32	target	Num	8		
28	team	Char	24	$24.	$24.
3	time	Num	8	TIME20.3	TIME20.3
7	venue	Char	4	$4.	$4.
29	venue_code	Num	8		
12	xg	Num	8	BEST12.	BEST32.
13	xga	Num	8	BEST12.	BEST32.

EDA : frequency of each team

SAS CODE

```
proc freq data=permier_league;
    tables team / nocum nopercent;
run;
```

OUTPUT

The FREQ Procedure

team	Frequency
Arsenal	71
Aston Villa	70
Brentford	34
Brighton and Hove Albion	72
Burnley	71
Chelsea	70
Crystal Palace	71
Everton	70
Fulham	38
Leeds United	71
Leicester City	70
Liverpool	38

OUTPUT

Manchester City	71
Manchester United	72
Newcastle United	72
Norwich City	33
Sheffield United	38
Southampton	72
Tottenham Hotspur	71
Watford	33
West Bromwich Albion	38
West Ham United	72
Wolverhampton Wanderers	71

APPLIED DATA MANIPULATIONS METHODS

Data Manipulation

Calculating total shots, The total number of goals scored by the team, The total number of goals conceded by the team, The number of shots that were directed towards the goal, The number of free kicks awarded to the team, The number of penalty kicks successfully converted into goals,The number of penalty kick attempts made by the team

SAS CODE

```
proc means data=permier_league sum;
    class team;
    var sh gf ga sot dist fk pk  pkatt ;
    output out=team_summary sum=;
run;

/* Printing the summarized dataset */
proc print data=team_summary;
    where _TYPE_ = 1;
    var team sh gf ga sot dist fk pk  pkatt ;
run;
```

Data Manipulation

- sh - Shots: The total number of attempts made by a team or player to score a goal.
- gf - Goals For: The total number of goals scored by the team.
- ga - Goals Against: The total number of goals conceded by the team.
- sot - Shots on Target: The number of shots that were directed towards the goal and would have gone in if not for a save or a block.
- fk - Free Kicks: The number of free kicks awarded to the team.
- pk - Penalty Kicks Scored: The number of penalty kicks successfully converted into goals.
- pkatt - Penalty Kicks Attempted: The number of penalty kick attempts made by the team

OUTPUT

Obs	team	sh	gf	ga	sot	dist	fk	pk	pkatt
2	Arsenal	959	107	79	296	1213.3	42	10	13
3	Aston Villa	898	97	92	306	1184.9	33	8	9
4	Brentford	379	41	49	119	549.1	7	6	6
5	Brighton and Hove Albion	894	71	88	243	1210.7	29	9	14
6	Burnley	727	62	100	222	1181.6	27	3	4
7	Chelsea	1025	125	63	359	1182.1	38	15	18
8	Crystal Palace	693	84	107	247	1141.4	30	8	11
9	Everton	764	81	103	246	1184.7	31	9	11
10	Fulham	440	27	53	123	671.7	10	3	6
11	Leeds United	962	100	122	326	1224.1	21	8	8
12	Leicester City	838	115	101	303	1252.4	34	12	14
13	Liverpool	600	68	42	201	626.8	20	6	6
14	Manchester City	1185	163	53	420	1158.4	34	12	17
15	Manchester United	983	126	95	360	1232.3	37	12	15
16	Newcastle United	796	86	117	256	1257.3	37	8	9
17	Norwich City	327	22	69	92	594.1	17	3	3
18	Sheffield United	319	20	63	92	635.6	5	3	4
19	Southampton	863	87	124	306	1246	40	8	9
20	Tottenham Hotspur	857	124	83	319	1218.6	51	8	8
21	Watford	352	31	67	115	591.8	20	1	2
22	West Bromwich Albion	336	35	76	107	675.2	16	4	4
23	West Ham United	875	114	91	289	1119	29	5	9
24	Wolverhampton Wanderers	809	69	81	266	1260.9	25	5	5

sas

From 2020-09-12 to 2022-04-25

Calculating average expected goals , The average distance in yards from which shots were taken, The average expected goals against, and the average possession percentage

SAS CODE

```
proc means data=permier_league noprint;
    class team;
    var dist xg xga poss;
    output out=result_mean mean=dist_mean xg_mean xga_mean poss_mean;
run;

proc print data=result_mean;
run;
```

Data Manipulation

- dist - Distance: The average distance (in meters or yards) from which shots were taken.
- xg - Expected Goals: A metric that estimates the likelihood of a shot resulting in a goal based on factors like shot angle, distance, and type.
- xga - Expected Goals Against: The expected number of goals that the team was likely to concede based on the quality of shots taken by the opposition.
- poss - Possession Percentage: The average percentage of time the team controlled the ball during the game.

OUTPUT

Obs	team	_TYPE_	_FREQ_	dist_mean	xg_mean	xga_mean	poss_mean
1		0	1389	17.011527378	1.3041756659	1.3384449244	49.702663787
2	Arsenal	1	71	17.088732394	1.4873239437	1.1845070423	53.112676056
3	Aston Villa	1	70	16.927142857	1.2742857143	1.3157142857	47.442857143
4	Brentford	1	34	16.15	1.2117647059	1.2794117647	44
5	Brighton and Hove Albion	1	72	16.815277778	1.2291666667	1.1083333333	53.277777778
6	Burnley	1	71	16.642253521	1.0056338028	1.5028169014	41.056338028
7	Chelsea	1	70	16.887142857	1.7142857143	0.9042857143	60.7
8	Crystal Palace	1	71	16.305714286	1.0295774648	1.3338028169	45.225352113
9	Everton	1	70	16.924285714	1.1785714286	1.3871428571	44.1
10	Fulham	1	38	17.676315789	1.0815789474	1.3921052632	49.578947368
11	Leeds United	1	71	17.24084507	1.4	1.7295774648	55.577464789
12	Leicester City	1	70	17.891428571	1.4014285714	1.4328571429	52.828571429
13	Liverpool	1	38	16.494736842	1.9210526316	1.1868421053	62.210526316
14	Manchester City	1	71	16.315492958	2.085915493	0.7704225352	65.478873239
15	Manchester United	1	72	17.115277778	1.5444444444	1.2569444444	53.986111111
16	Newcastle United	1	72	17.4625	1.0625	1.4125	39.347222222
17	Norwich City	1	33	18.003030303	0.8757575758	1.9575757576	42.939393939
18	Sheffield United	1	38	16.726315789	0.8289473684	1.6421052632	41.842105263
19	Southampton	1	72	17.305555556	1.1958333333	1.4555555556	50.263888889
20	Tottenham Hotspur	1	71	17.163380282	1.523943662	1.2098591549	51.746478873
21	Watford	1	33	17.933333333	1.0515151515	1.7212121212	40.818181818
22	West Bromwich Albion	1	38	17.768421053	0.8894736842	1.7815789474	38.157894737
23	West Ham United	1	72	15.541666667	1.3736111111	1.3055555556	45.375
24	Wolverhampton Wanderers	1	71	17.75915493	0.9929577465	1.3183098592	49.661971831

Data Manipulation

Most Used Formations in Permier League

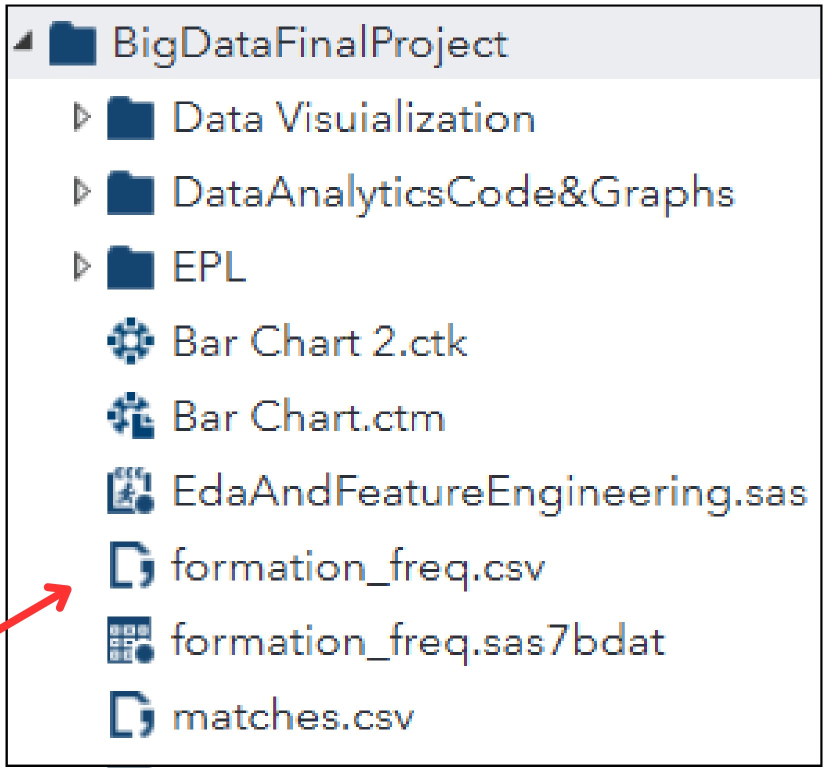
OUTPUT

formation	Frequency	Percent
3-4-1-2	50	3.6
3-4-3	209	15.05
3-5-1-1	10	0.72
3-5-2	138	9.94
4-1-4-1	78	5.62
4-2-2-2	6	0.43
4-2-3-1	344	24.77
4-3-2-1	4	0.29
4-3-3	246	17.71
4-4-1-1	46	3.31
4-4-2	206	14.83
4-5-1	16	1.15

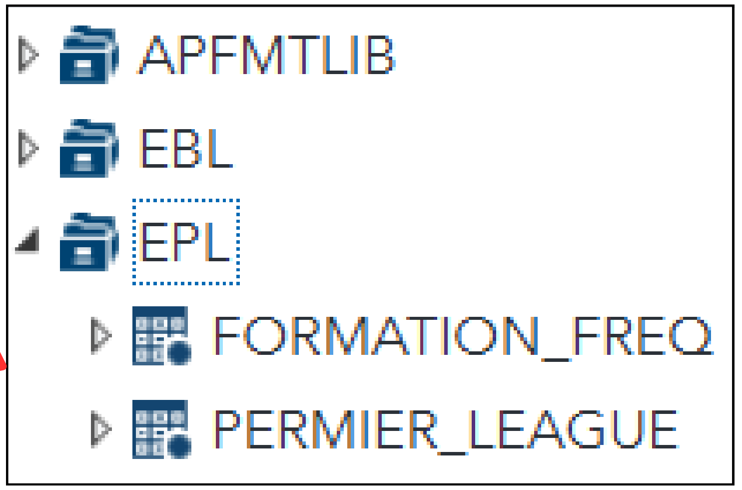
SAS CODE

```
/*Saving the most formations used in a csv file */
/*Defining the library */
libname EPL "/home/u63511609/BigDataFinalProject";

/*Capturing the PROC FREQ output in a dataset */
ods output OneWayFreqs=freq_output;
proc freq data=permier_league;
    tables formation / nocum;
run;
ods output close;

/*Exporting the dataset to a CSV file */
proc export data=freq_output
    outfile="/home/u63511609/BigDataFinalProject/formation_freq.csv"
    dbms=csv
    replace;
run;

/*Importing the CSV file into the EPL library */
proc import datafile="/home/u63511609/BigDataFinalProject/formation_freq.csv"
    out=EPL.formation_freq
    dbms=csv
    replace;
    guessingrows=max;
run;
```

OUTPUT



OUTPUT



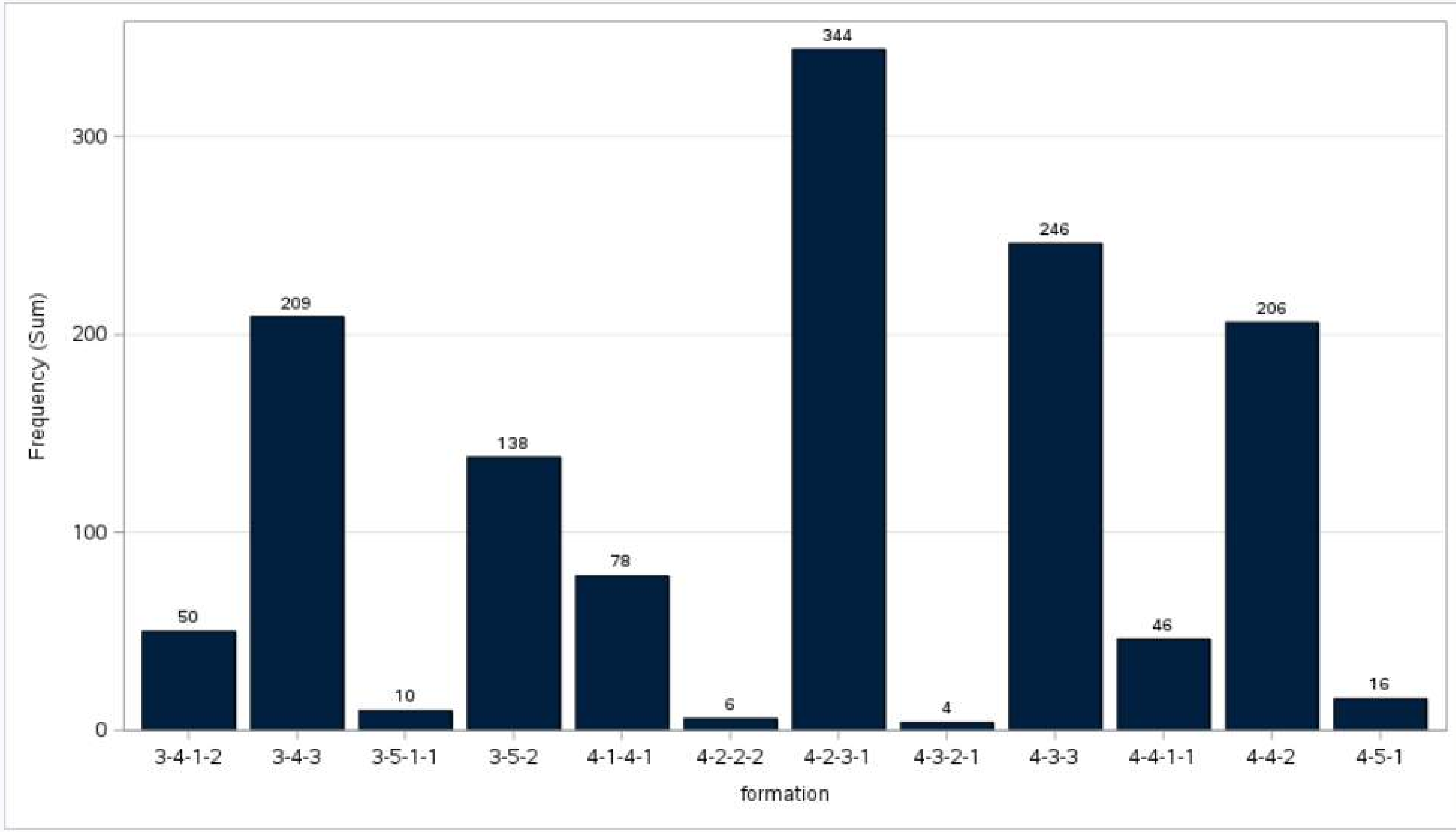
DATA VISUIALIZATION

- BigDataFinalProject
  - Data Visuialization
    - AvgShootingDistance.ctk
    - Ball_Contollling_Through_time.ctk
    - ConcededGoalsDisturbuation.ctk
    - EPL_Formation_Disturbution.ctk
    - FreeKickDisturbuation.ctk
    - GA_GF.ctk
    - gf_sot_rel.ctk
    - HomeAwayFK.sas
    - HomeAwayGF.sas
    - HomeAwayGF2.sas

- HomeAwayGF2.sas
- HomeAwayPkatt.sas
- HomeAwaySH.sas
- HomeAwaySOT.sas
- HomeAwayXGA_Mean.sas
- HomeAwayXG_Mean.sas
- HoneAwayGA.sas
- PenaltyDisturbution.ctk
- scoredGoalsByEachTeam.ctk
- ShoootingOnTargetDisturbuation.ctk
- TotalAttempsDisturbution.ctk

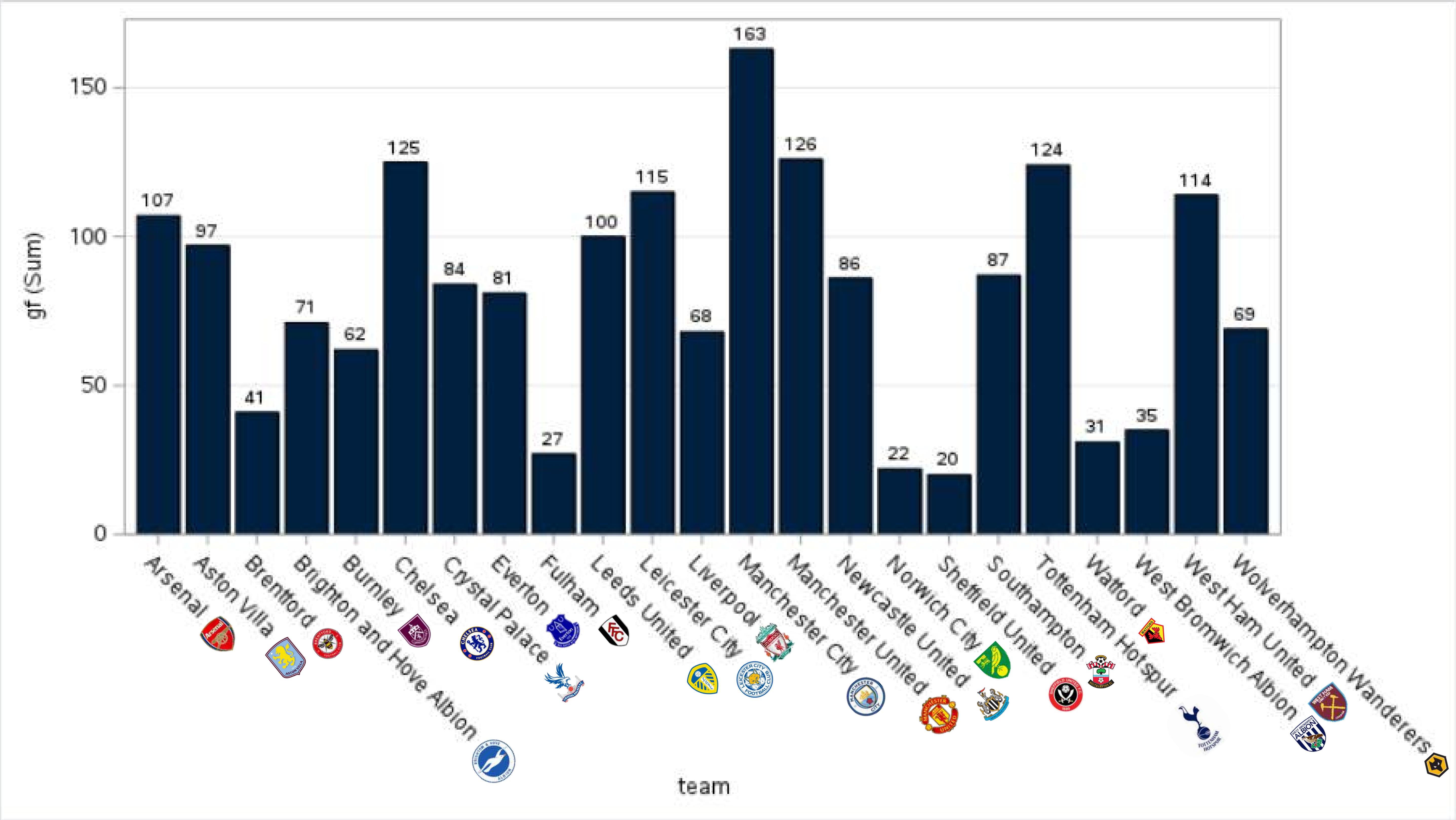
Most Used Formations in Permier League

From 2020-09-12 to 2022-04-25



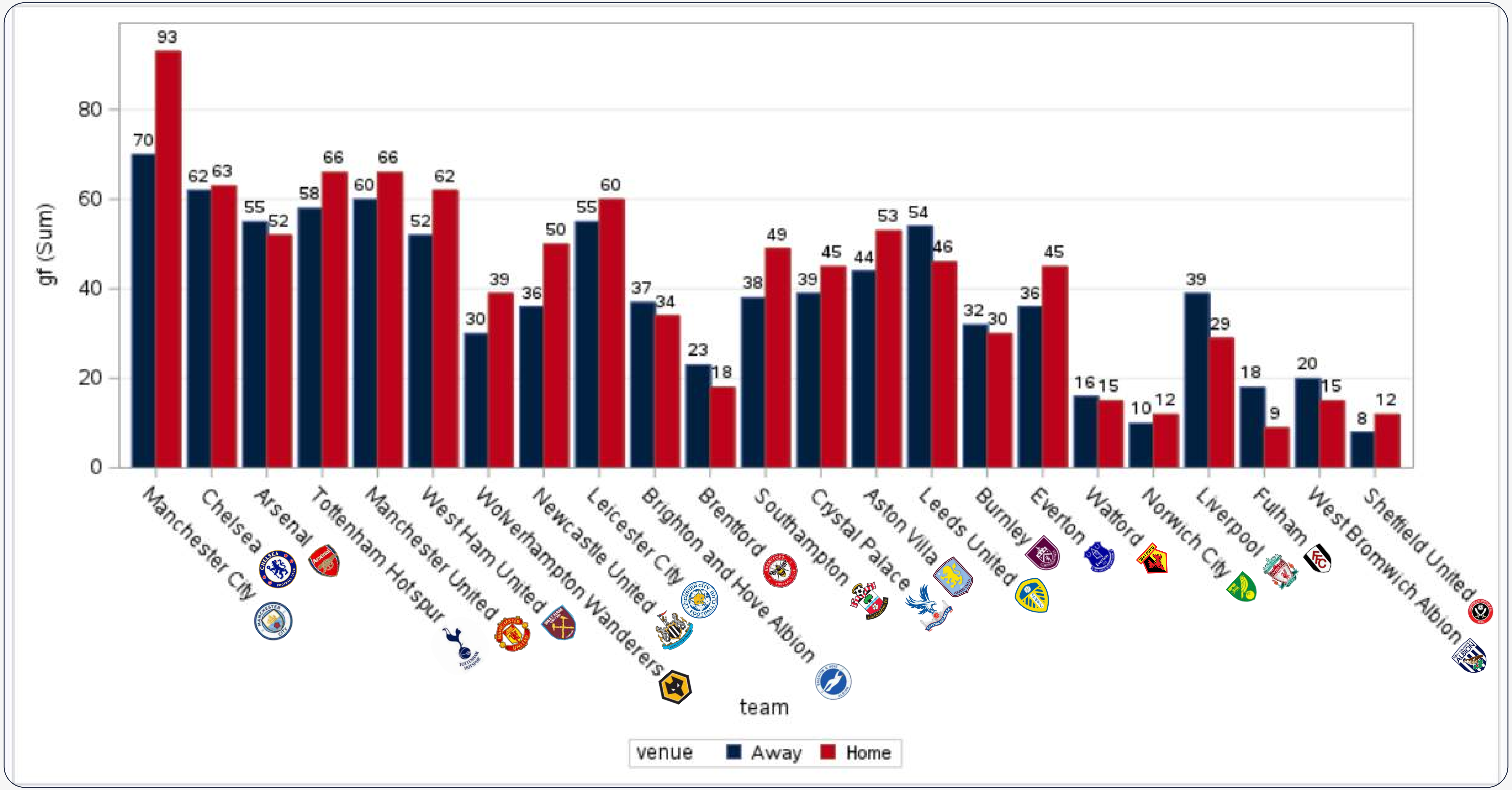
Total Number of Goals Scored by Each Team In EPL

From 2020-09-12 to 2022-04-25



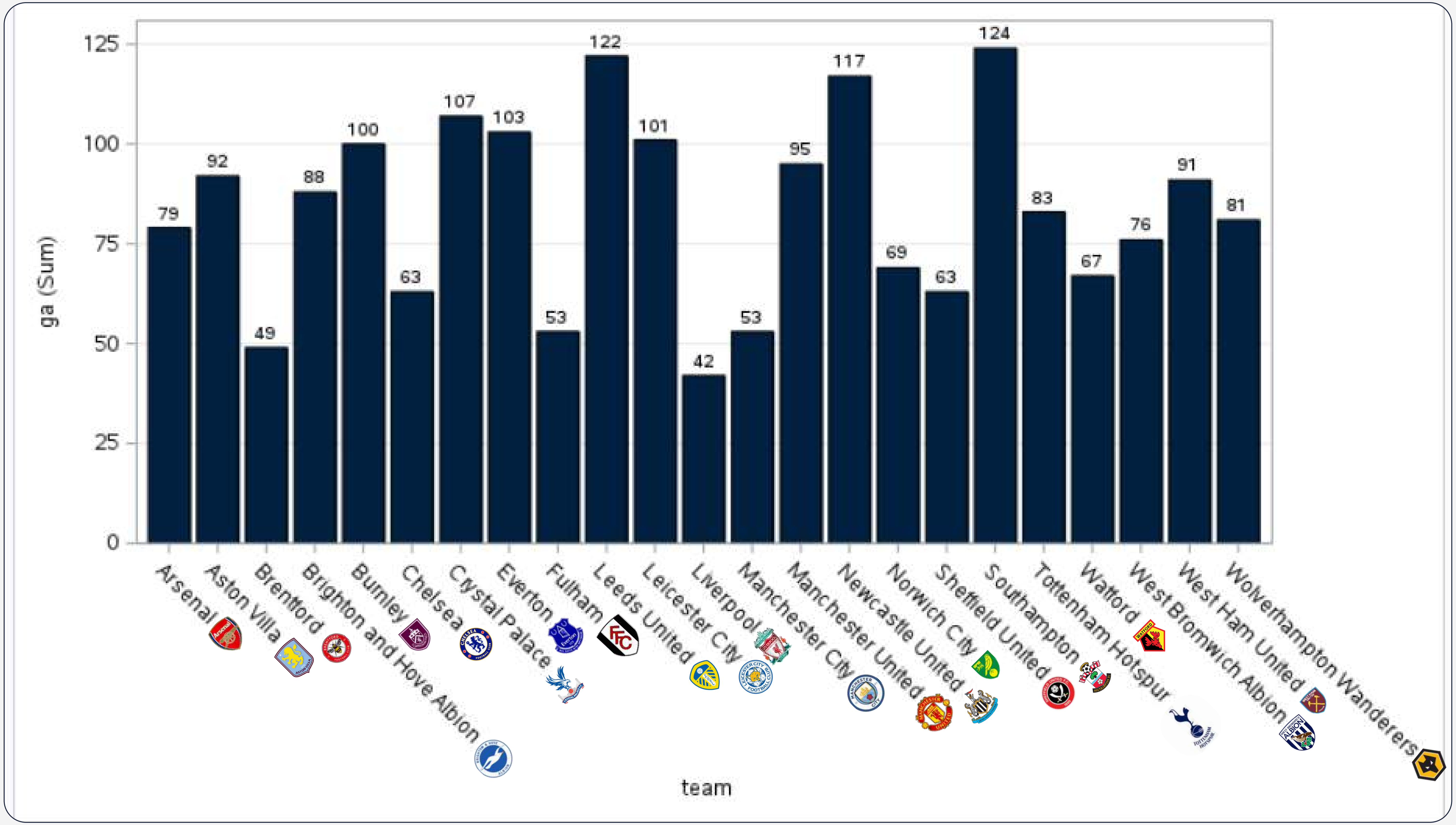
Total Number of Goals Scored by Each Team in EPL

From 2020-09-12 to 2022-04-25



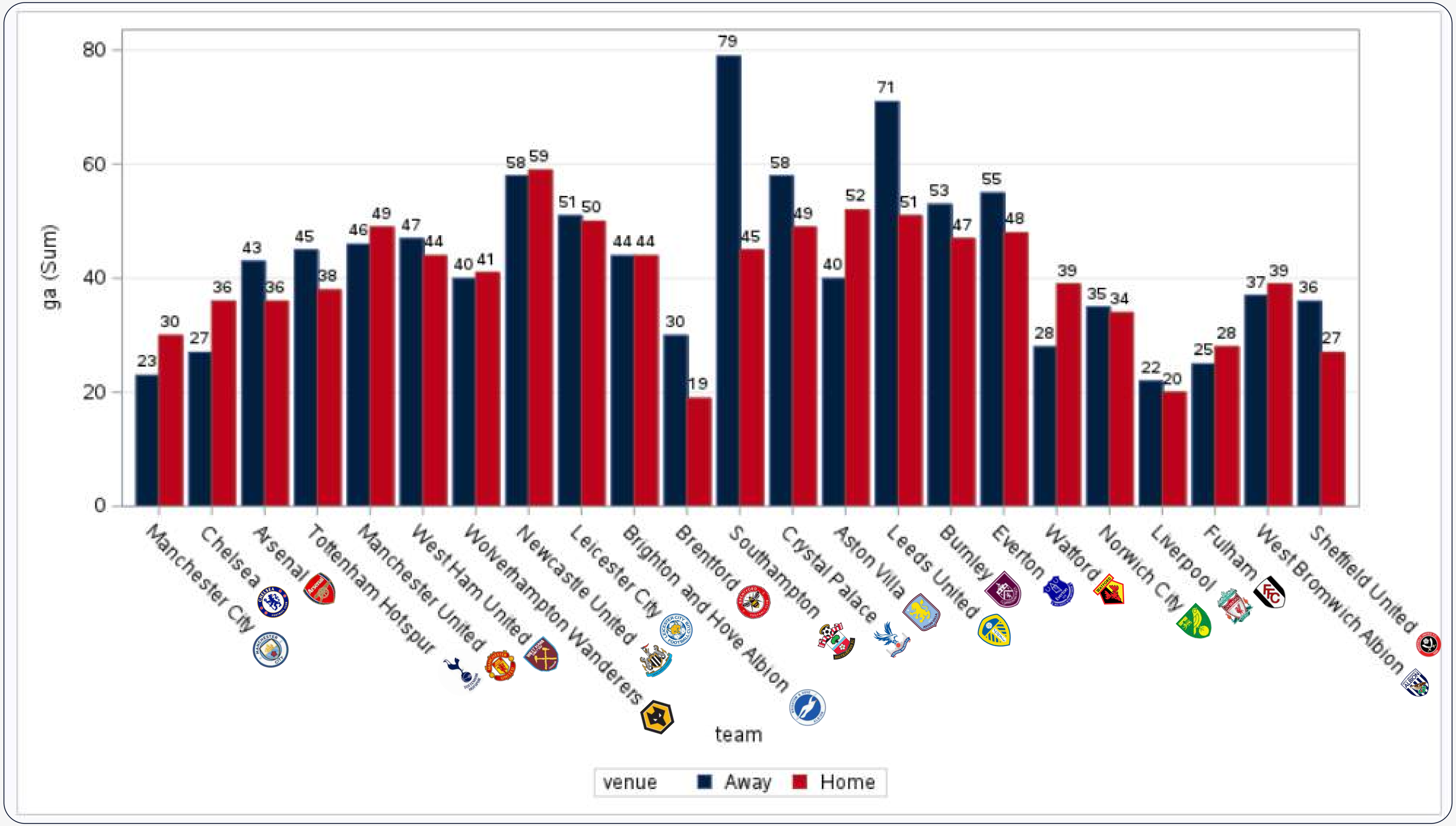
Total Number of Goals Conceded by Each team in EPL

From 2020-09-12 to 2022-04-25



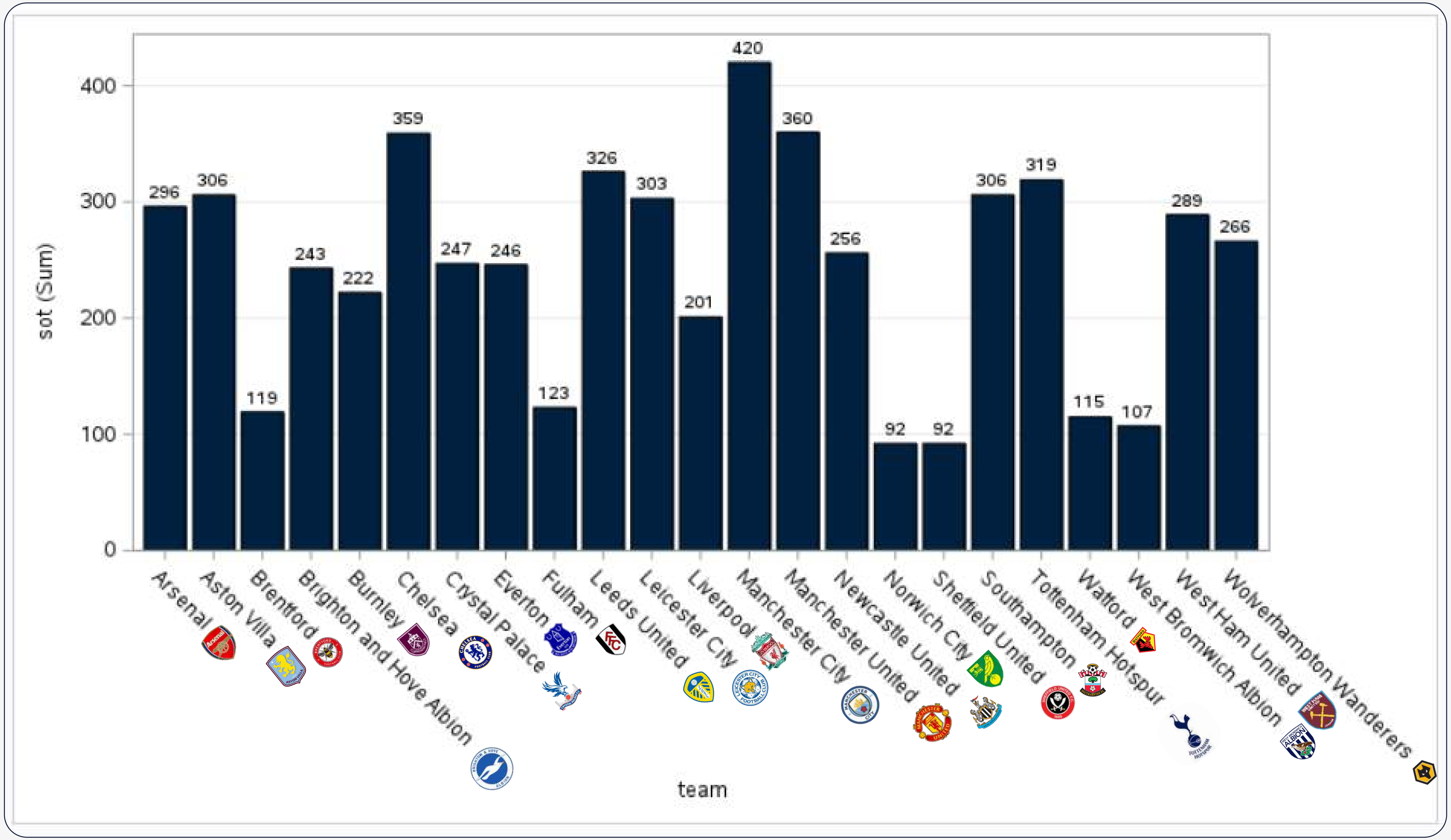
Total Number of Goals Conceded by Each Team in EPL

From 2020-09-12 to 2022-04-25



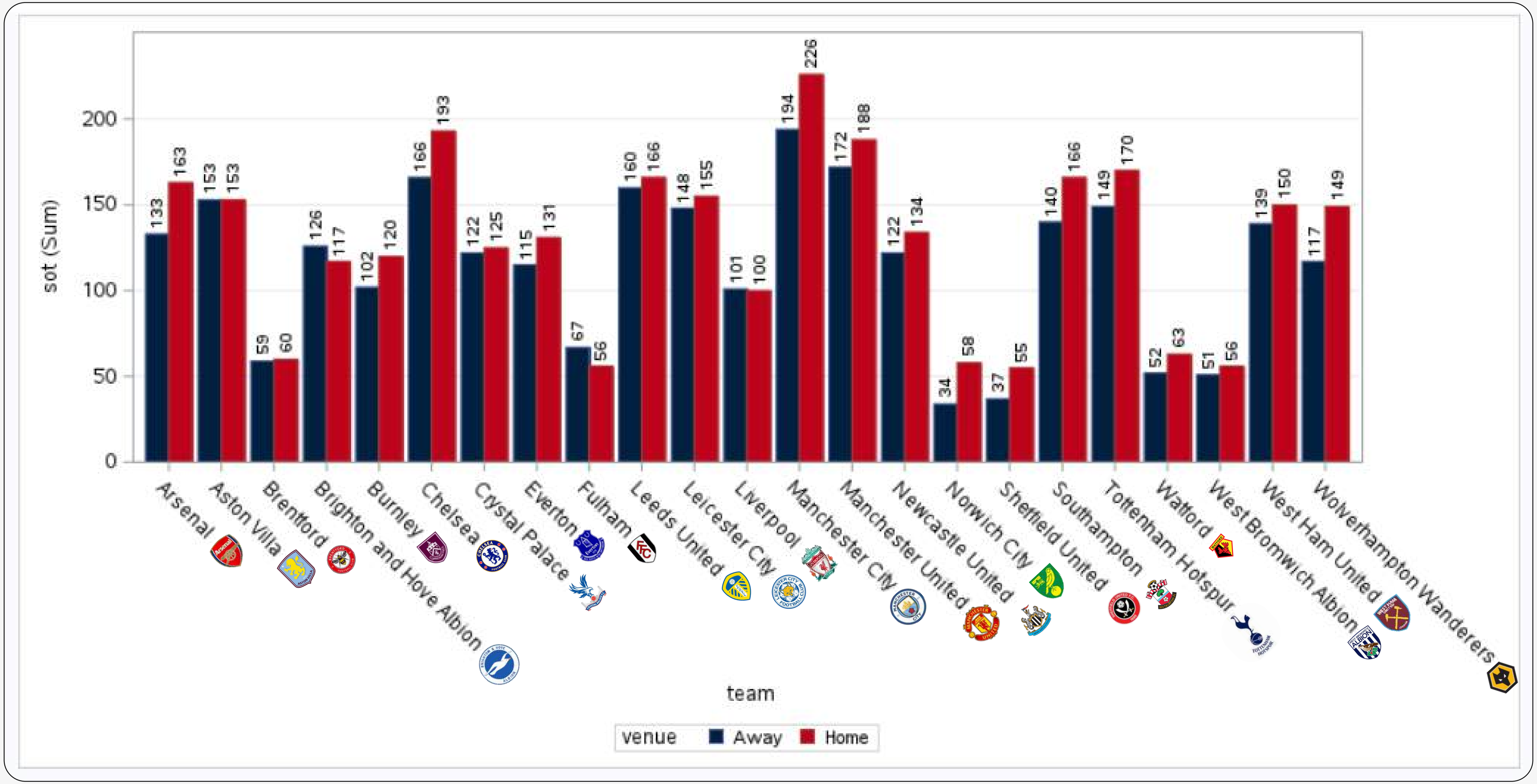
Number of Shots that were Directed Towards the Goal and would have Gone in if not for a Save or a Block in EPL

From 2020-09-12 to 2022-04-25



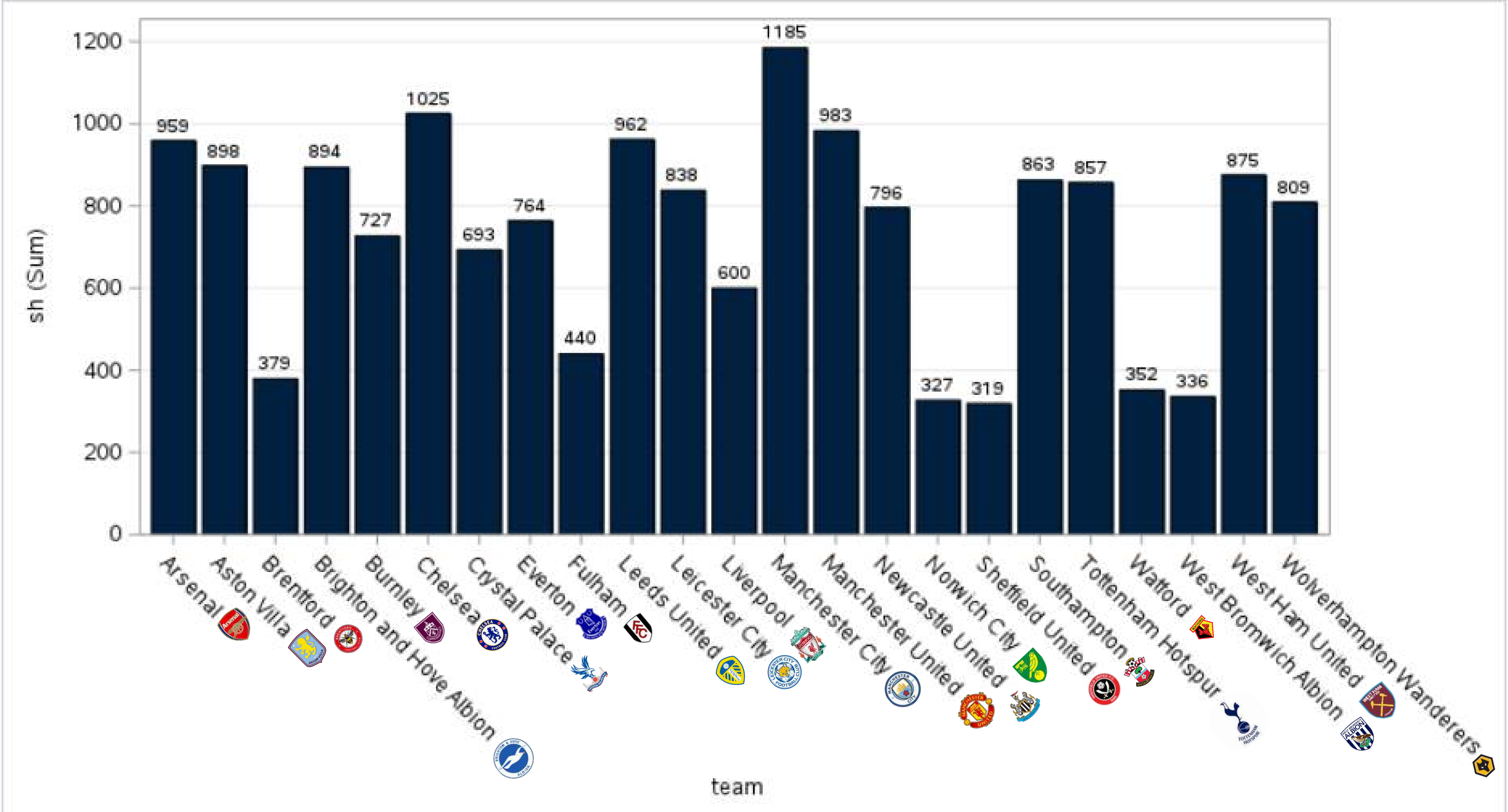
Number of Shots that were Directed Towards the Goal and would have gone in if not for a Save or a Block in EPL

From 2020-09-12 to 2022-04-25



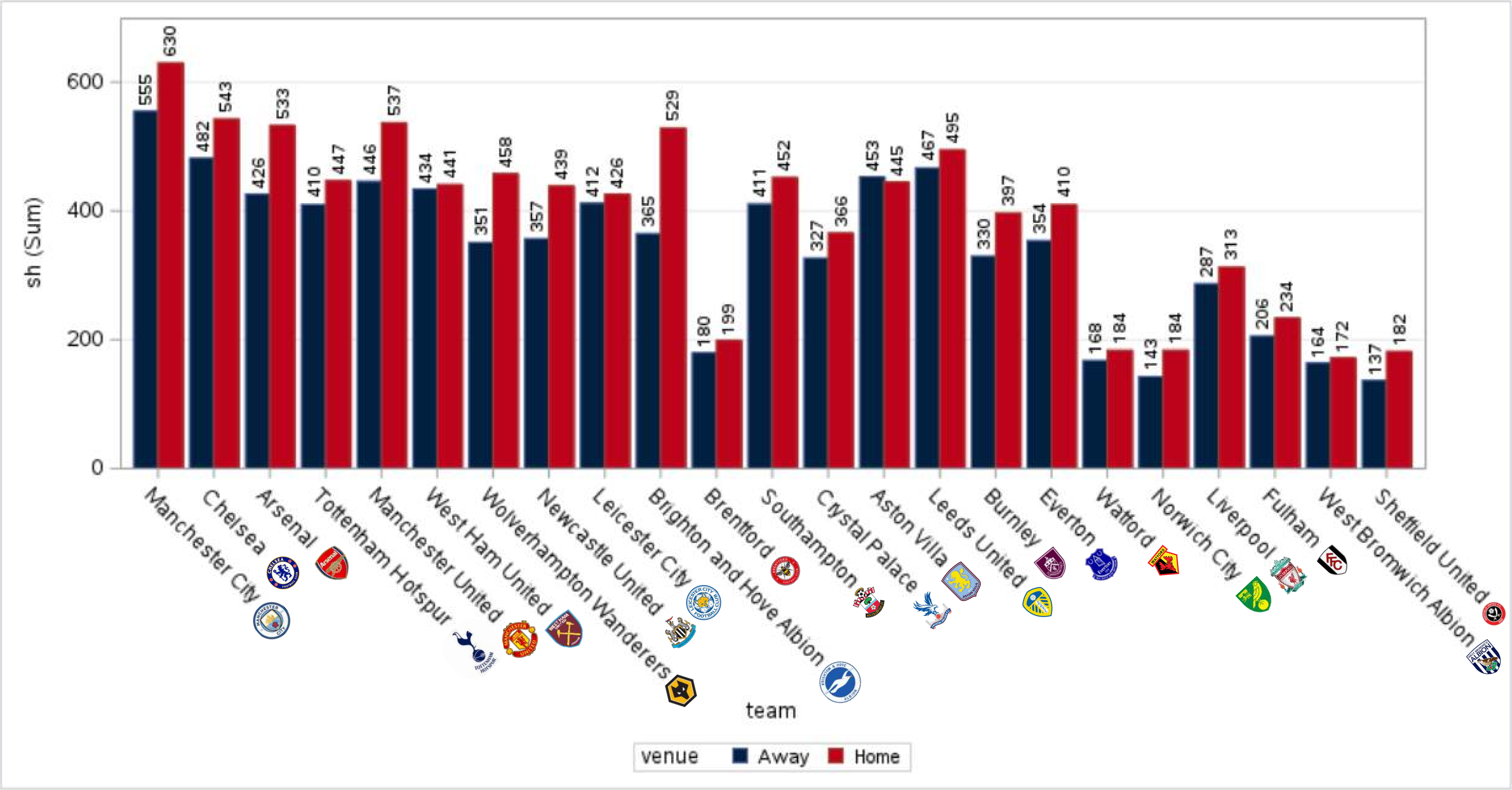
Total Number of Attempts made by Each Team IN EPL to Score a Goal.

From 2020-09-12 to 2022-04-25



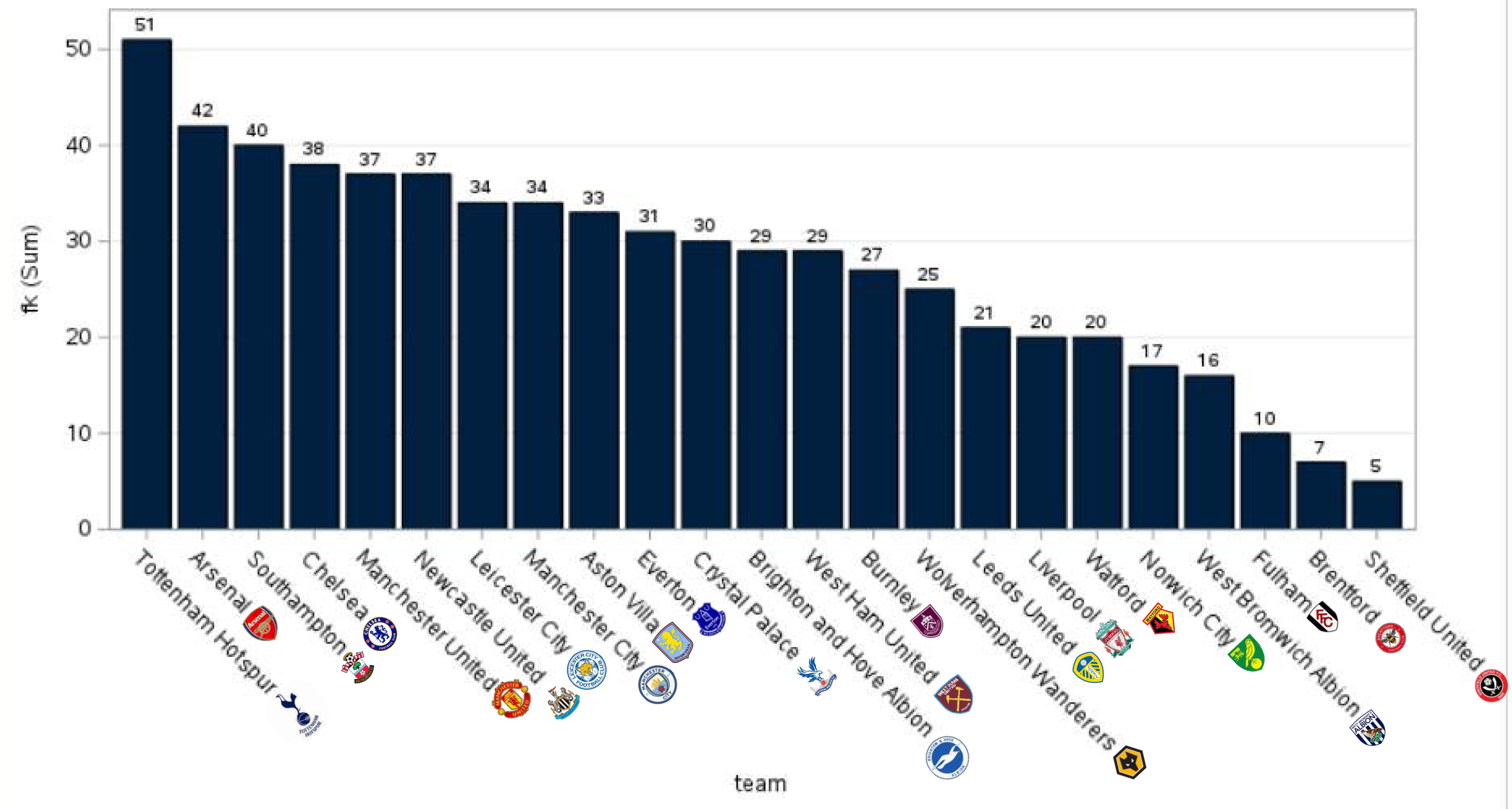
Total Number of Attempts Made by Each team IN EPL to Score a Goal.

From 2020-09-12 to 2022-04-25



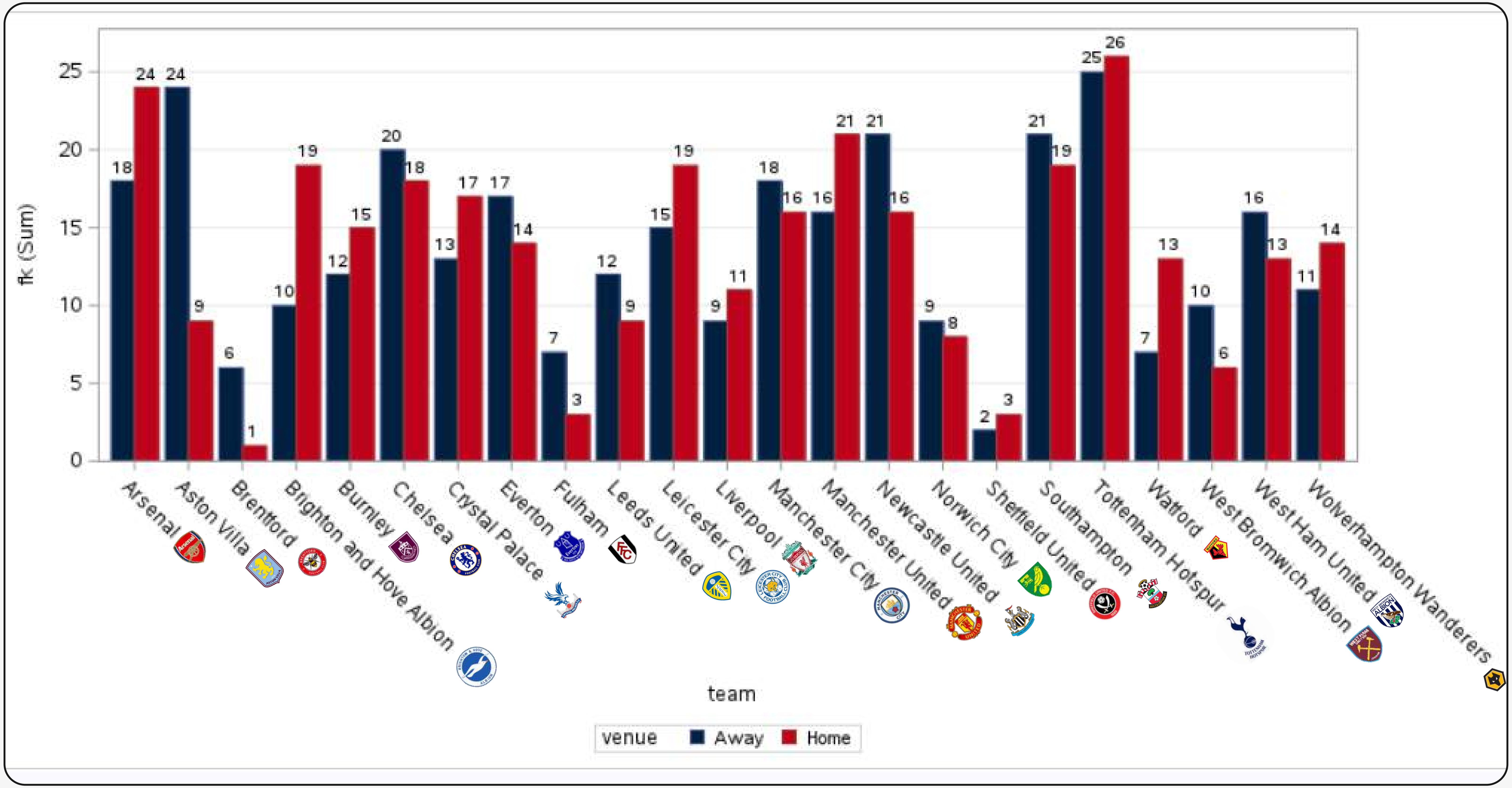
Number of Free kicks Awarded to Each team in EPL

From 2020-09-12 to 2022-04-25



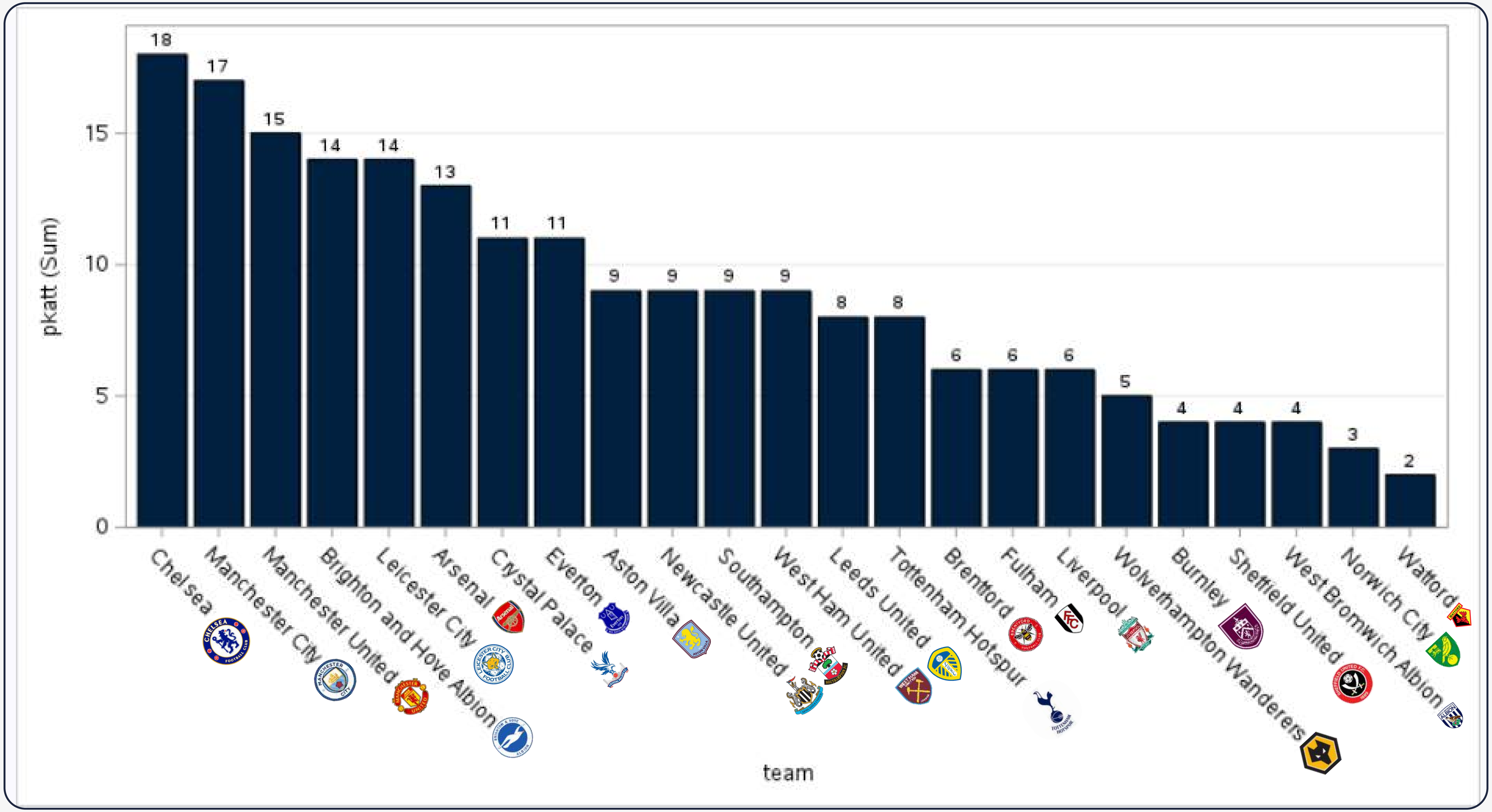
Number of Free kicks Awarded to Each team in EPL

From 2020-09-12 to 2022-04-25



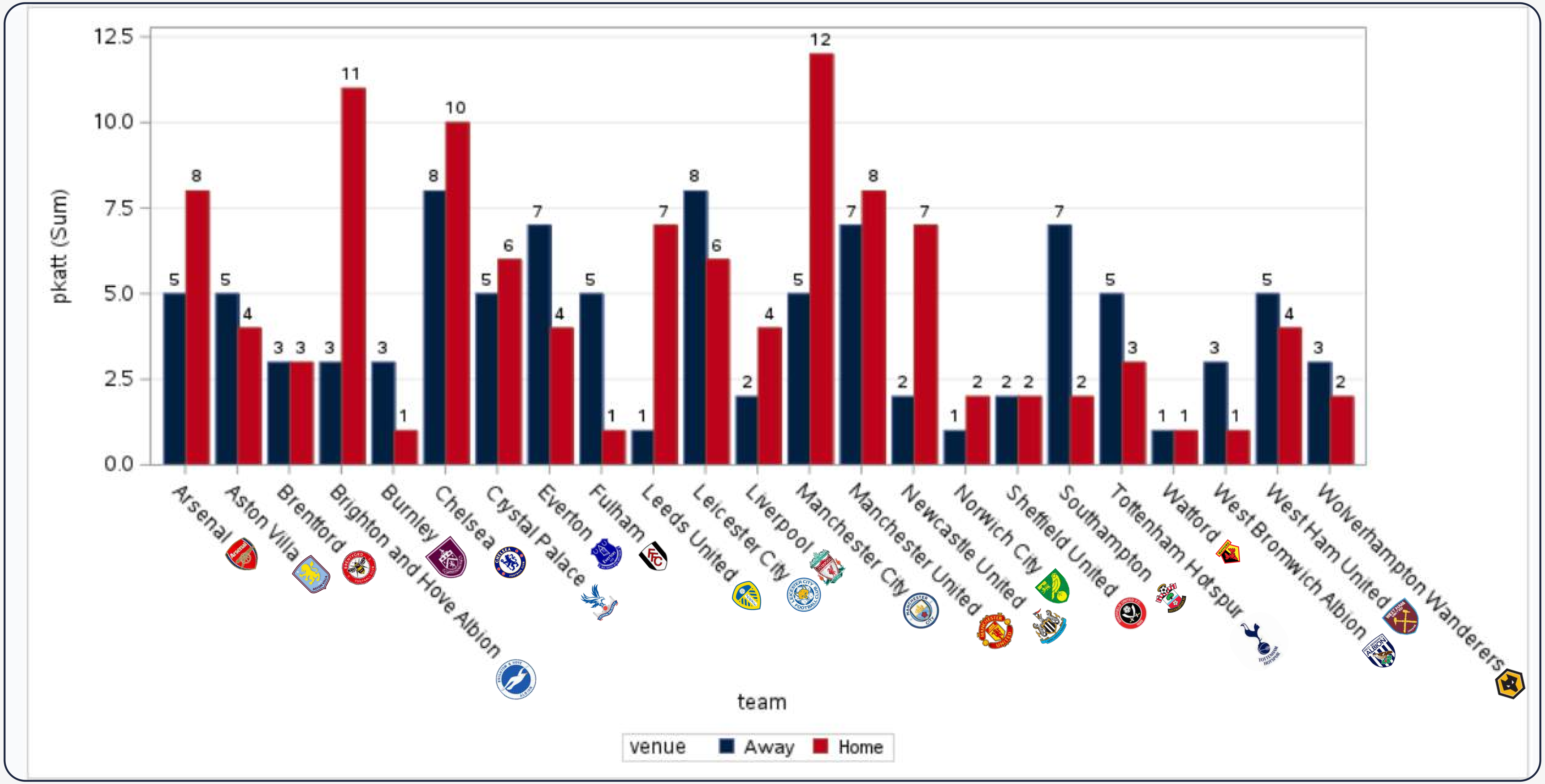
THE NUMBER OF PENALTY KICK ATTEMPTS MADE BY EACH TEAM IN EPL

From 2020-09-12 to 2022-04-25



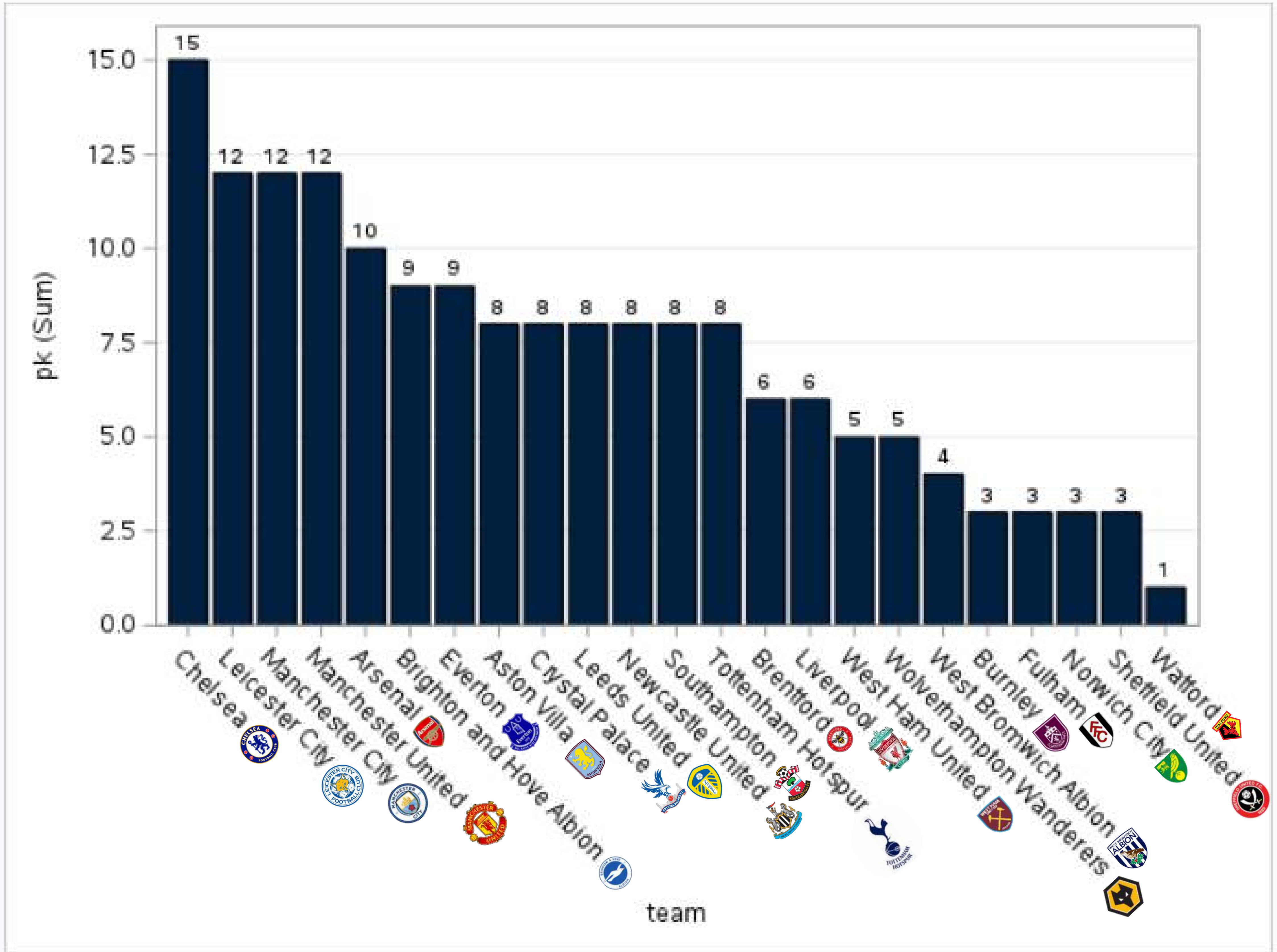
THE NUMBER OF PENALTY KICK ATTEMPTS MADE BY EACH TEAM IN EPL

From 2020-09-12 to 2022-04-25



THE NUMBER OF PENALTY KICKS SUCCESSFULLY CONVERTED INTO GOALS.

From 2020-09-12 to 2022-04-25



BEST 10 TEAMS IN PERCENTAGE OF SUCCESS PENALTY FRPM WHOLE PENALTY

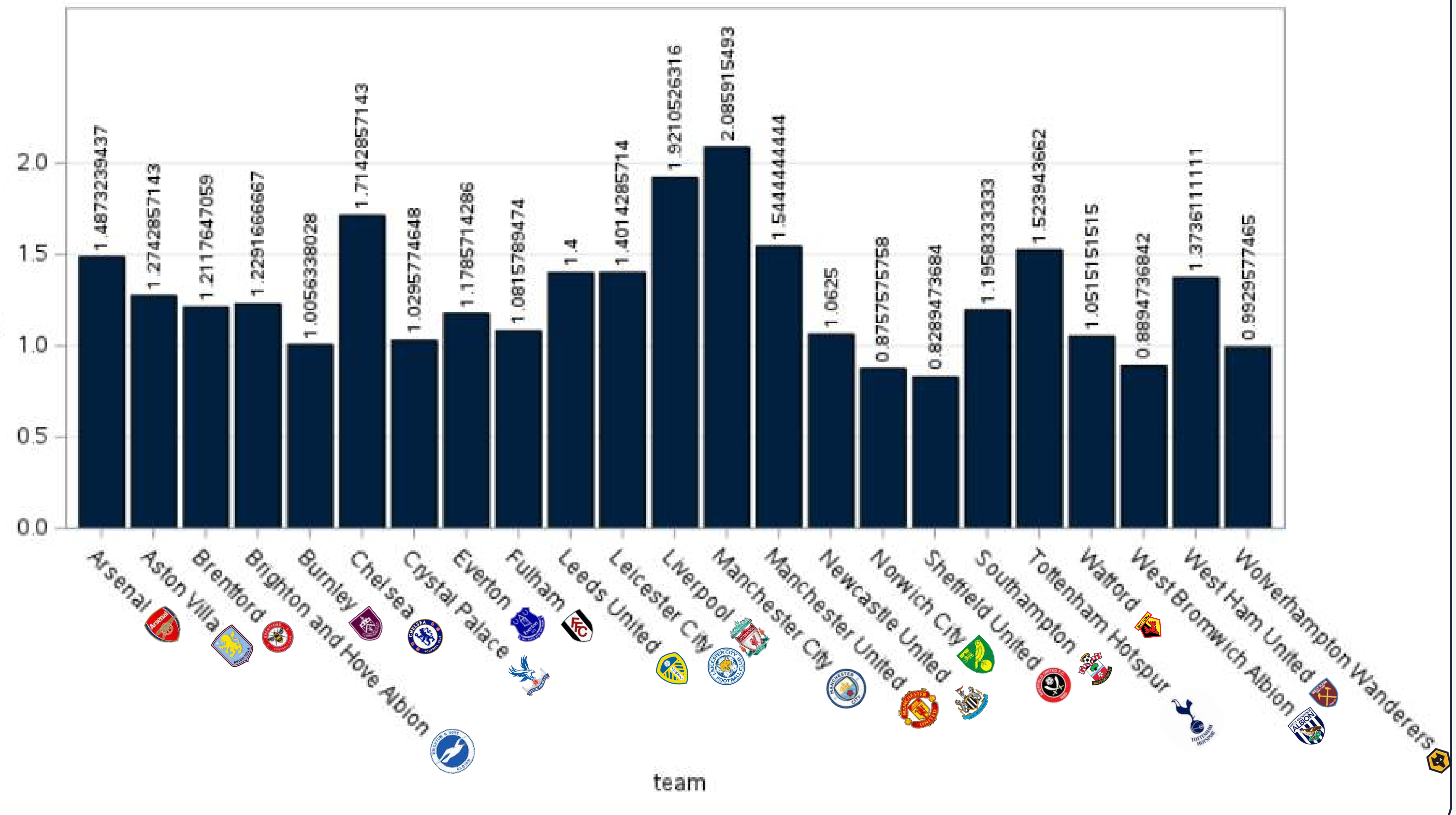
From 2020-09-12 to 2022-04-25

	TEAM	PK score	%
1		8/8	100 %
2		8/9	88.8 %
3		12/14	85.7 %
4		15/18	83.3 %
5		12/5	80%

	TEAM	PK score	%
6		9/11	81.8 %
7		4/5	80%
8		10/13	76.9 %
9		8/11	72.7 %
10		12/17	70%

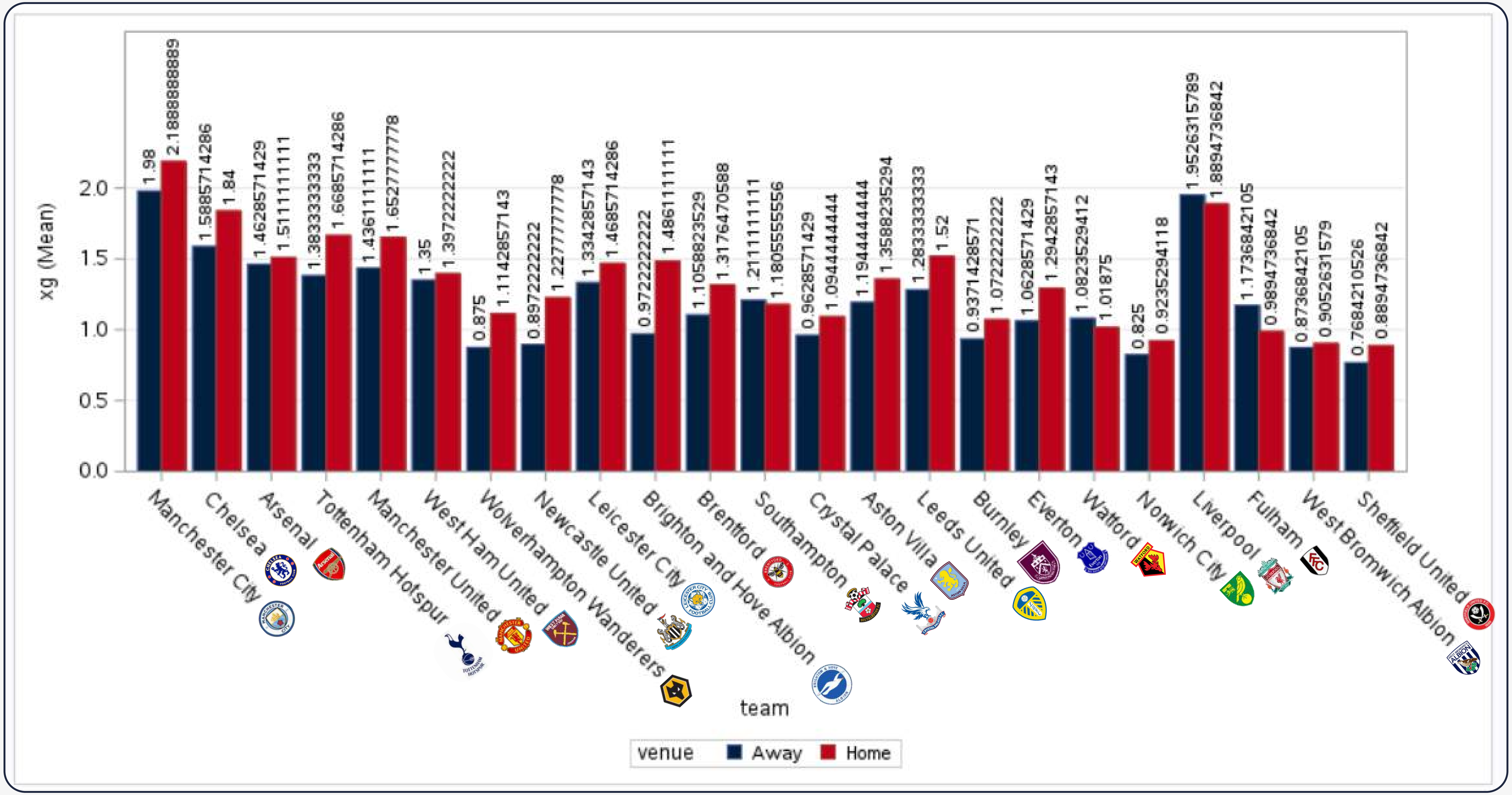
XG MEAN FOR EACH TEAM IN EPL

From 2020-09-12 to 2022-04-25



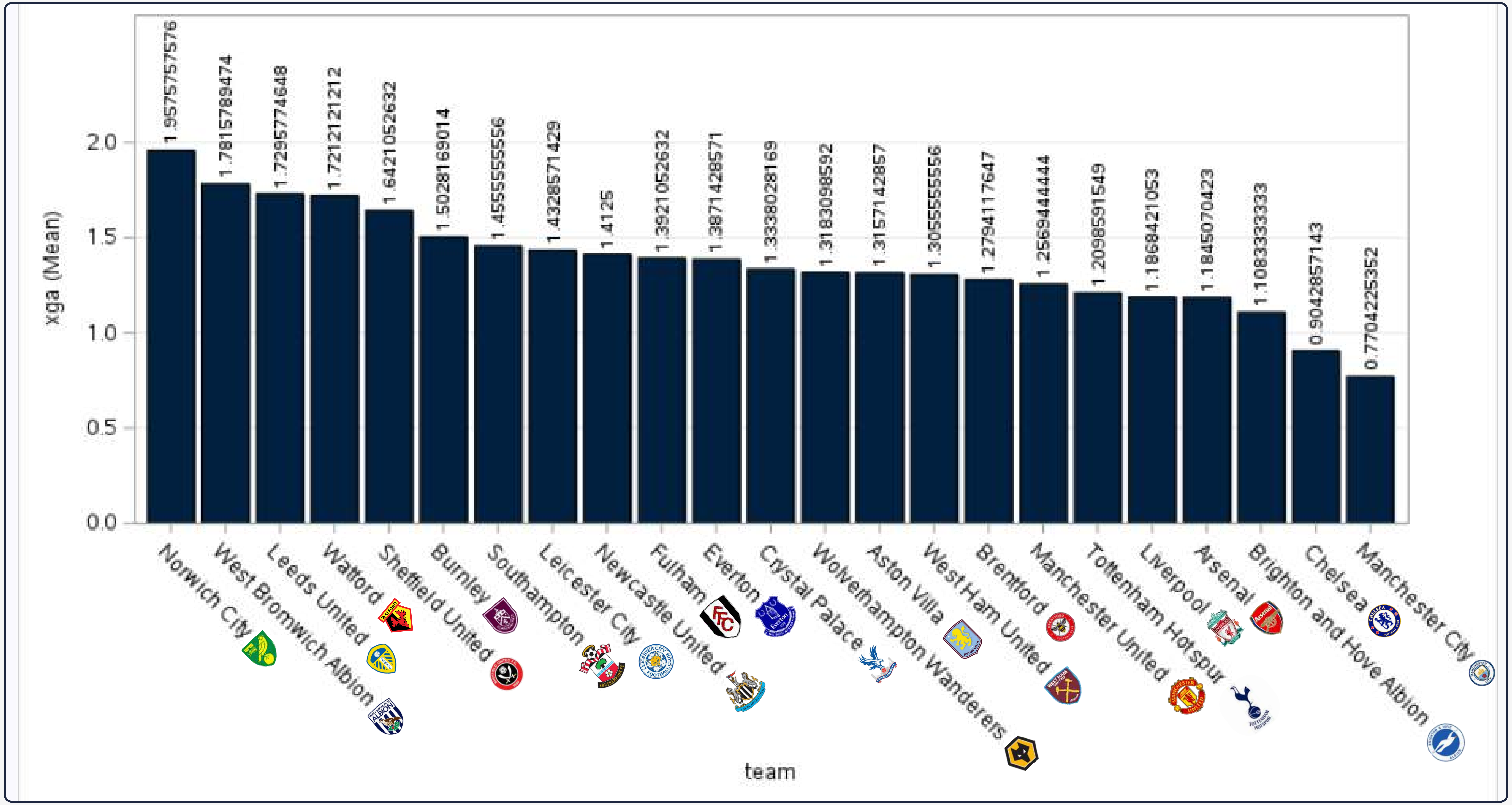
XG MEAN FOR EACH TEAM IN EPL

From 2020-09-12 to 2022-04-25



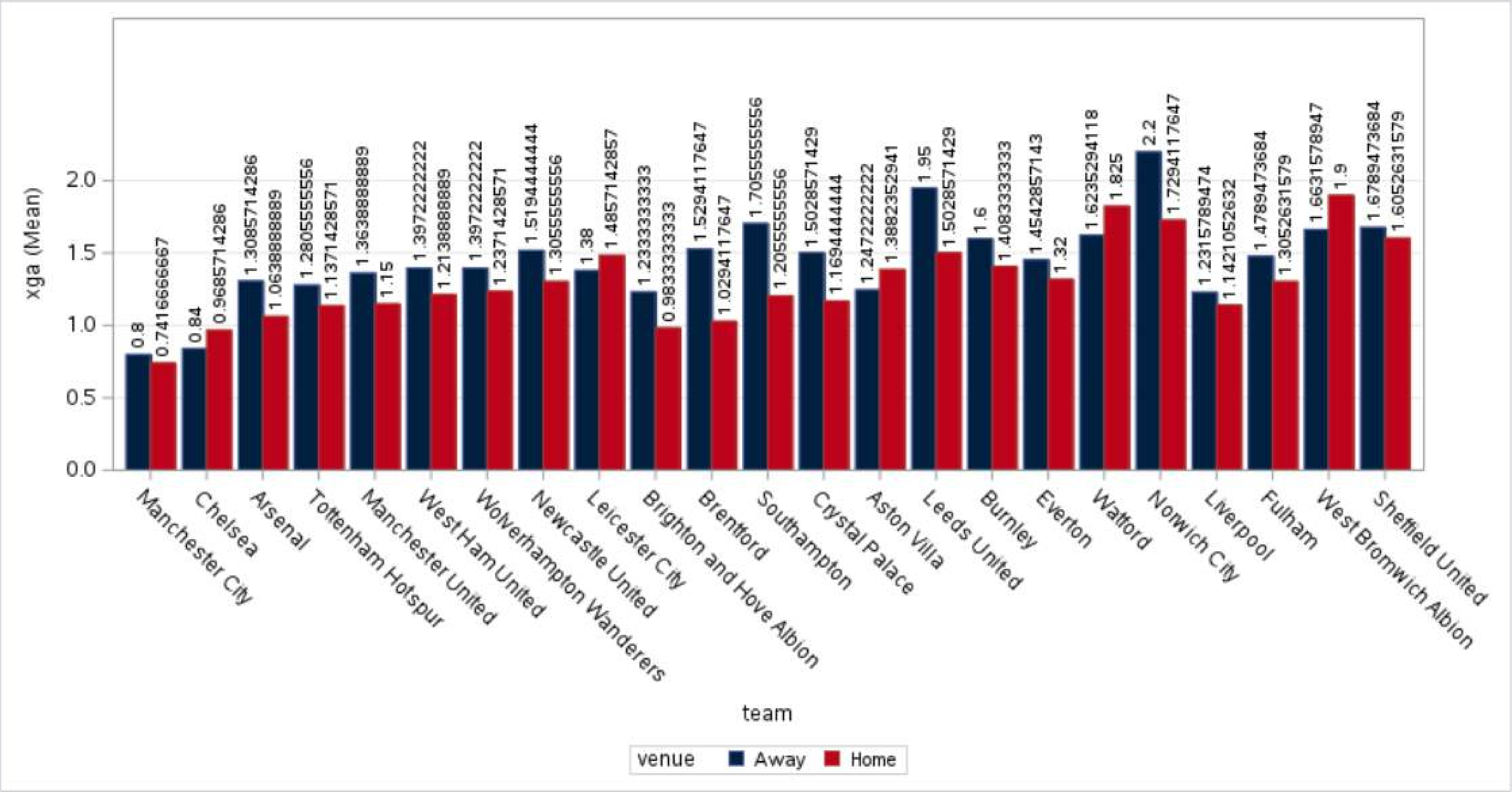
HIGHEST EXPECTED GOALS AGAINST FOR EACH TEAM IN EPL

From 2020-09-12 to 2022-04-25



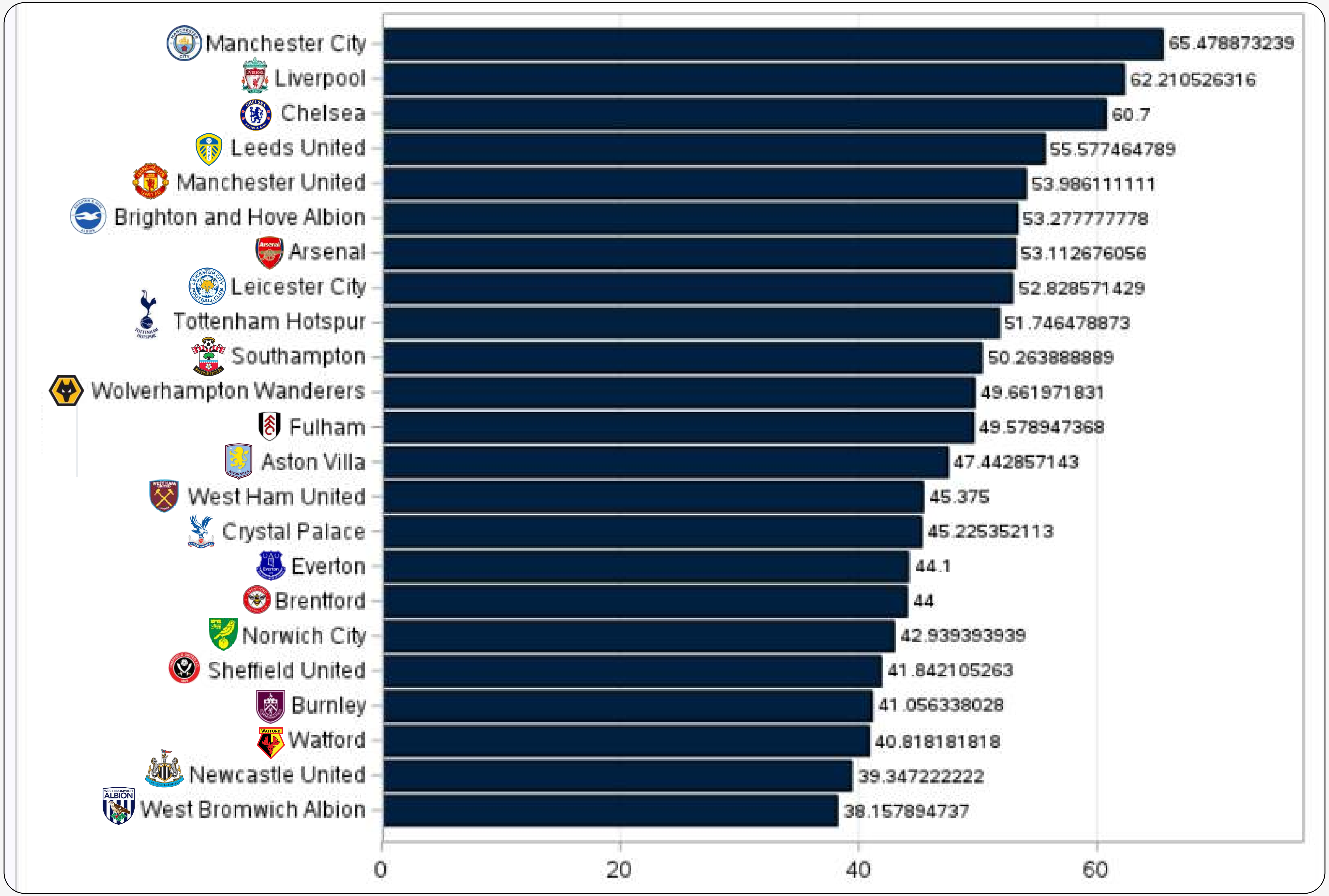
HIGHEST EXPECTED GOALS AGAINST FOR EACH TEAM IN EPL

From 2020-09-12 to 2022-04-25



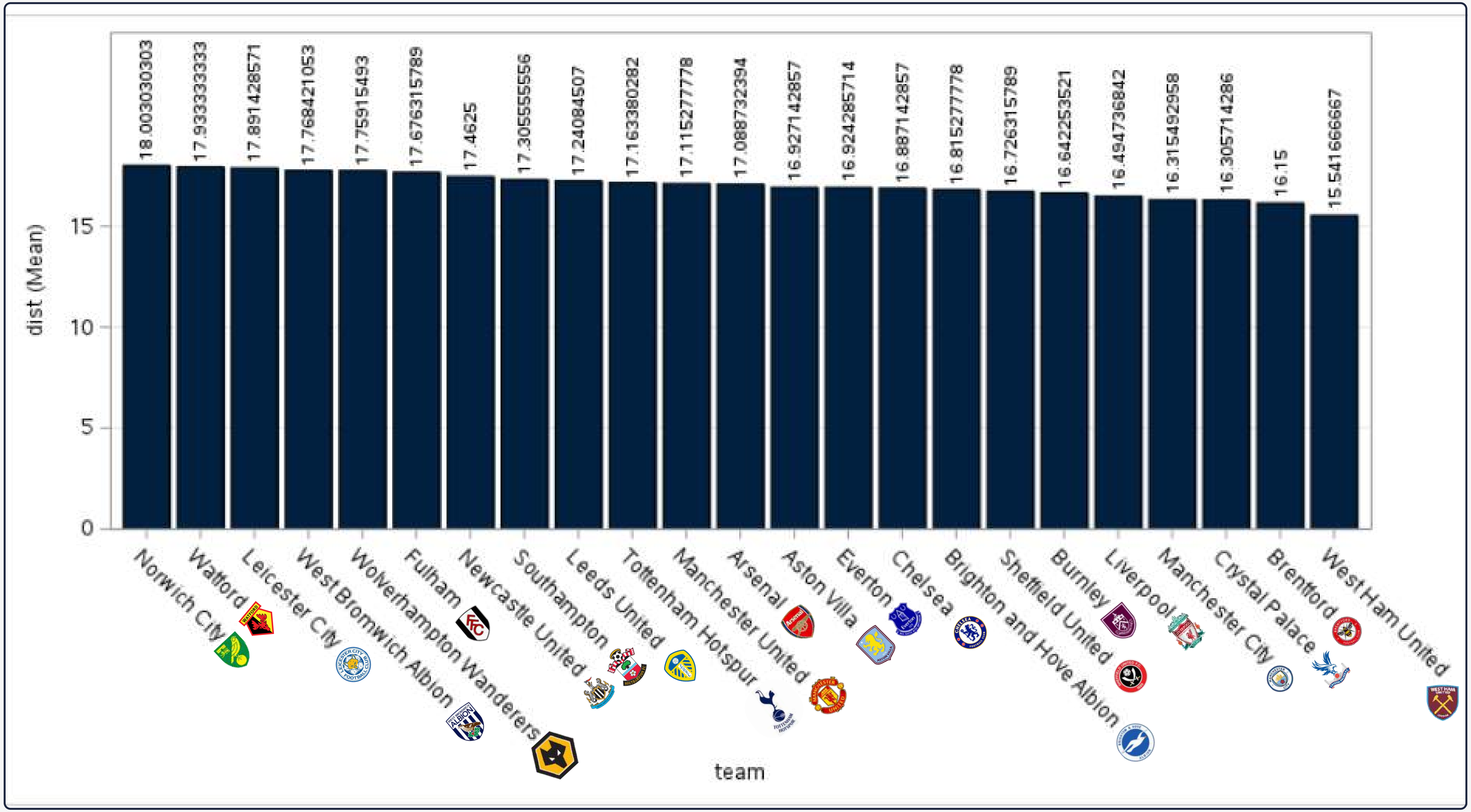
THE AVERAGE PERCENTAGE OF TIME THE TEAM CONTROLLED THE BALL DURING THE GAME.

From 2020-09-12 to 2022-04-25



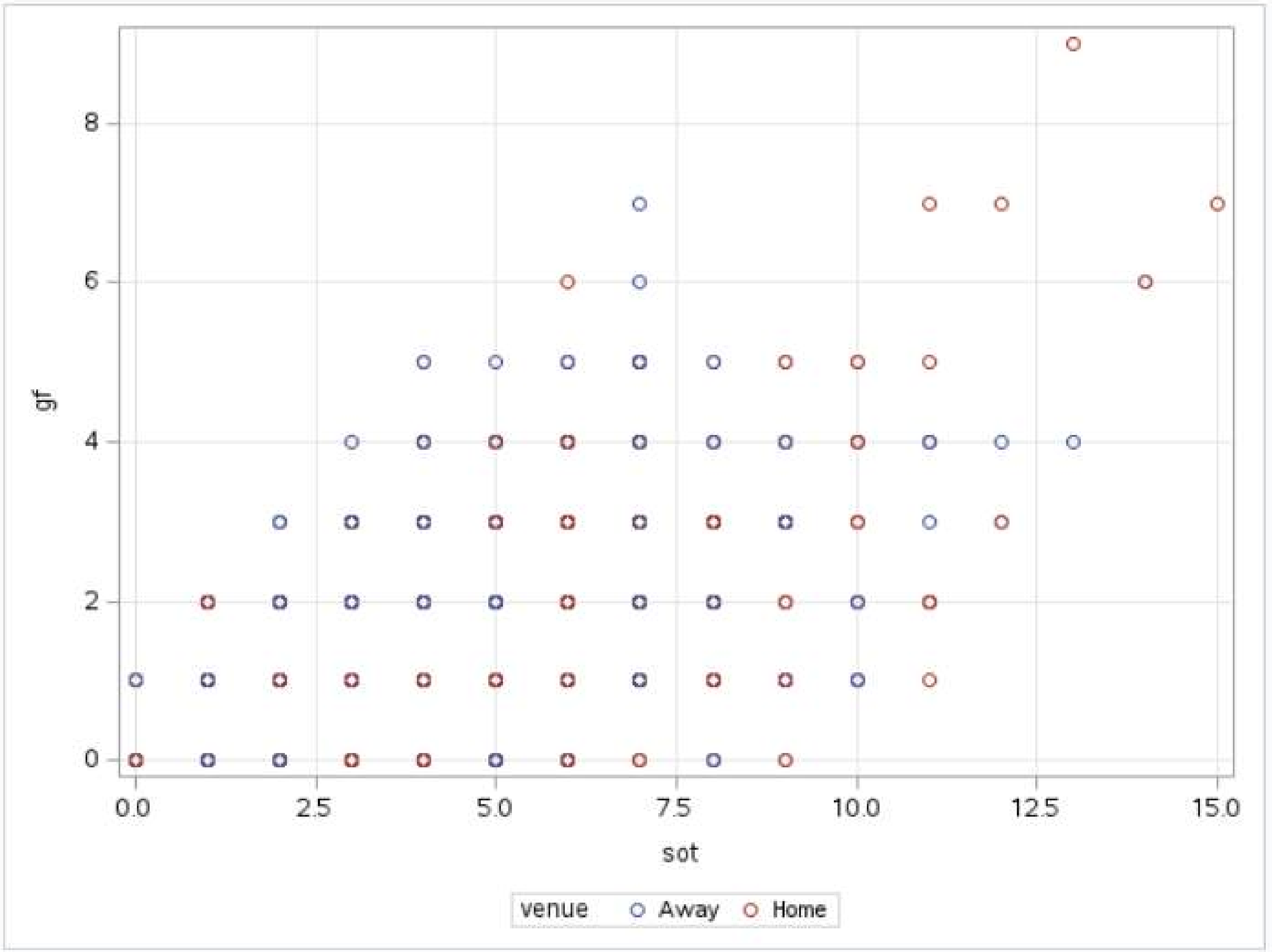
THE AVERAGE DISTANCE FROM WHICH SHOTS WERE TAKEN IN EPL

From 2020-09-12 to 2022-04-25

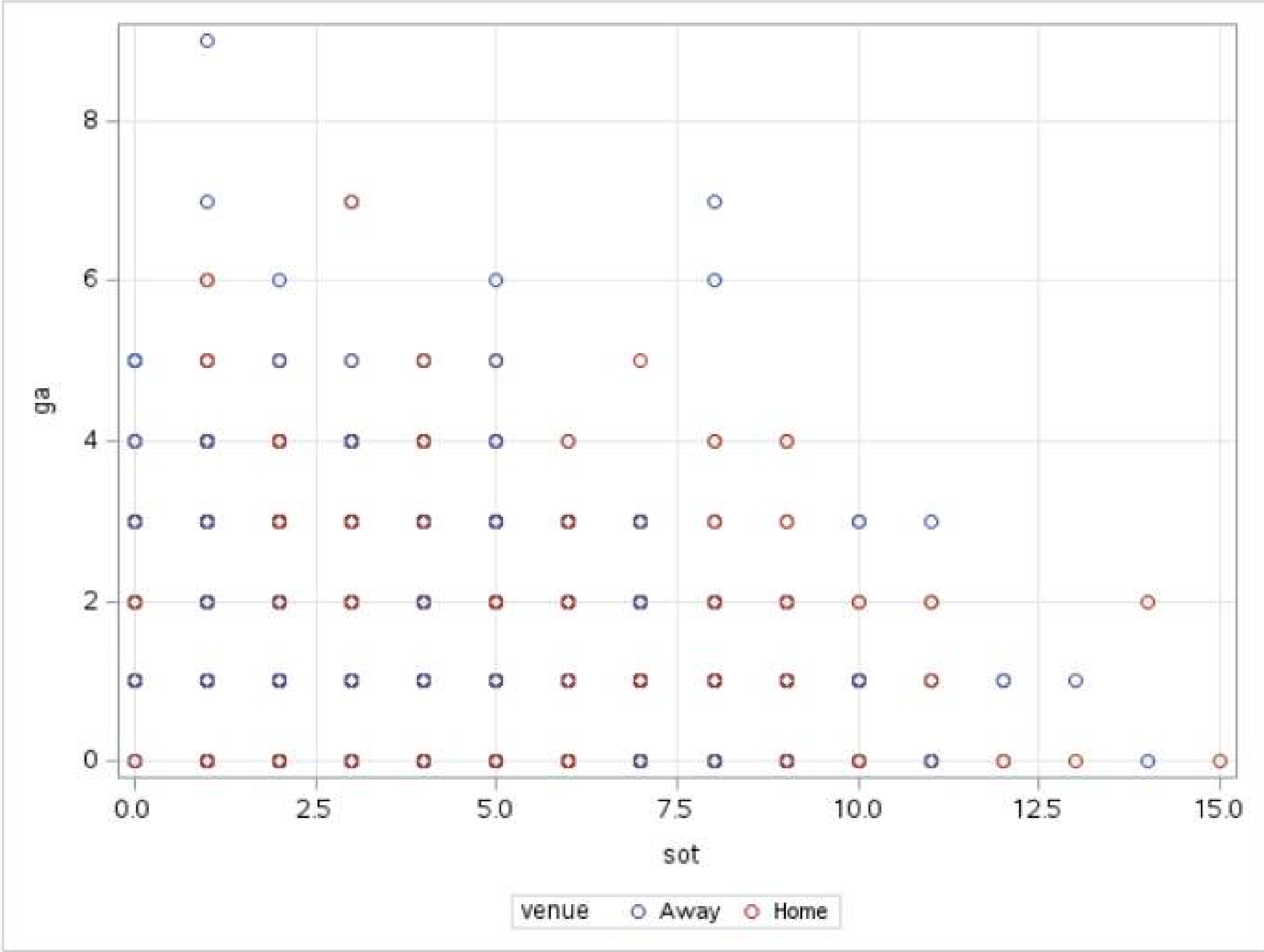


From 2020-09-12 to 2022-04-25

The Relationship between Scored Goals and Shooting On Target

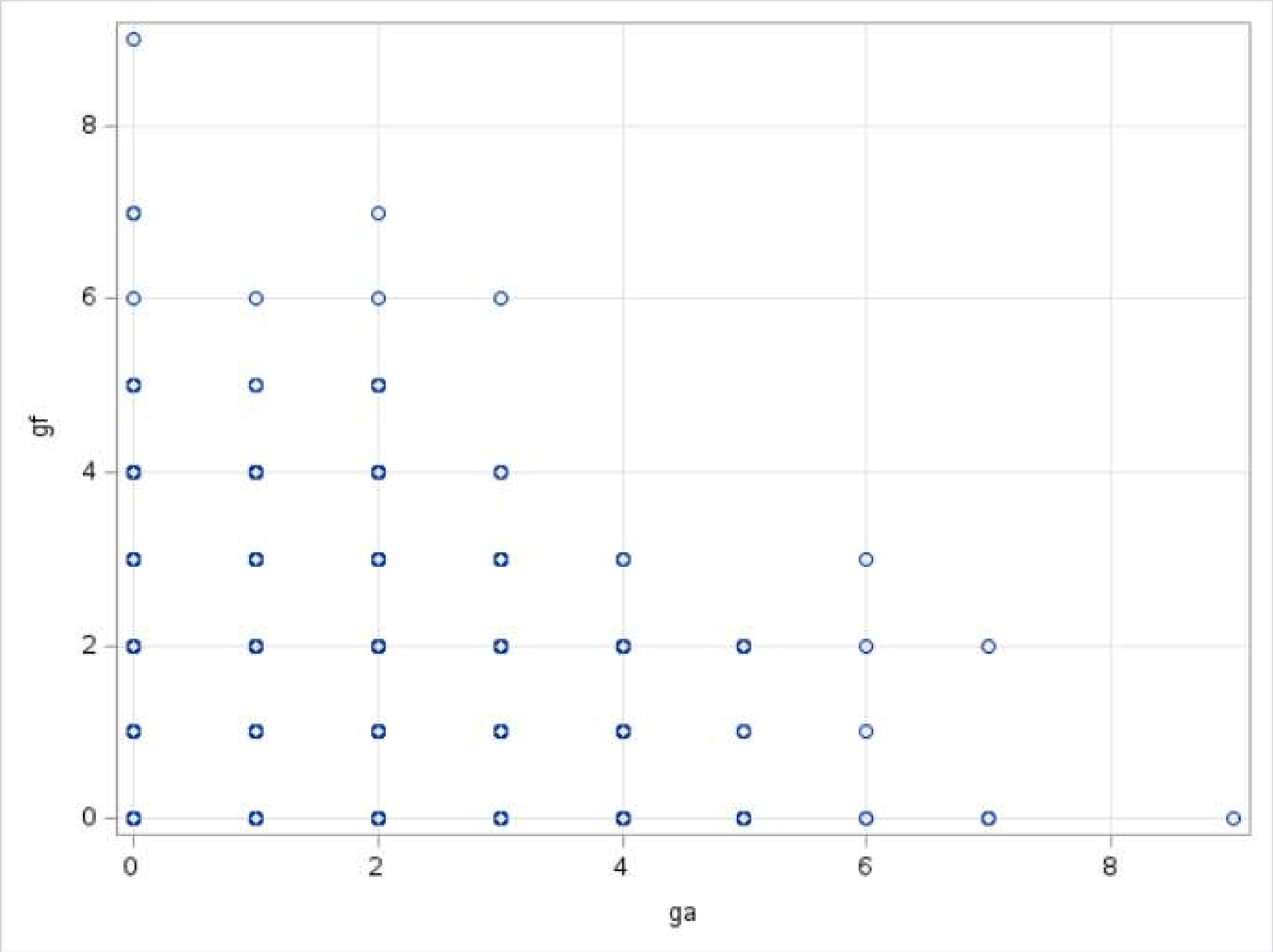


The Relationship between Conceded Goals and Shooting On Target

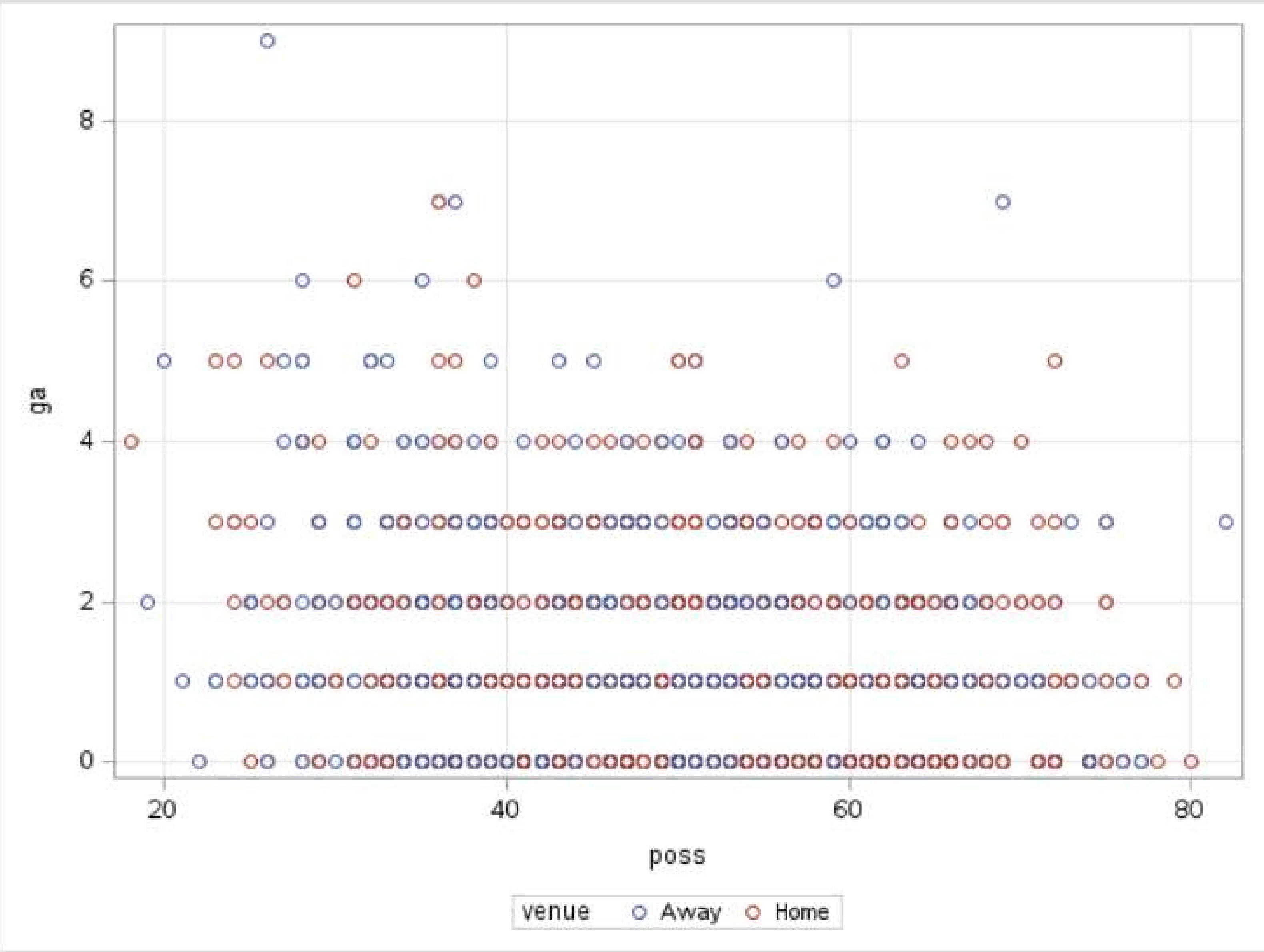


From 2020-09-12 to 2022-04-25

The Relationship between Scored Goals and CONCEDED GOALS



The Relationship between The average percentage of time the team controlled the ball during the game. and CONCEDED GOALS



BIG DATA COURSE FINAL PROJECT

STATISTICS

- BigDataFinalProject
  - Data Visuialization
  - DataAnalyticsCode&Graphs
  - EPL
  - Statistics
    - statisticsBySAS.sas
    - statisticsBySAS.sas~

From 2020-09-12 to 2022-04-25

SCORED GOALS

```
/*calc total scored goals by each team in home*/
proc sql;
    select sum(gf) as Total_Goals_Home
    from EPL.PERMIER_LEAGUE
    where venue = 'Home';
quit;
```

Total_Goals_Home
963

```
/*Mean Scored Goals At Home*/
proc sql;
    select mean(gf) as Mean_Goals_Home
    from EPL.PERMIER_LEAGUE
    where venue = 'Home';
quit;
```

Mean_Goals_Home
1.387608

```
/*calculating thetotal scored goals Away*/
proc sql;
    select sum(gf) as Total_Goals_Away
    from EPL.PERMIER_LEAGUE
    where venue = 'Away';
quit;
```

Total_Goals_Away
892

```
/*Mean Scored Goals At Away*/
proc sql;
    select mean(gf) as Mean_Goals_Away
    from EPL.PERMIER_LEAGUE
    where venue = 'Away';
quit;
```

Mean_Goals_Away
1.283453

CONCEDE GOALS

```
proc sql;
    select sum(ga) as Total_Goals_Against_Home
    from EPL.PERMIER_LEAGUE
    where venue = 'Home';
quit;
```

Total_Goals_Against_Home
925

```
proc sql;
    select sum(ga) as Total_Goals_Against_Away
    from EPL.PERMIER_LEAGUE
    where venue = 'Away';
quit;
```

Total_Goals_Against_Away
993

```
/*Mean Conceded Goal Home*/
proc sql;
    select mean(ga) as Mean_Goals_Against_Home
    from EPL.PERMIER_LEAGUE
    where venue = 'Home';
quit;
```

Mean_Goals_Against_Home
1.332853

```
/*Mean Conceded Goal Away*/
proc sql;
    select mean(ga) as Mean_Goals_Against_Away
    from EPL.PERMIER_LEAGUE
    where venue = 'Away';
quit;
```

Mean_Goals_Against_Away
1.428777

Statistics

From 2020-09-12 to 2022-04-25

SHOOTING ON TARGET

```
/*total shoots on target*/
proc sql;
    select sum(sot) as Total_Shots_On_Target_Home
    from EPL.PERMIER_LEAGUE
    where venue = 'Home';
quit;
```

Total_Shots_On_Target_Home
2954

```
/*total shoots on target Away*/
proc sql;
    select sum(sot) as Total_Shots_On_Target_Away
    from EPL.PERMIER_LEAGUE
    where venue = 'Away';
quit;
```

Total_Shots_On_Target_Away
2659

```
/*mean shoots on target Home*/
proc sql;
    select mean(sot) as Average_Shots_On_Target_Home
    from EPL.PERMIER_LEAGUE
    where venue = 'Home';
quit;
```

Average_Shots_On_Target_Home
4.256484

```
/*mean shoots on target Away*/
proc sql;
    select mean(sot) as Average_Shots_On_Target_Away
    from EPL.PERMIER_LEAGUE
    where venue = 'Away';
quit;
```

Average_Shots_On_Target_Away
3.825899

POSSESSION PERCENTAGE

```
/*controling at home*/
proc sql;
    select mean(poss) as Average_Possession_Home
    from EPL.PERMIER_LEAGUE
    where venue = 'Home';
quit;
```

Average_Possession_Home
50.78963

```
proc sql;
    select mean(poss) as Average_Possession_Away
    from EPL.PERMIER_LEAGUE
    where venue = 'Away';
quit;
```

Average_Possession_Away
48.61727

PKATT

```
/*Total pk HOME */
proc sql;
    select sum(pkatt) as Total_Pkatt_Home
    from EPL.PERMIER_LEAGUE
    where venue = 'Home';
quit;
```

Total_Pkatt_Home
109

```
/*Total pk Away */
proc sql;
    select sum(pkatt) as Total_Pkatt_Away
    from EPL.PERMIER_LEAGUE
    where venue = 'Away';
quit;
```

Total_Pkatt_Away
96

IS PLAYING AT HOME MAKE A DIFFERENCE ? vs

Based on results of statistics

- Teams In EPL scored 963 goal at their stadium while total scored 892 goals out thier stadium with a difference equal 71.
- Avg Scored Goals at home is equal to 1.38 while Average scored goals Away was 1.2
- Total conced goals in Home is equal to 925 while total Away conceded goals is equal to 993 with difference equal 68.
- Avg conceded goals in home is equal to 1.33 while avg concded away goals is equal to 1.43
- Total shots on target in home equal 2954 shot while total shots on target Away 2659 with a difference 295 shots
- Avg shots on target in Home equal 4.2 shot while the avg shots on target Away equal 3.8

BIG DATA COURSE FINAL PROJECT

REFREE STATISTICS

- refree
  - referee_performance_summary.csv
  - referee_performance_summary_with_percentages.csv
  - refree_analysis.sas
  - refree_analysis.sas~

SAS CODE

```
/* Creating a table with referee statistics */
proc sql;
    create table referee_stats as
    select
        referee,
        count(*) as Total_Matches, /* Total matches for each referee */
        count(distinct team) as Unique_Teams, /* Count of unique teams per referee */
        sum(pkatt) as Total_PK, /* Sum of penalty kicks (pkatt) */
        sum(fk) as Total_Free_Kicks /* Sum of free kicks (fk) */
    from matches
    group by referee
    order by referee;
quit;


/* Printing the resulting table to verify */
proc print data=referee_stats;
    title "Referee Statistics";
run;
```

Statistics

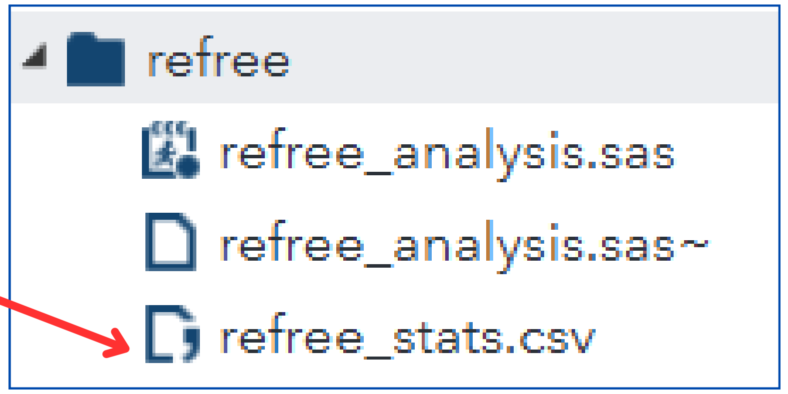
From 2020-09-12 to 2022-04-25

This table show total matches supervised for each referee and how wany different he managed their matches and how he perform in English Premier League

Obs	referee	Total_Matches	Unique_Teams	Total_PK	Total_Free_Kicks
1	Andre Marriner	81	21	13	27
2	Andy Madley	61	20	9	29
3	Anthony Taylor	100	22	22	52
4	Chris Kavanagh	70	21	11	35
5	Craig Pawson	90	21	10	53
6	Darren England	50	18	10	32
7	David Coote	80	22	12	36
8	Graham Scott	48	21	6	20
9	Jarred Gillett	14	11	1	7
10	John Brooks	6	5	1	1
11	Jonathan Moss	91	20	10	43
12	Kevin Friend	79	20	12	30
13	Lee Mason	22	16	3	11
14	Martin Atkinson	96	22	9	29
15	Michael Oliver	99	20	22	45
16	Michael Salisbu	4	4	1	4
17	Mike Dean	87	23	13	41
18	Paul Tierney	85	22	12	37
19	Peter Bankes	54	19	4	29
20	Robert Jones	40	16	7	16
21	Simon Hooper	50	18	6	25
22	Stuart Attwell	78	23	11	28
23	Tony Harrington	4	4	0	3

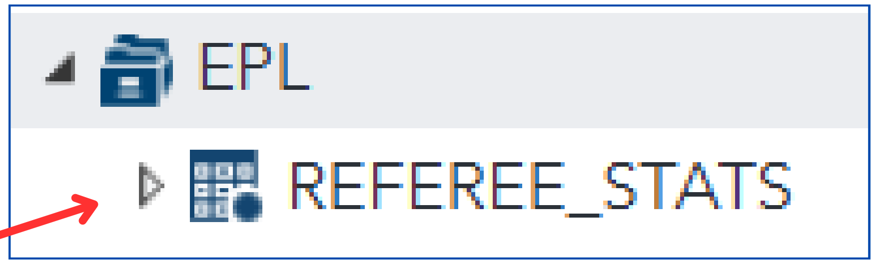
Saving output in a csv format

```
/* Exporting the table to a CSV file if needed */
proc export data=referee_stats
    outfile="/home/u63511609/BigDataFinalProject/refree/refree_stats.csv"
    dbms=csv
    replace;
run;
```



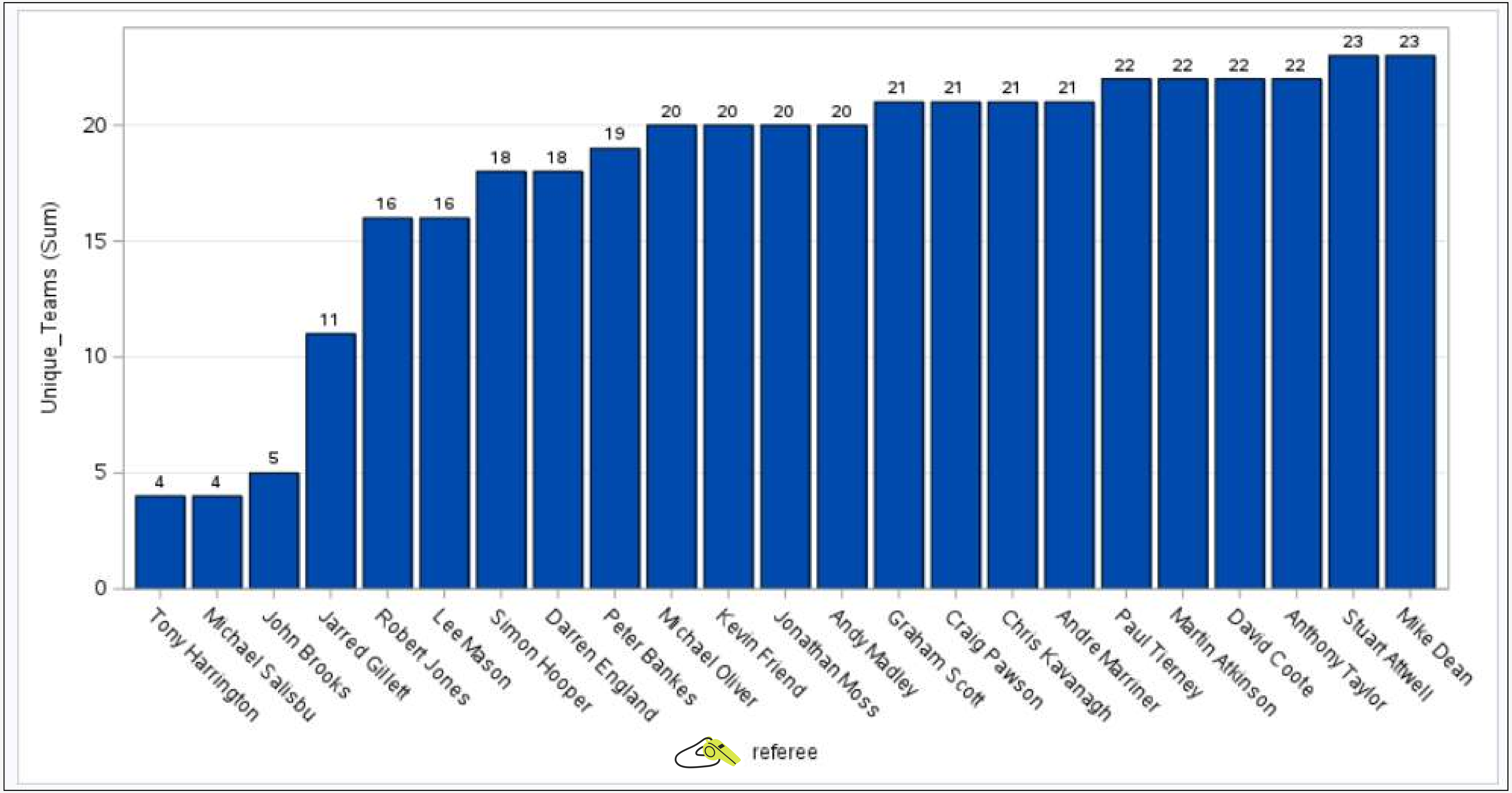
Saving the csv to EPL Library

```
/* Defining the EPL library */
libname EPL '/home/u63511609/BigDataFinalProject/EPL';

/* Saving the dataset directly in the EPL library */
data EPL.referee_stats;
    set referee_stats;
run;
```

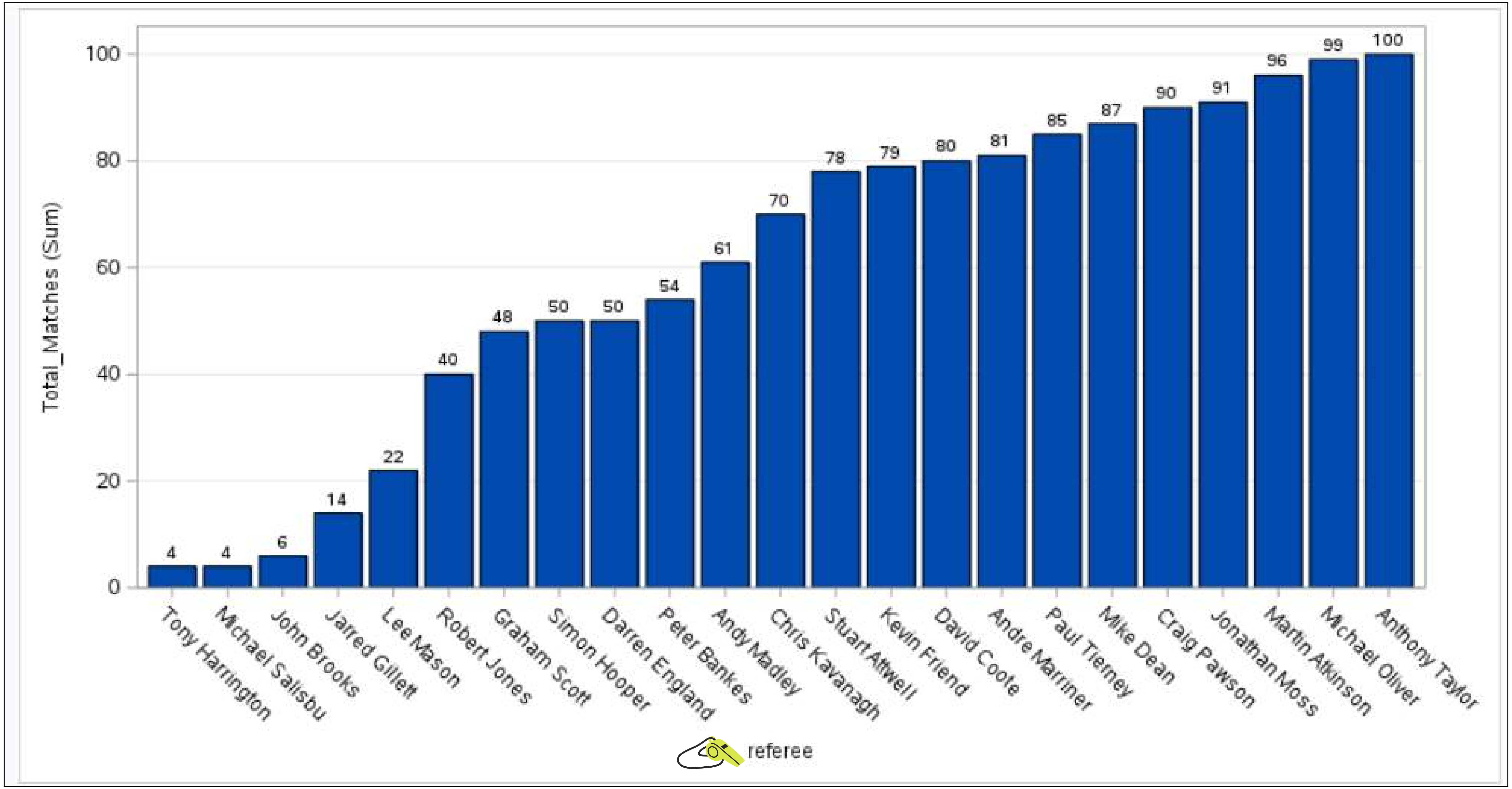


From 2020-09-12 to 2022-04-25

REFREES WHO MANGED MATCHES FOR DIFFERENT TEAMS IN EPL

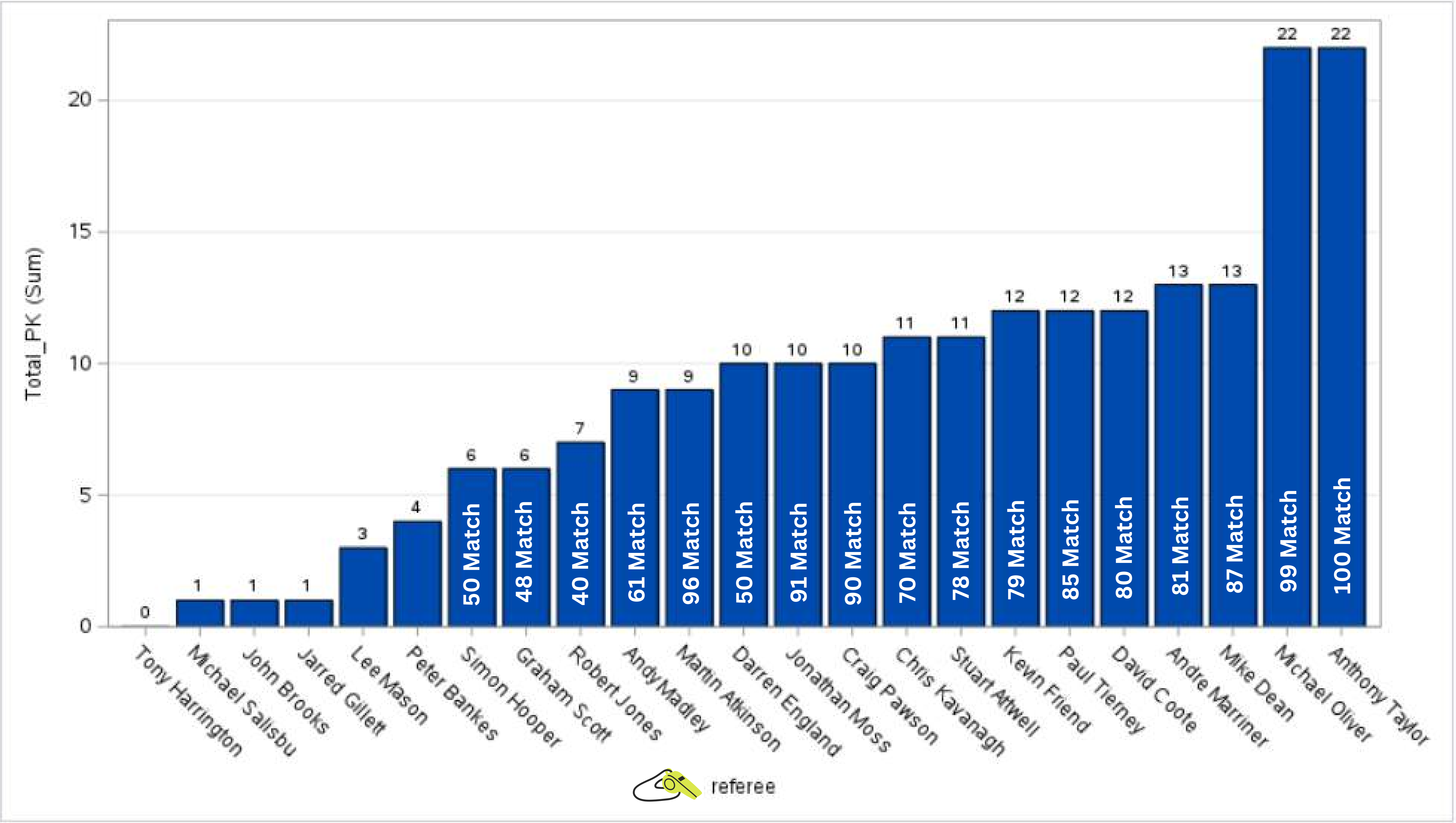


TOTAL MATCHES FOR EACH REFREE



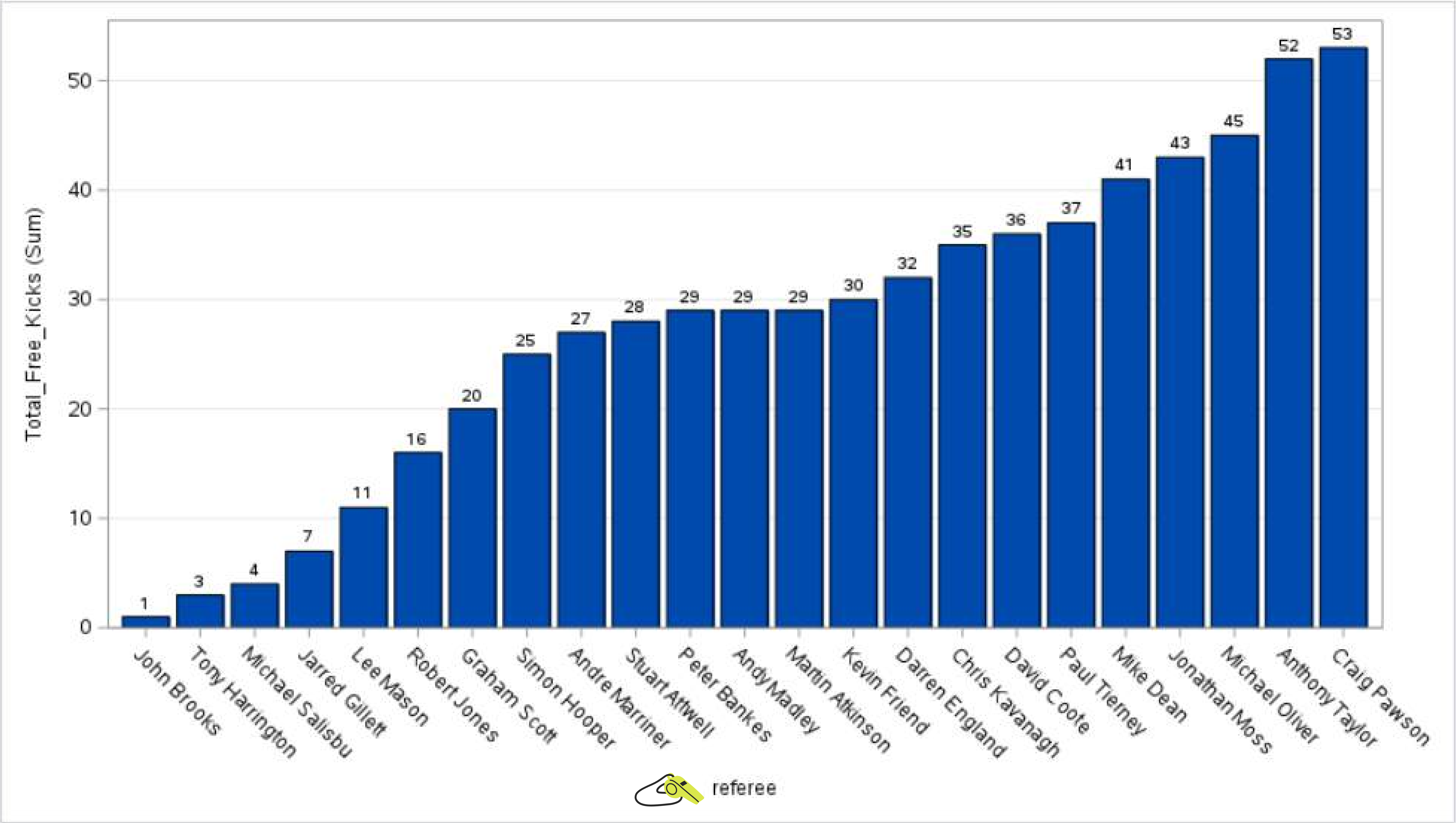
From 2020-09-12 to 2022-04-25

PK THAT ACCUARED UNDER CONTROL OF EACH REFREE IN EPL



From 2020-09-12 to 2022-04-25

FK THAT ACCUARED UNDER CONTROL OF EACH REFREE IN EPL



SAS CODE

Penalty Kicks Attempted For Top 6 Teams vs Other Team In English Premier League

```
%let top_6 = 'Manchester City', 'Chelsea', 'Arsenal', 'Tottenham Hotspur', 'Manchester United', 'Liverpool';
/* Filtering the data for matches where the team is in the top_6 list */
data filtered_matches;
    set matches;
    if team in (&top_6);
run;

/* Calculating the total pkatt for all matches */
proc sql;
    select sum(pkatt) as total_pkatt_all
    into :total_pkatt_all
    from matches;
quit;

/* Calculating the total pkatt for filtered matches */
proc sql;
    select sum(pkatt) as total_pkatt_filtered
    into :total_pkatt_filtered
    from filtered_matches;
quit;

/* Calculating  the difference */
%let diff = %sysevalf(&total_pkatt_all - &total_pkatt_filtered);

/* Creating a table for output */
data pkatt_summary;
    total_pkatt_all = &total_pkatt_all;
    total_pkatt_filtered = &total_pkatt_filtered;
    difference = &diff;
run;

/* Displaying the results */
proc print data=pkatt_summary noobs;
    title "PKATT Summary and Difference";
run;
```

From 2020-09-12 to 2022-04-25

OUT PUT

PKATT Summary and Difference

total_pkatt_all	total_pkatt_filtered	difference
205	54	151

26.34 %

TOP 6 IN EPL

73.65 %

OTHER TEAMS

DATA PREPROCESSING

Feature Transformation of venue

SAS CODE

```
proc sort data=permier_league;
    by venue;
run;

data permier_league;
    set permier_league;
    if venue = "Home" then venue_code = 1;
    else if venue = "Away" then venue_code = 0;
run;
```

OUTPUT

venue
Away
Home
Away
Away
Home
Away
Home
Home
Away
Away
Away
Home
Away
Home
Away
Home
Away
Home

Transform

venue_code
0
1
0
0
1
0
1
1
0
0
0
1
0
1
0
1
0
1

FEATURE ENGINEERING

Feature Transformation of opponent

SAS CODE

```
proc format;
    value $opp_code
        'Tottenham' = 1
        'Norwich City' = 2
        'Arsenal' = 3
        'Leicester City' = 4
        'Southampton' = 5
        'Chelsea' = 6
        'Liverpool' = 7
        'Burnley' = 8
        'Brighton' = 9
        'Crystal Palace' = 10
        'Manchester Utd' = 11
        'Everton' = 12
        'West Ham' = 13
        'Aston Villa' = 14
        'Watford' = 15
        'Wolves' = 16
        'Leeds United' = 17
        'Newcastle Utd' = 18
        'Brentford' = 19
        'Manchester City' = 20
        'Sheffield Utd' = 21
        'Fulham' = 22
        'West Brom' = 23
        other = .;  /* for Handling any missing or unlisted values */
run;
```

OUTPUT

Transform

opponent	opp_code
Tottenham	1
Leicester City	4
Chelsea	6
Liverpool	7
Brighton	9
Manchester Utd	11
Aston Villa	14
Watford	15
Newcastle Utd	18
Brentford	19
Arsenal	3
Southampton	5
Norwich City	2
Everton	12
Crystal Palace	10
Burnley	8
Arsenal	3
Liverpool	7

FEATURE ENGINEERING

Feature Extraction “Extracting Hour From Time”

SAS CODE

```
data permier_league;
    set permier_league;
    hour = input(scan(time, 1, ':'), 8.);
run;
```

OUTPUT

time
16:30:00.000
15:00:00.000
12:30:00.000
16:30:00.000
17:30:00.000
12:30:00.000
20:15:00.000
17:30:00.000
14:15:00.000
20:15:00.000
12:30:00.000
17:30:00.000
17:30:00.000
17:30:00.000
20:00:00.000
15:00:00.000
16:30:00.000
17:30:00.000

Hour Feature From Time

hour
5
5
4
5
6
4
7
6
5
7
4
6
6
6
7
5
5
6

FEATURE ENGINEERING

Feature Extraction “Extracting Target Feature”

SAS CODE

```
data permier_league;
    set permier_league;

    if result = 'W' then target = 1;
    else target = 0;
run;

proc print data=permier_league (obs=5);
    var result target;
run;
```

1	→	Winning
0	→	Not Winning

OUTPUT

result
L
W
W
D
W
W
W
W
W
W
W
D
W
W
D
W
W
D

Target Feature
From result →

target
0
1
1
0
1
1
1
1
1
1
1
0
1
1
0
1
1
0

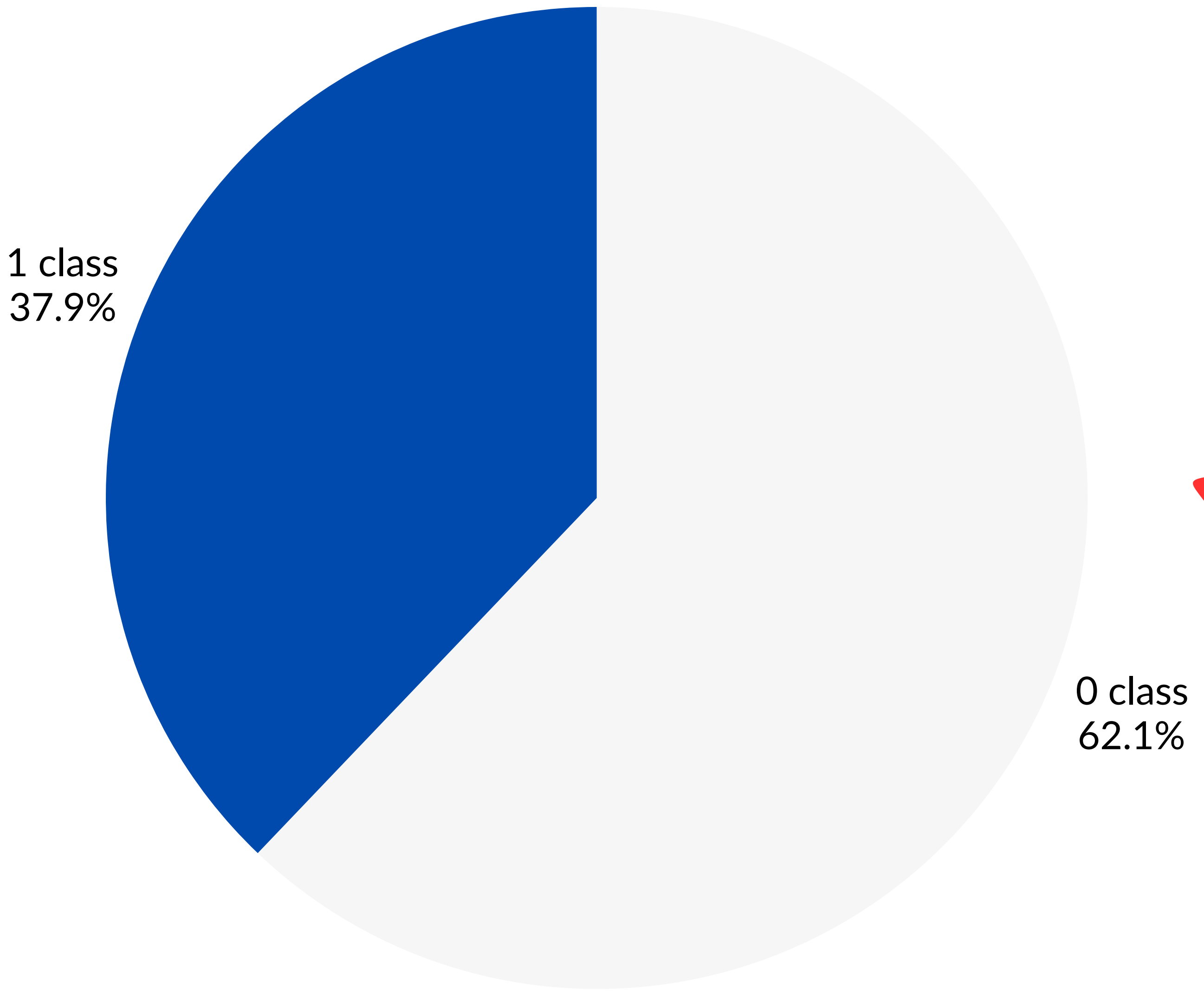
CKECKING IF DATA BALANCED OR NOT

SAS CODE

```
proc freq data=permier_league;
    tables target / nocum nopercent;
run;
```

The FREQ Procedure

target	Frequency
0	863
1	526



Data is approximately balanced

BIG DATA COURSE FINAL PROJECT

UPLOADING DATA TO SAS VIYA

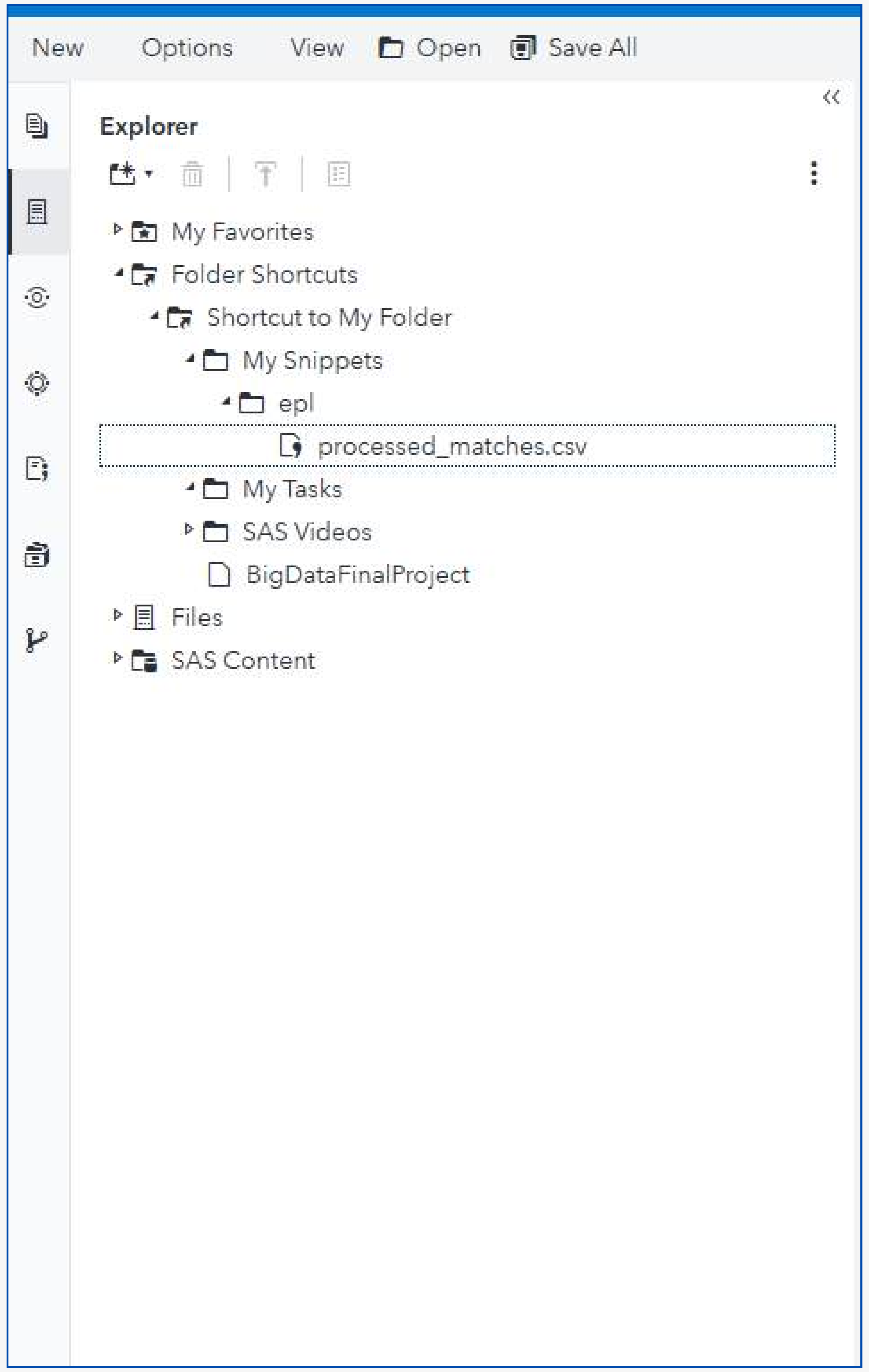
SAVING THE DATASET AFTER PROCESSING IN A NEW CSV FILE

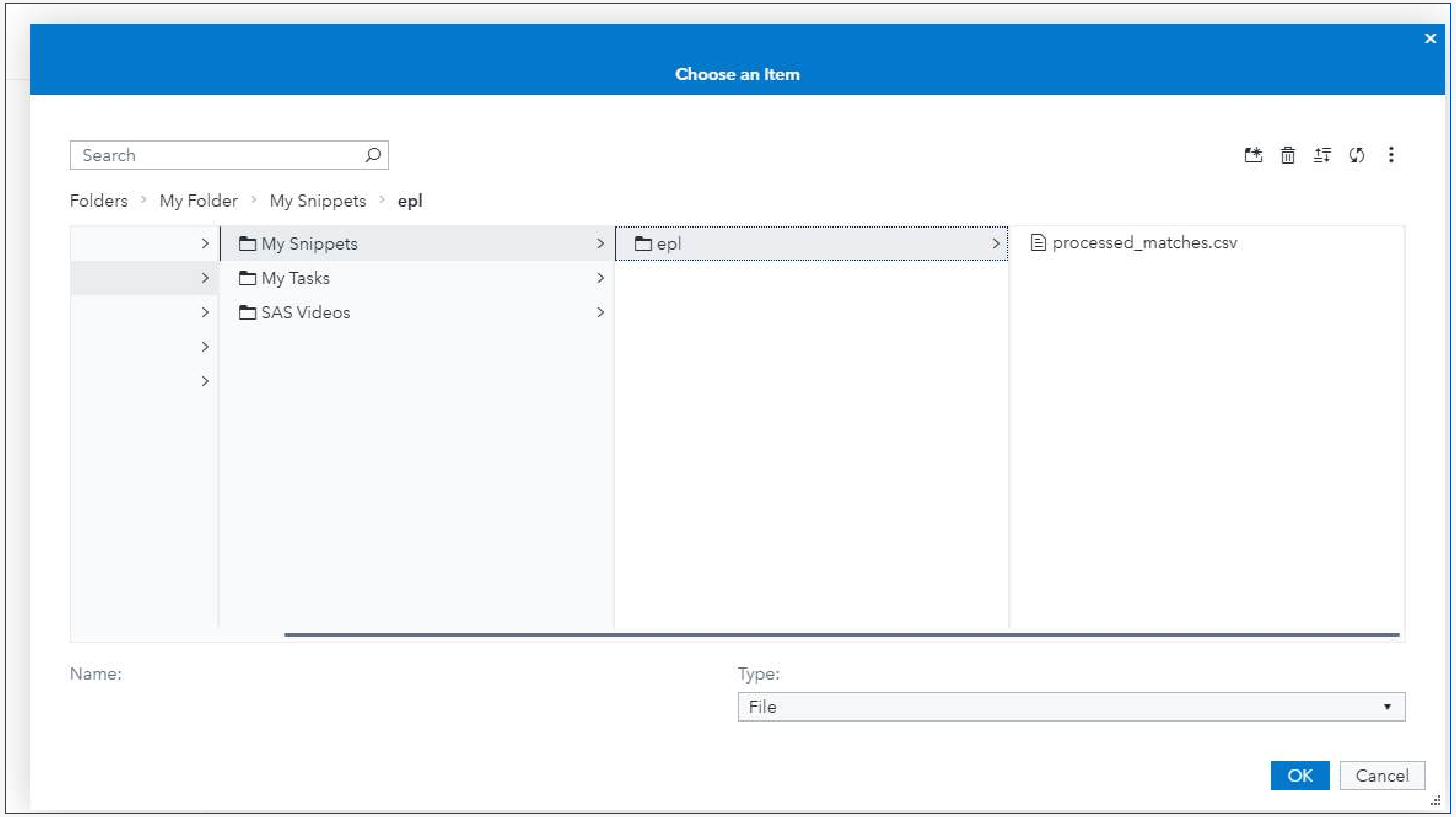
SAS CODE

```
proc export data=permier_league
    outfile="/home/u63511609/BigDataFinalProject/processed_matches.csv"
    dbms=csv
    replace;  /* Overwrite the file if it already exists */
run;
```

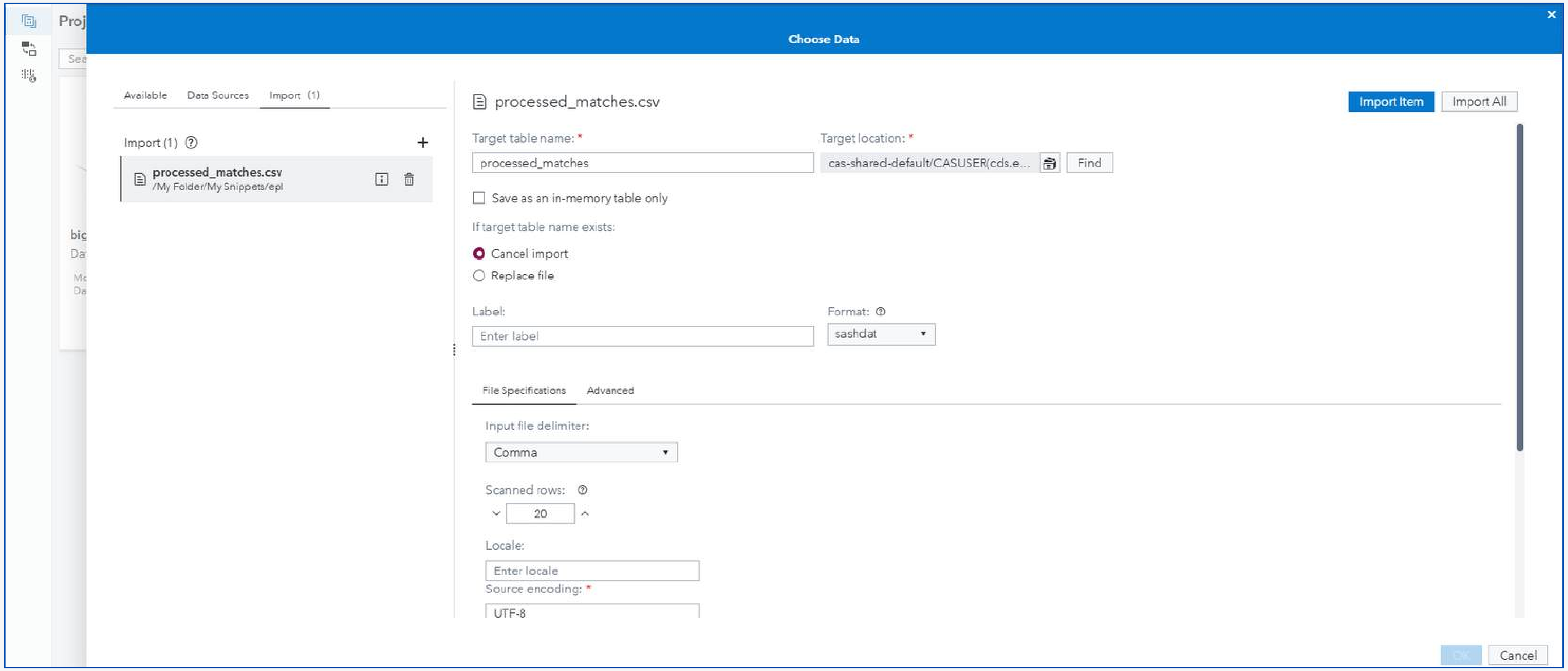
- BigDataFinalProject
  - EdaAndFeatureEngineering.sas
  - EdaAndFeatureEngineering.sas~
  - matches.csv
  - processed_matches.csv

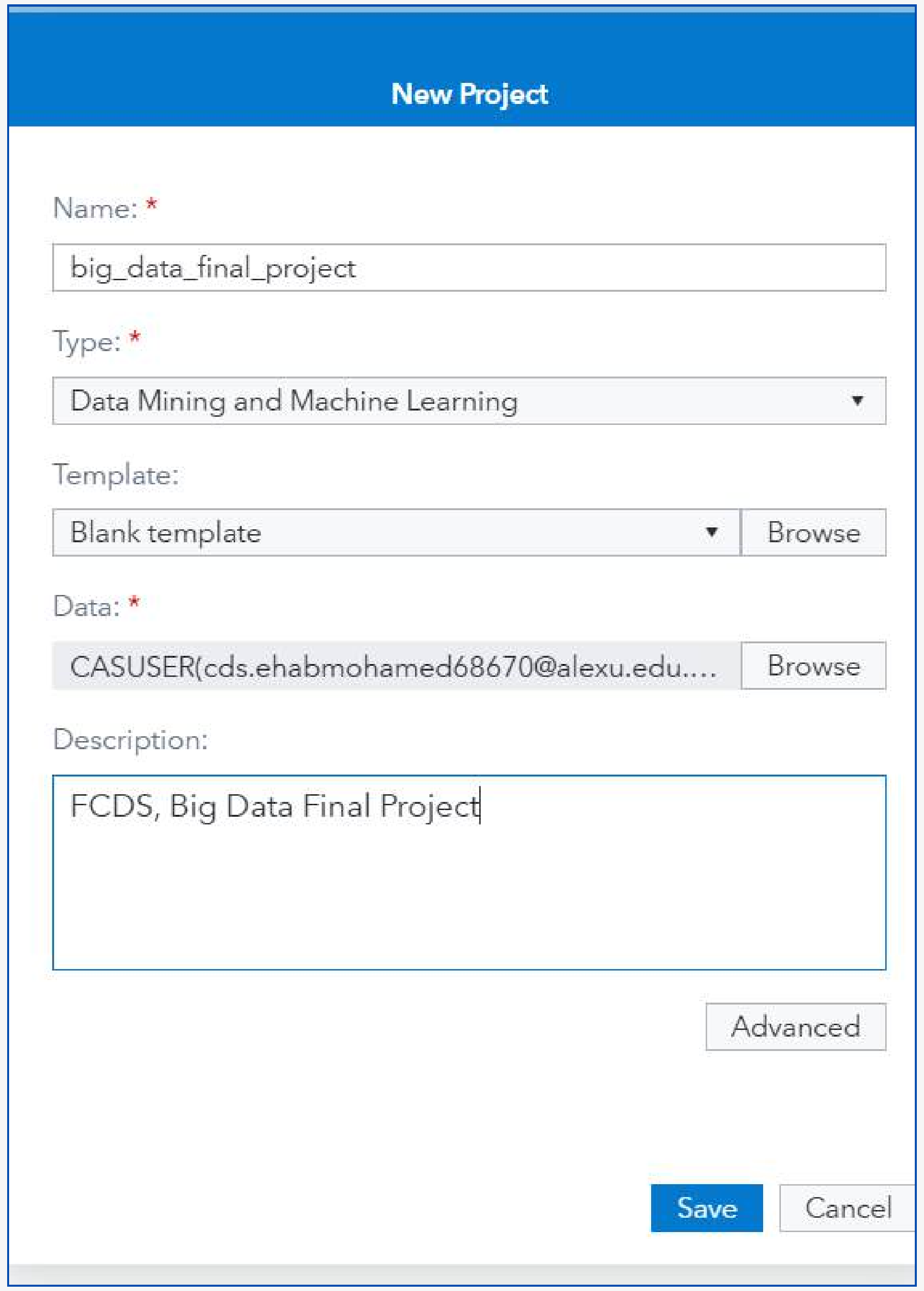
UPLOADING PROCESSED DATA TO SAS VIYA





UPLOADING PROCESSED DATA TO SAS VIYA





MODELS BUILDING

Football Match Unpredictability

- Football matches are inherently difficult to predict due to random events like goals, injuries, or referee decisions.
- Low Misclassification rates are not uncommon in sports prediction models because of this randomness
- they might still be reasonable compared to a random guess baseline
- The model could still provide valuable insights and predictions for football matches, especially when combined with expert analysis or other strategies.

SELECTED MODEL	REASONS OF SELECTION
Random Forest	1. Handles Complex and Nonlinear Data Football match outcomes depend on multiple interacting factors such as: • Team statistics: goals scored, goals conceded, possession percentage. • Player performance: passes completed, shots on target, player fitness. 2. Robust to Overfitting Random Forest combines multiple decision trees using a bagging approach: • It reduces the risk of overfitting by averaging predictions from many trees. • Overfitting is common in football models with limited data, but Random Forest minimizes this risk. 3. Works Well with Categorical and Numerical Data & Random Forest handles both data types naturally without extensive preprocessing Football analysis often includes both categorical data like home/away and numerical data like goals scored, possession)
Gaussian Processes Classification	**1. Works Well with Small Datasets** • Football analysis often faces limited training data, especially for specific leagues or teams. • GNB performs well on **small datasets** where complex models like neural networks might overfit. This is because it requires fewer data points to estimate the parameters of a Gaussian distribution. 2. Good Baseline Model Gaussian Naive Bayes is a strong baseline model for football classification tasks. It provides a simple and interpretable starting point. 3. Computationally Efficient GNB is a fast and lightweight model which makes it ideal for Real-time predictions in football.

FEATURES SELECTION

big_data_final_project

Data | Pipelines | Pipeline Comparison | Insights

Filter

Variable Name	Role
attendance	ID
day_code	Input
hour	Input
opp_code	Input
venue_code	Input
dist	Rejected
fk	Rejected
ga	Rejected
gf	Rejected
pk	Rejected
pkatt	Rejected
poss	Rejected
season	Rejected
sh	Rejected
sot	Rejected
xg	Rejected
xga	Rejected
captain	Rejected

Variable Importance			
Variable	Importance	Std Dev Importance	Relative Importance
opp_code	37.4447	7.7791	1.0000
hour	20.3868	6.6453	0.5445
day_code	15.0236	4.5908	0.4012
venue_code	6.4356	3.2241	0.1719

DATA SPLITTING

Project Settings

- Partition Data
- Event-Based Sampling
- Node Configuration
- Rules
- Output Library
- Logging
- Compute Context

Partition Data

Create partition variable

Note: These settings are active only when a partition variable is not set within the data. Using a data source with a pre-defined partition variable or manuall selecting a partition variable will override these settings.

Method:

Stratify

Training:

60 60.00%

Validation:

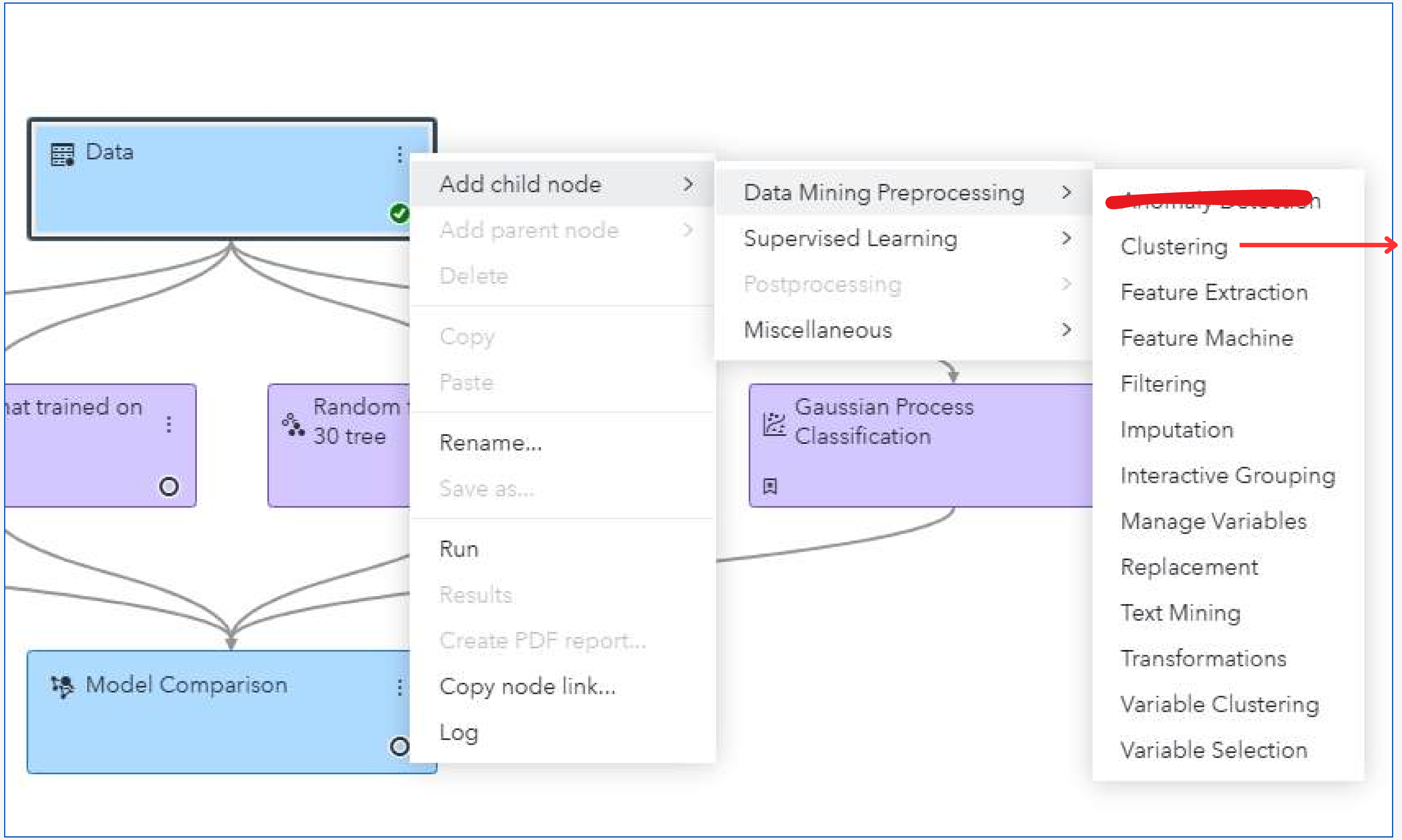
30 30.00%

Test:

10 10.00%

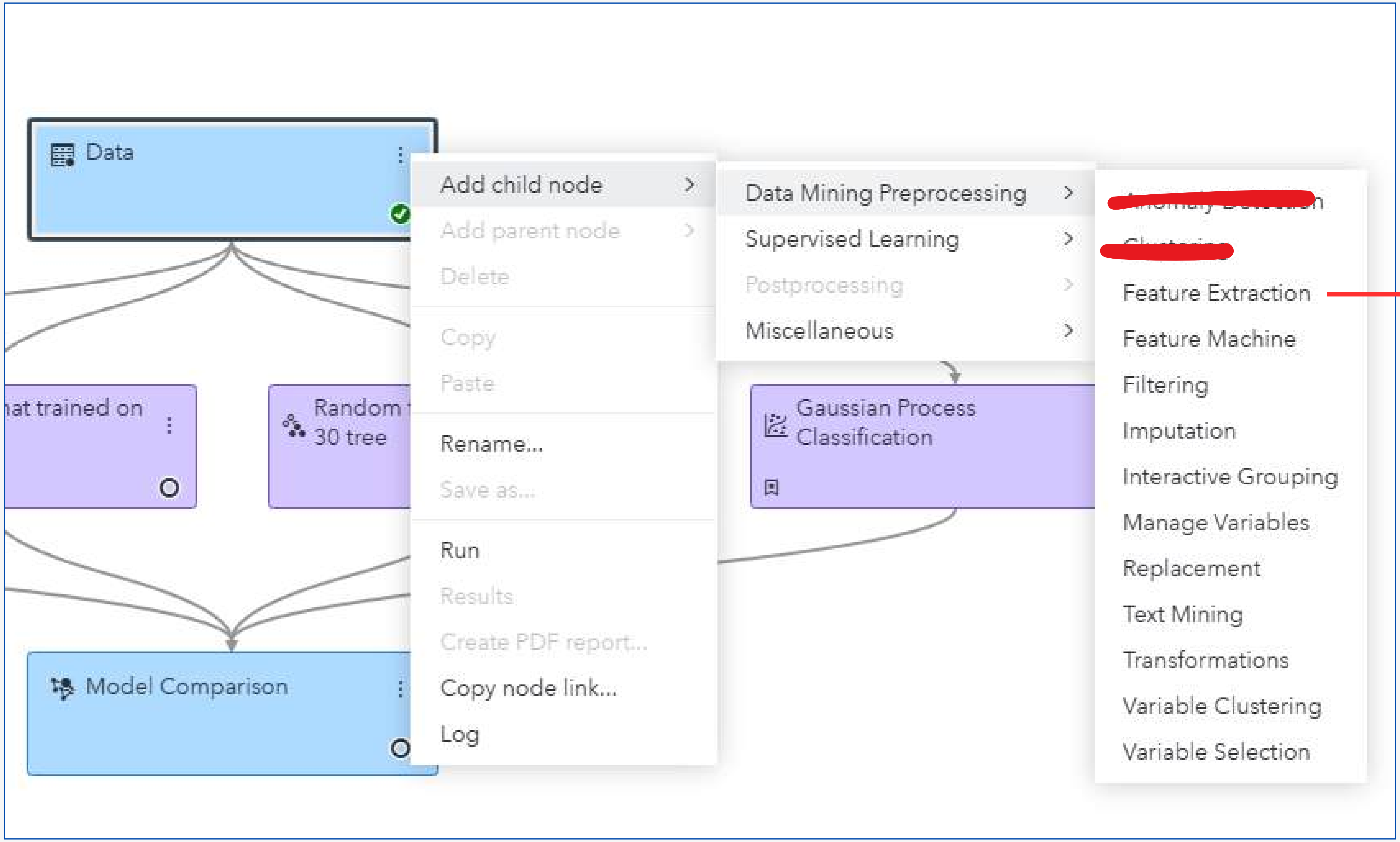
	Training	Validation	Test	Total
Number of Observations Read	833	417	139	1389
Number of Observations Used	833	417	139	1389

MODELS BUILDING BY SAS VIYA



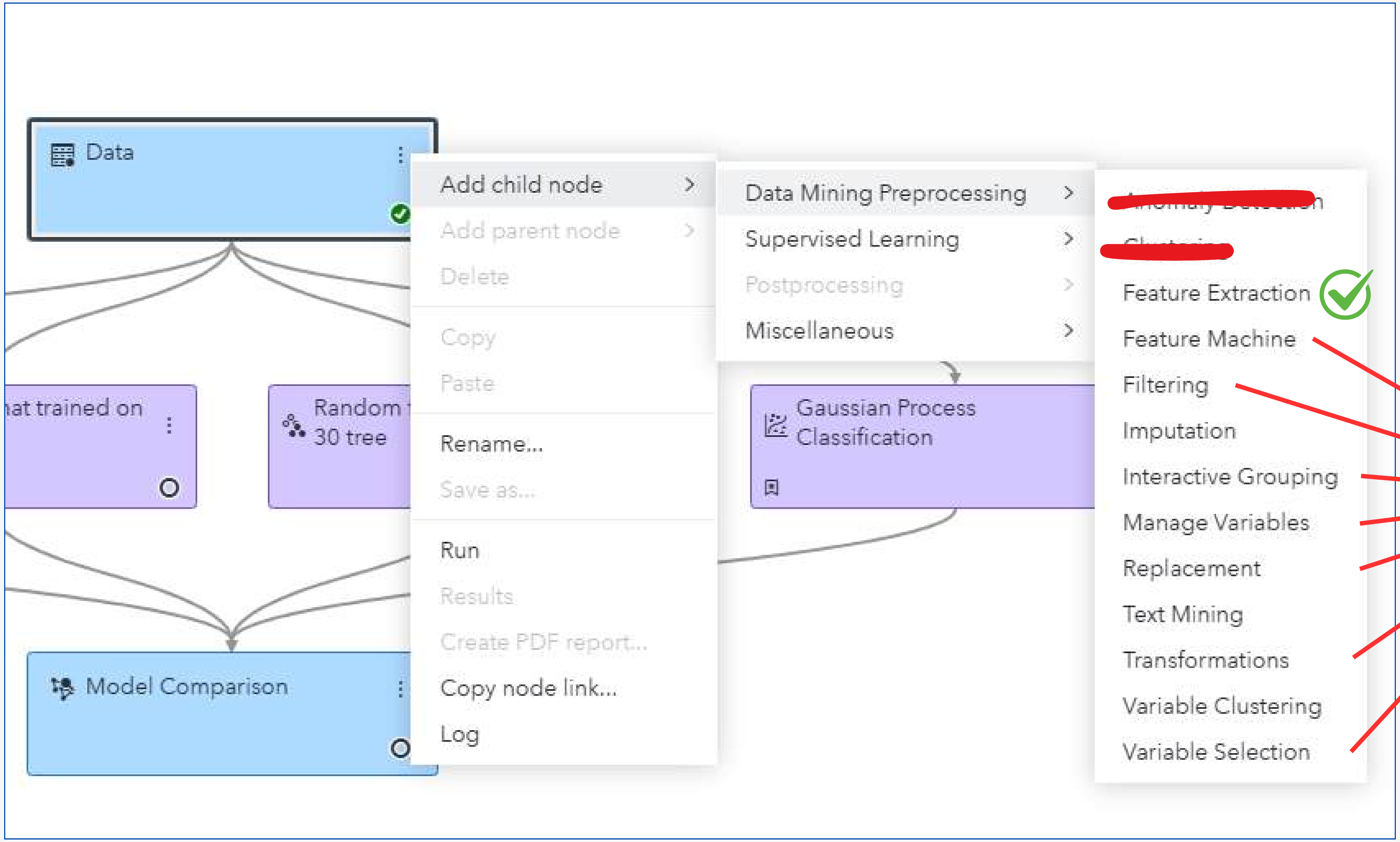
The Model that will use is a supervised model and target is completely label so the problem of When you there is a small amount of labeled data and a large amount of unlabeled data is not found

MODELS BUILDING BY SAS VIYA



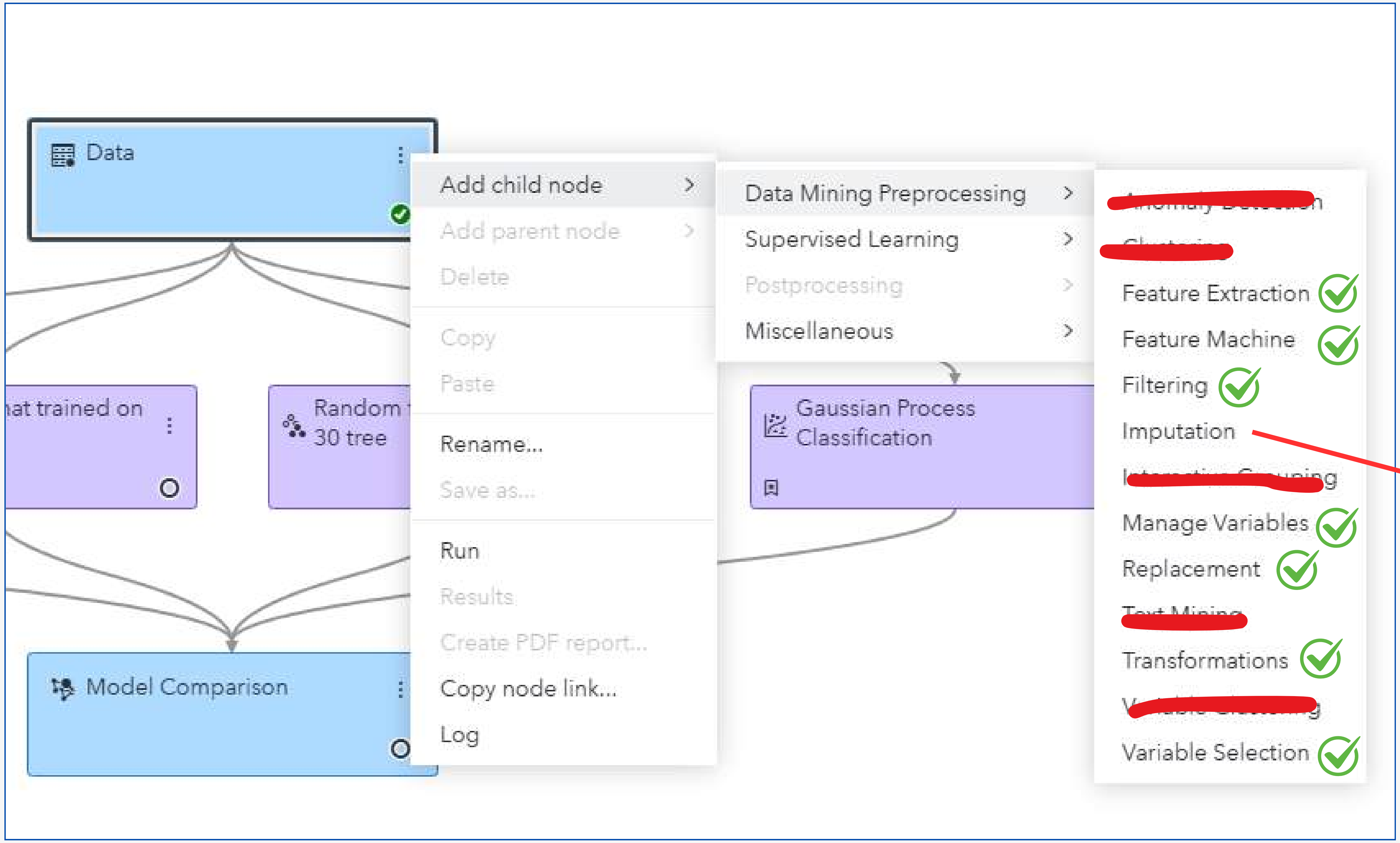
Done by sas studio code when target feature extracted from result feature

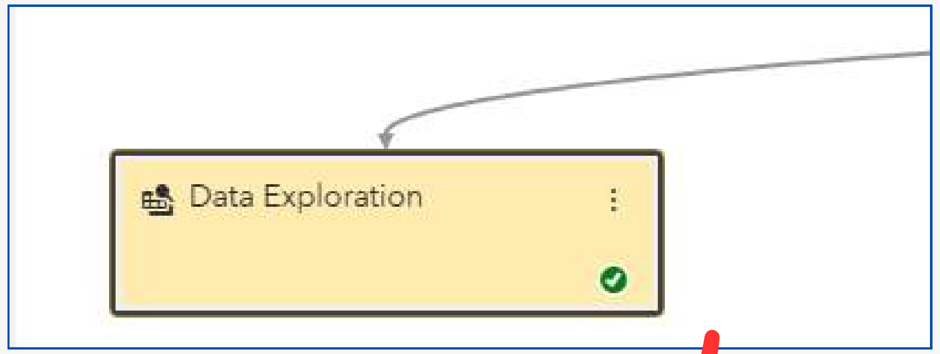
MODELS BUILDING BY SAS VIYA



All feature engineering process done with sas code and feature selection done based on domain knowledge of the target problem

MODELS BUILDING BY SAS VIYA



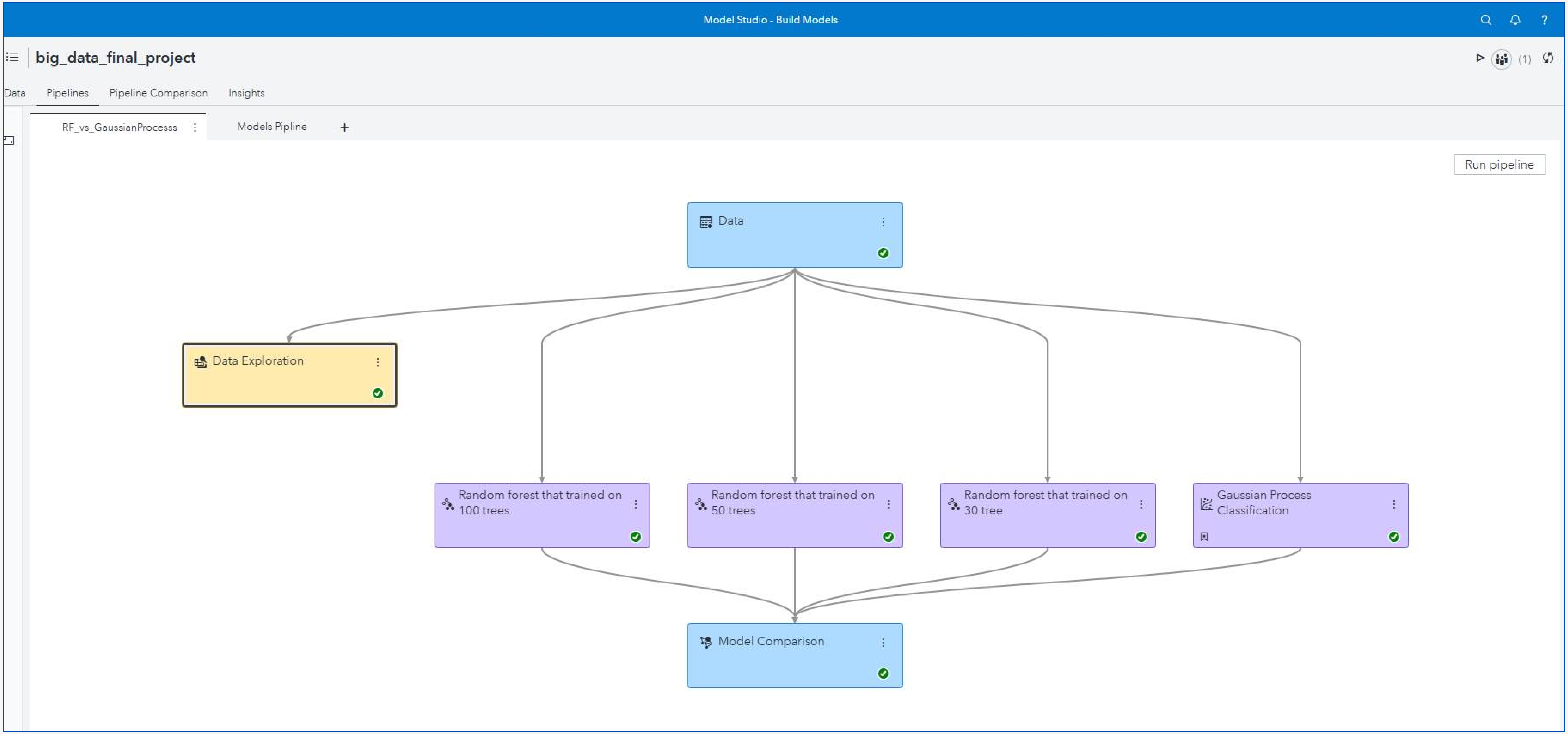


Missing Values

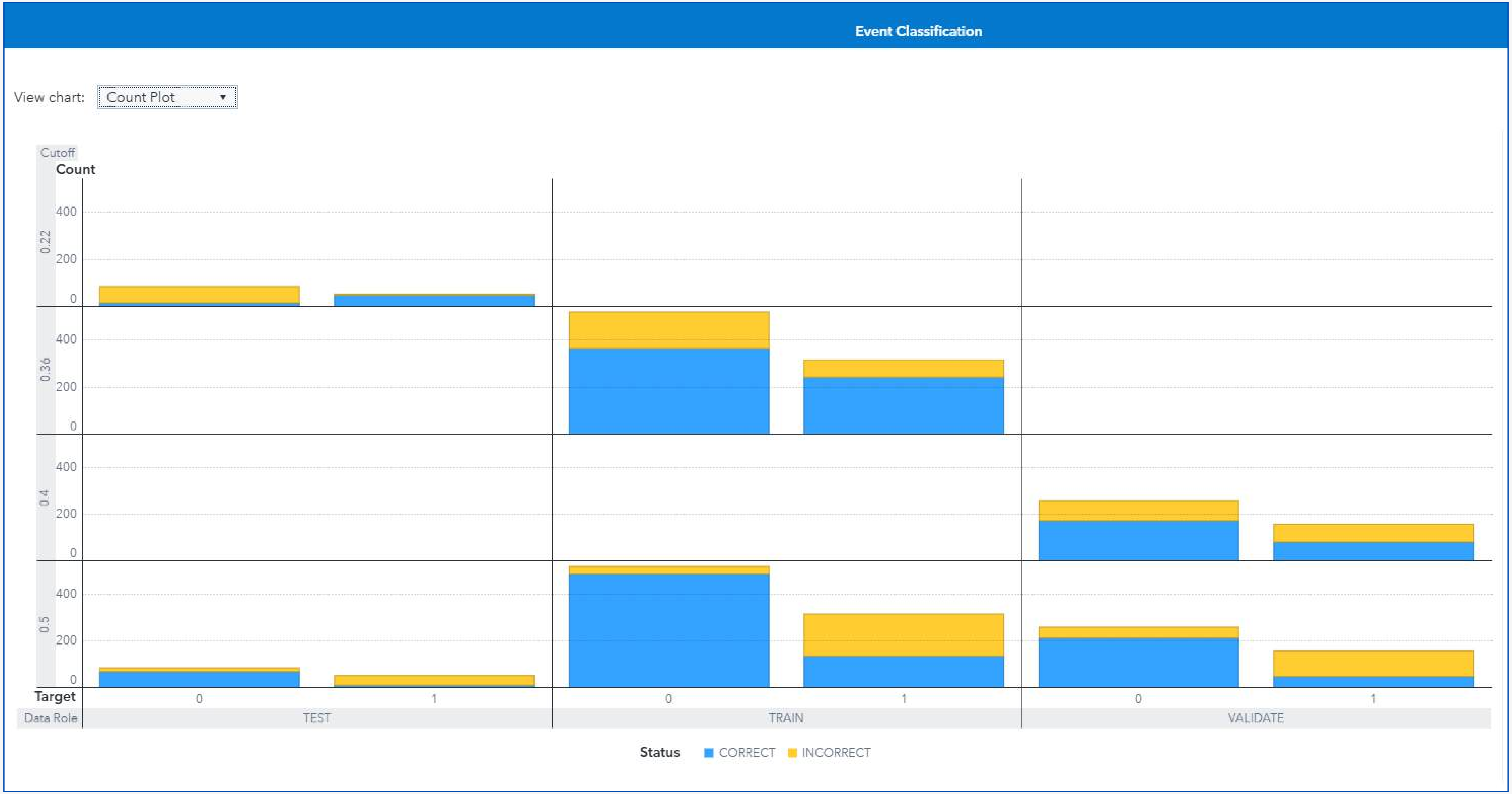
Variable Name	Number of Missing Values	Percentage Missing
day_code	0	0
hour	0	0
opp_code	0	0
target	0	0
venue_code	0	0

No Need To Imputation

MODELS BUILDING BY SAS VIYA



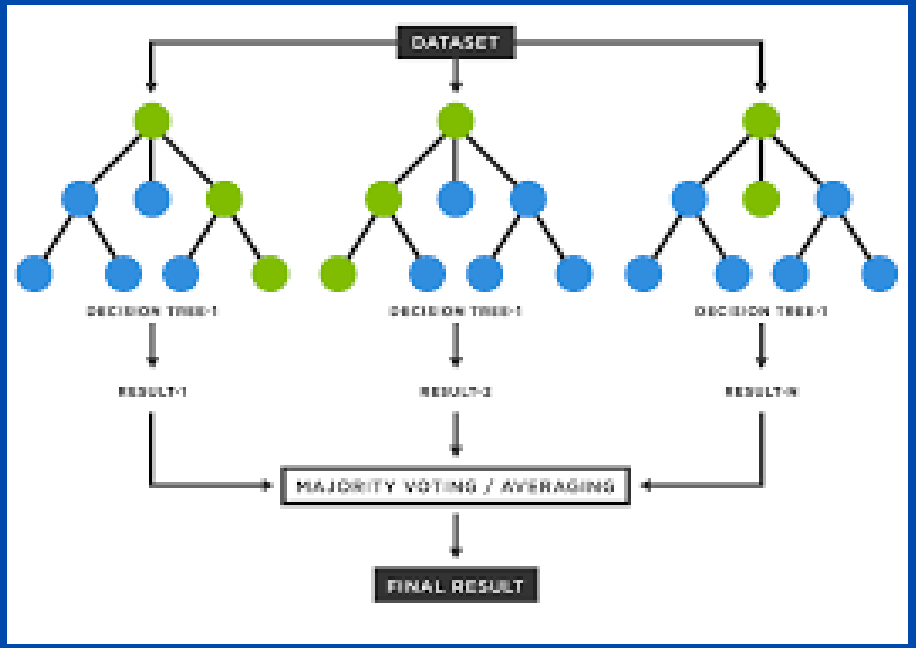
RANDOM FOREST EVALUATION



RANDOM FOREST EVALUATION

Fit Statistics										
Target ...	Data Role	Partitio...	Formatt...	Numbe...	Averag...	Divisor ...	Area Unde... ↑	Root Av...	Misclas...	Multi-Cl...
target	TEST	2	2	139	0.2503	139	0.5325	0.5003	0.4101	0.6942
target	VALIDATE	0	0	417	0.2340	417	0.5953	0.4838	0.3717	0.6631
target	TRAIN	1	1	833	0.1790	833	0.8205	0.4231	0.2545	0.5370

Gini Co...	Gamma	Tau	KS Cutoff	KS at U...	Misclas...	Misclass...
0.0649	0.0668	0.0309	0.2200	0.0404	0.5180	0.4101
0.1907	0.1946	0.0900	0.4000	0.1325	0.3885	0.3717
0.6411	0.6527	0.3019	0.3600	0.3680	0.2725	0.2545



USING RANDOM FOREST MODEL **ON** UNSEEN DATA

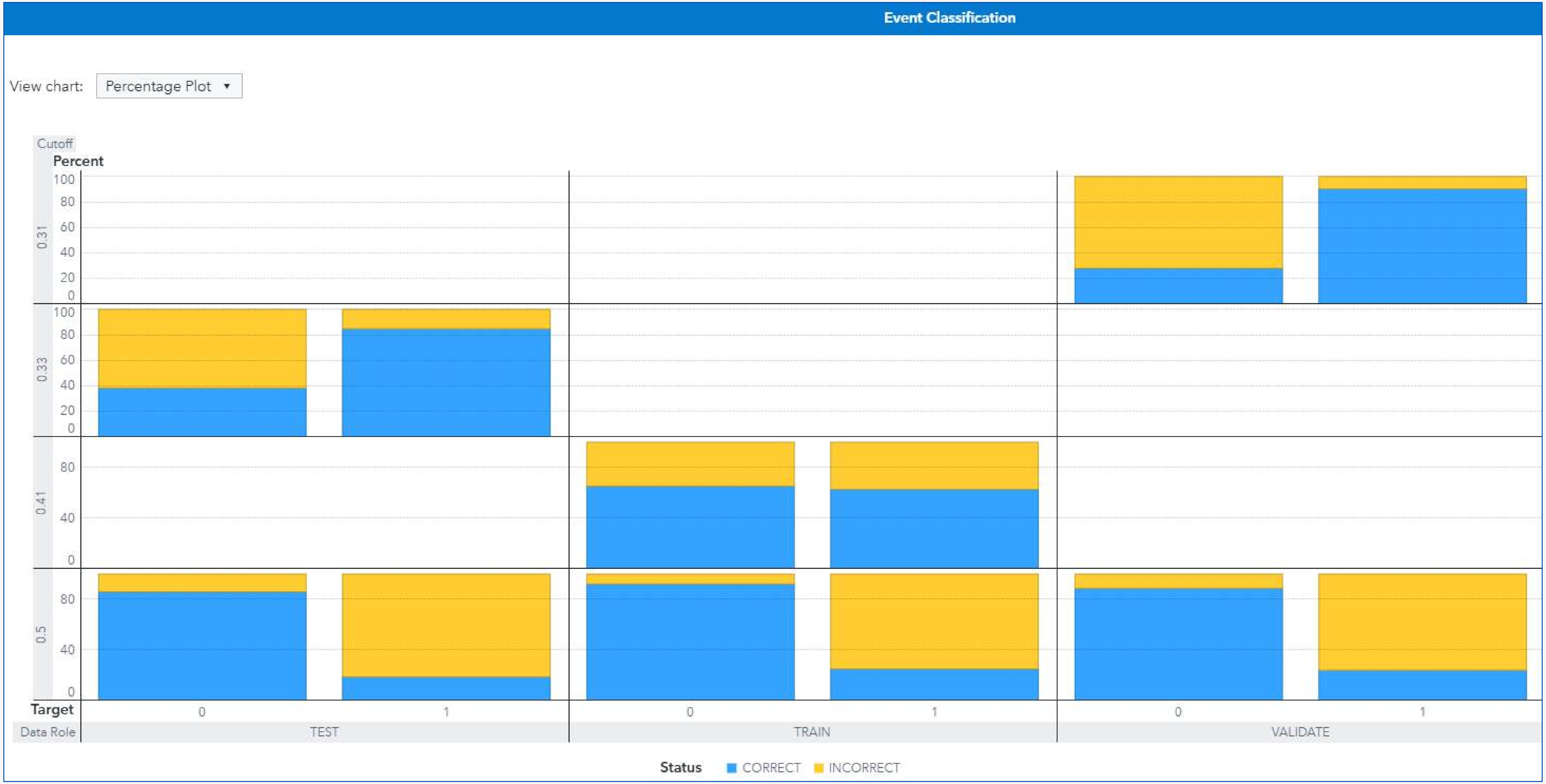
Model exposed to 417 Match in Vaildation Set

Predicted **246** Match True

Model exposed to 139 Match in Testing Set

Predicted **74** Match True

GAUSSIAN PROCESSES CLASSIFICATION EVALUATION

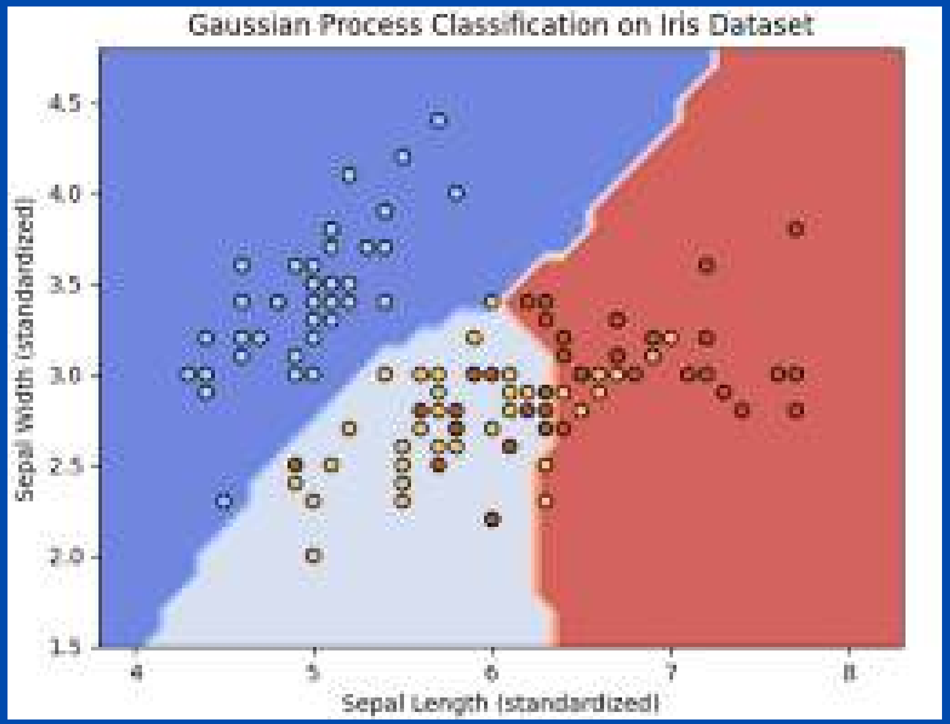


GAUSSIAN PROCESSES CLASSIFICATION EVALUATION

Fit Statistics

Target ...	Data Role	Partitio...	Formatt...	Numbe...	Average Squared Error	Divisor ...	Root Av...	Misclas...	Multi-Cl...	KS (You...	Area Un...	Gini Co...
target	TEST	2	2	139	0.2321	139	0.4818	0.3957	0.6575	0.2328	0.6124	0.2249
target	TRAIN	1	1	833	0.2094	833	0.4576	0.3313	0.6064	0.2811	0.6929	0.3858
target	VALIDATE	0	0	417	0.2243	417	0.4736	0.3573	0.6381	0.1869	0.6134	0.2268

Misclas...	Multi-Cl...	KS (You...	Area Un...	Gini Co...	Gamma	Tau	KS Cutoff	KS at U...
0.3957	0.6575	0.2328	0.6124	0.2249	0.2394	0.1069	0.3300	0.0491
0.3313	0.6064	0.2811	0.6929	0.3858	0.4048	0.1817	0.4100	0.1736
0.3573	0.6381	0.1869	0.6134	0.2268	0.2377	0.1070	0.3100	0.1285



USING GAUSSIAN PROCESSES CLASSIFICATION MODEL ON UNSEEN DATA

Model exposed to 417 Match in Vaildation Set

Predicted **255** Match True

Model exposed to 139 Match in Testing Set

Predicted **85** Match True

CHAMPION MODEL

Model Comparison

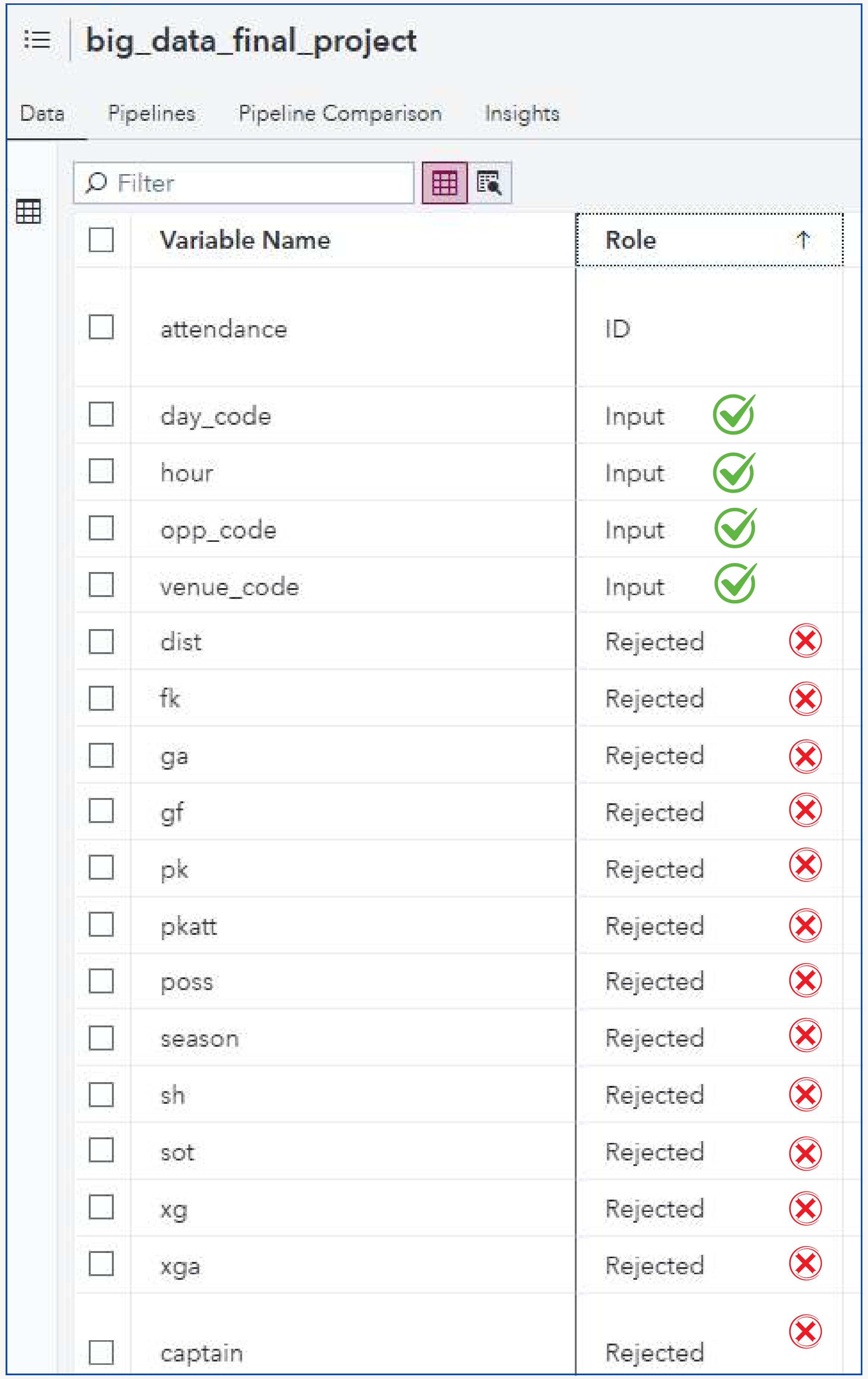
Champi...	Name	Algorith...	KS (You...	Accuracy	Averag...	Area Un...	Cumula...	Cumula...	Cutoff	Data Role	Depth	F1 Score	False Di...	False Po...	Gain	Gini Co...	ROC Se.
★	Gaussian Process Classification	Gaussian Process Classification	0.2328	0.6043	0.2321	0.6124	1.4151	14.1509	0.5000	TEST	10	0.2667	0.5455	0.1395	0.4151	0.2249	0.049
	Random forest that trained on 100 trees	Forest	0.1411	0.5899	0.2503	0.5325	0.7547	7.5472	0.5000	TEST	10	0.2963	0.5714	0.1860	-0.2453	0.0649	0.040
	Random forest that trained on 50 trees	Forest	0.1178	0.5971	0.2508	0.5323	0.7547	7.5472	0.5000	TEST	10	0.3333	0.5484	0.1977	-0.2453	0.0645	0.066
	Random forest that trained on 30 tree	Forest	0.0946	0.5971	0.2519	0.5293	0.9434	9.4340	0.5000	TEST	10	0.3171	0.5517	0.1860	-0.0566	0.0586	0.059

BIG DATA COURSE FINAL PROJECT

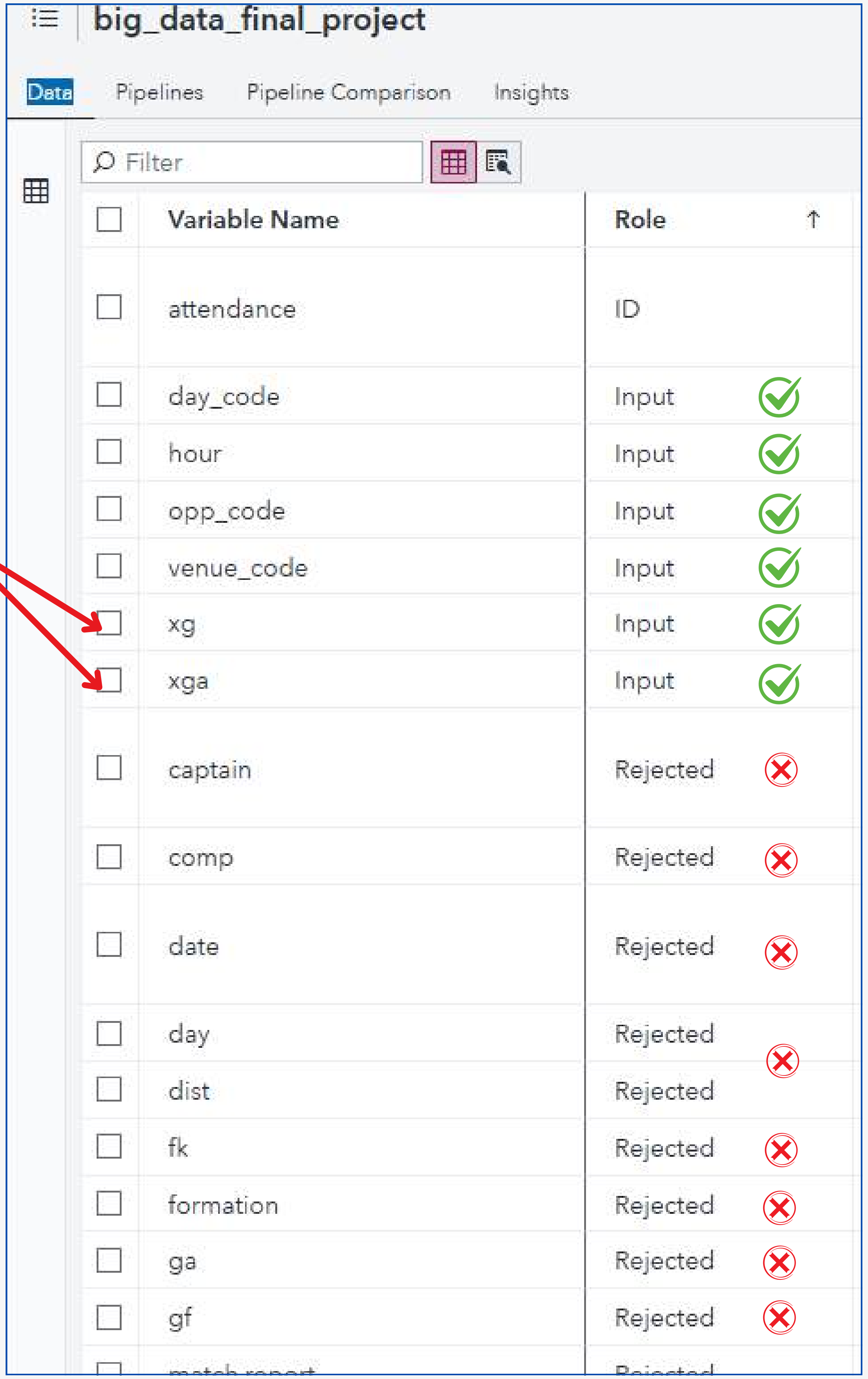
IMPROVING MODEL PERFROMANCE

FACULTY OF COMPUTERS AND DATA SCIENCE

FEATURES SELECTION

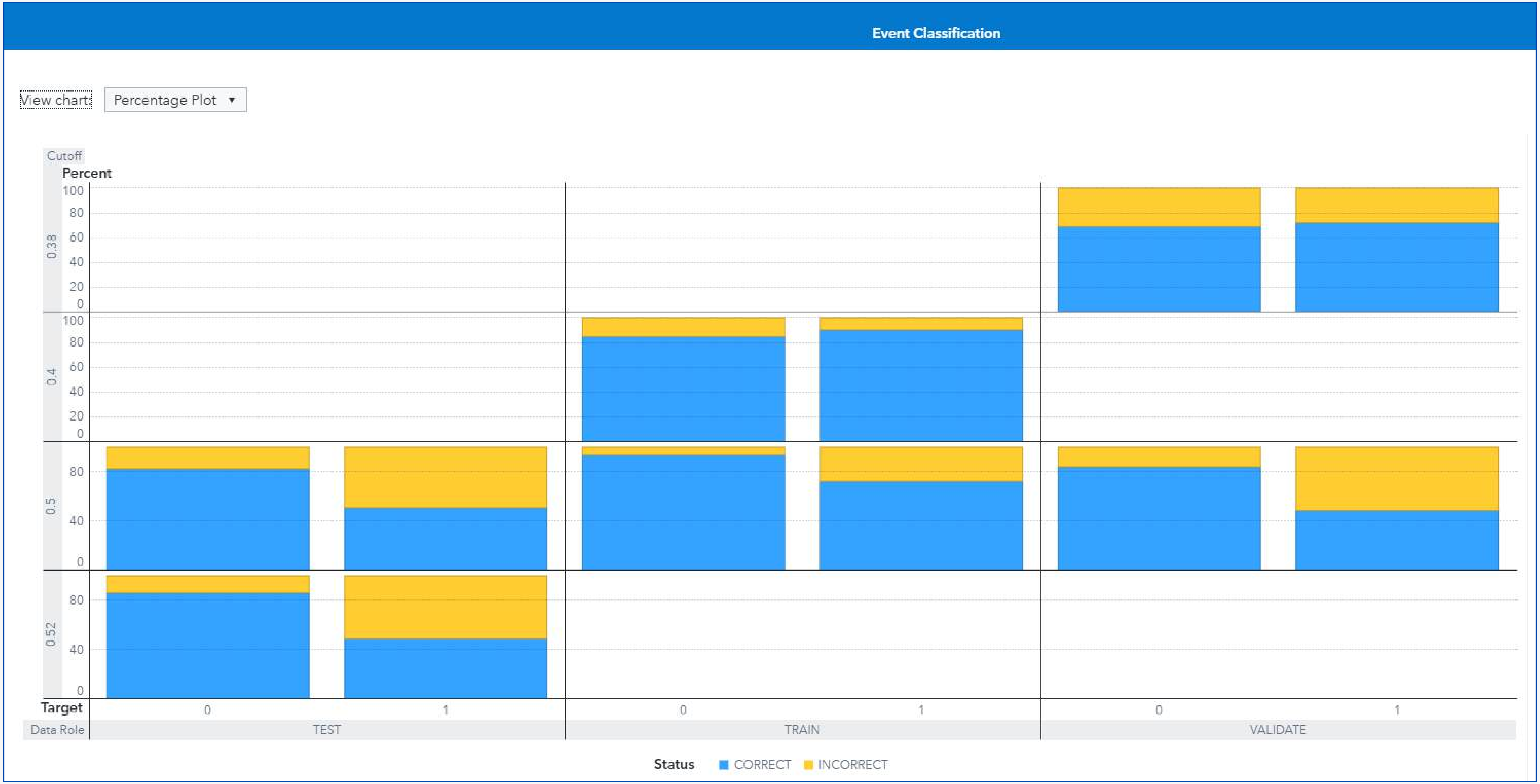


Adding xg, xga fratures and retraining the models



RANDOM FOREST EVALUATION

After Adding XG, XGA



RANDOM FOREST EVALUATION

After Adding XG, XGA

Fit Statistics

Target ...	Data Role	Partitio...	Formatt...	Numbe...	Averag...	Area Under ROC ↓	Divisor ...	Root Av...	Misclas...	Multi-Cl...
target	TRAIN	1	1	833	0.1169	0.9422	833	0.3419	0.1429	0.3826
target	VALIDATE	0	0	417	0.1802	0.7800	417	0.4244	0.2926	0.5348
target	TEST	2	2	139	0.1980	0.7318	139	0.4450	0.2950	0.5816

Fit Statistics

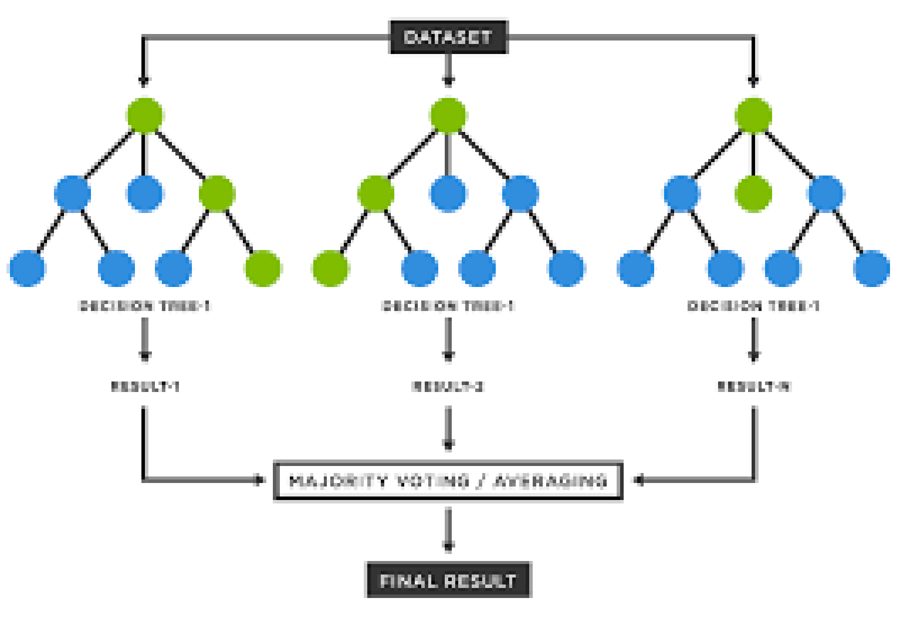
Root Av...	Misclas...	Multi-Cl...	KS (You...	Gini Co...	Gamma	Tau	KS Cutoff	KS at U...	Misclas...	Misclass...
0.3419	0.1429	0.3826	0.7491	0.8843	0.8883	0.4164	0.4000	0.6620	0.1321	0.1429
0.4244	0.2926	0.5348	0.4126	0.5600	0.5659	0.2642	0.3800	0.3290	0.2974	0.2926
0.4450	0.2950	0.5816	0.3510	0.4636	0.4697	0.2203	0.5200	0.3350	0.2806	0.2950

RANDOM FOREST EVALUATION
After Adding XG, XGA

Event Classification

View chart: Table

Cutoff	Cutoff Source	Target Name	Response	Event	Value	Training Frequ...	Validation Freq...	Test Frequency	Training Percen...
0.3800	KS	target	CORRECT	1	True Positive	.	114	.	.
0.3800	KS	target	INCORRECT	1	False Negative	.	44	.	.
0.3800	KS	target	CORRECT	0	True Negative	.	179	.	.
0.3800	KS	target	INCORRECT	0	False Positive	.	80	.	.
0.4000	KS	target	CORRECT	1	True Positive	284	.	.	90.1587
0.4000	KS	target	INCORRECT	1	False Negative	31	.	.	9.8413
0.4000	KS	target	CORRECT	0	True Negative	439	.	.	84.7490
0.4000	KS	target	INCORRECT	0	False Positive	79	.	.	15.2510
0.5000	Default	target	CORRECT	1	True Positive	228	77	27	72.3810
0.5000	Default	target	INCORRECT	1	False Negative	87	81	26	27.6190
0.5000	Default	target	CORRECT	0	True Negative	486	218	71	93.8224
0.5000	Default	target	INCORRECT	0	False Positive	32	41	15	6.1776
0.5200	KS	target	CORRECT	1	True Positive	.	.	26	.
0.5200	KS	target	INCORRECT	1	False Negative	.	.	27	.
0.5200	KS	target	CORRECT	0	True Negative	.	.	74	.
0.5200	KS	target	INCORRECT	0	False Positive	.	.	12	.



USING RANDOM FOREST MODEL ON UNSEEN DATA

After Adding XG, XGA

- Model exposed to 417 Match in Vaildation Set

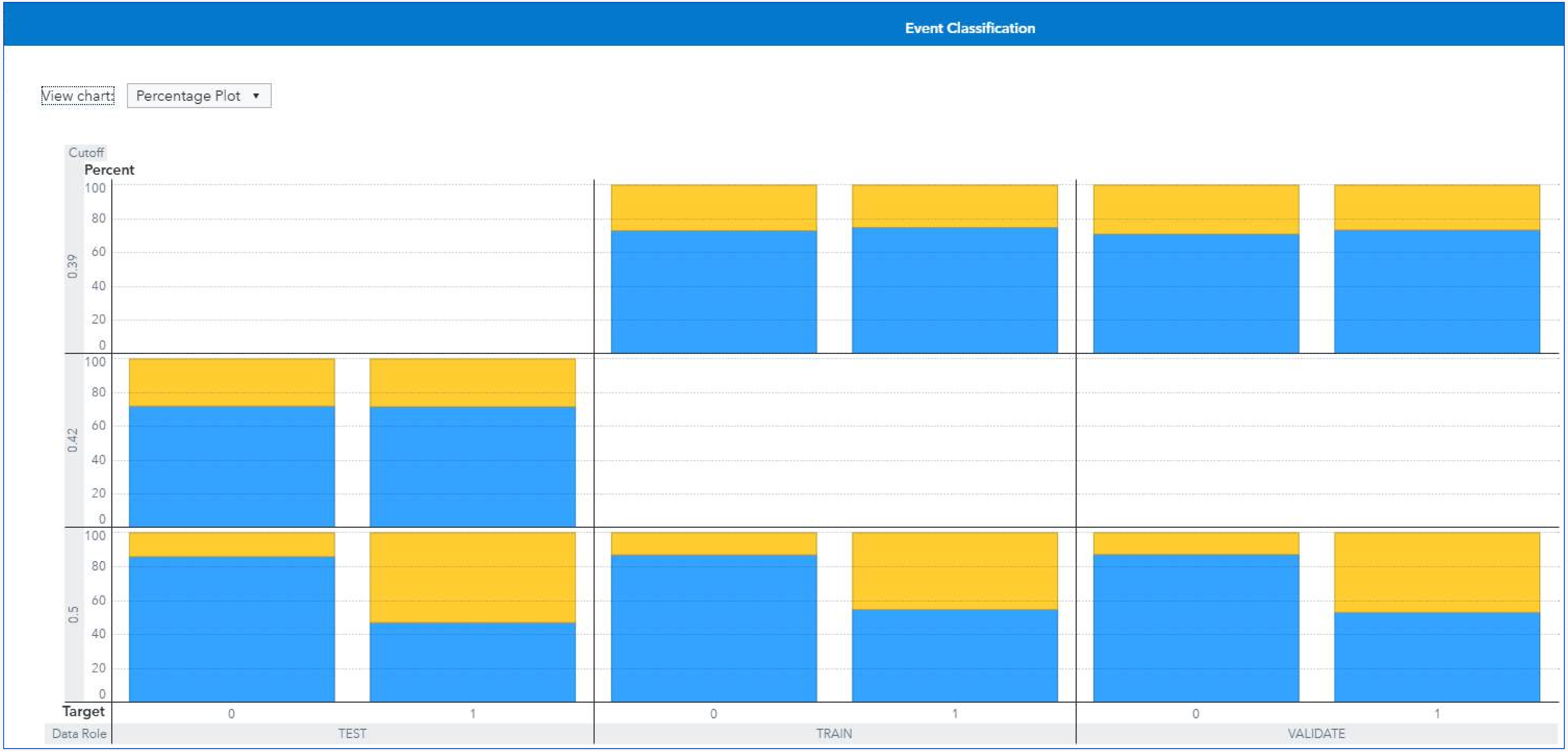
  Predicted **325** Match True

- Model exposed to 139 Match in Testing Set

  Predicted **101** Match True

GAUSSIAN PROCESSES CLASSIFICATION EVALUATION

After Adding XG, XGA



GAUSSIAN PROCESSES CLASSIFICATION EVALUATION

After Adding XG, XGA

Fit Statistics

Target ...	Data Role	Partitio...	Formatt...	Numbe...	Averag...	Divisor ...	Root Av...	Misclas...	Multi-Cl...	KS (You...	Area Un...
target	TEST	2	2	139	0.2009	139	0.4482	0.2878	0.5888	0.4379	0.7352
target	TRAIN	1	1	833	0.1735	833	0.4165	0.2509	0.5244	0.4789	0.8141
target	VALIDATE	0	0	417	0.1908	417	0.4368	0.2566	0.5655	0.4446	0.7696

Gini Co...	Gamma	Tau	KS Cutoff	KS at U...	Misclas...	Misclass...
0.4704	0.4777	0.2235	0.4200	0.3322	0.2806	0.2878
0.6281	0.6353	0.2958	0.3900	0.4199	0.2629	0.2509
0.5391	0.5457	0.2544	0.3900	0.4042	0.2806	0.2566

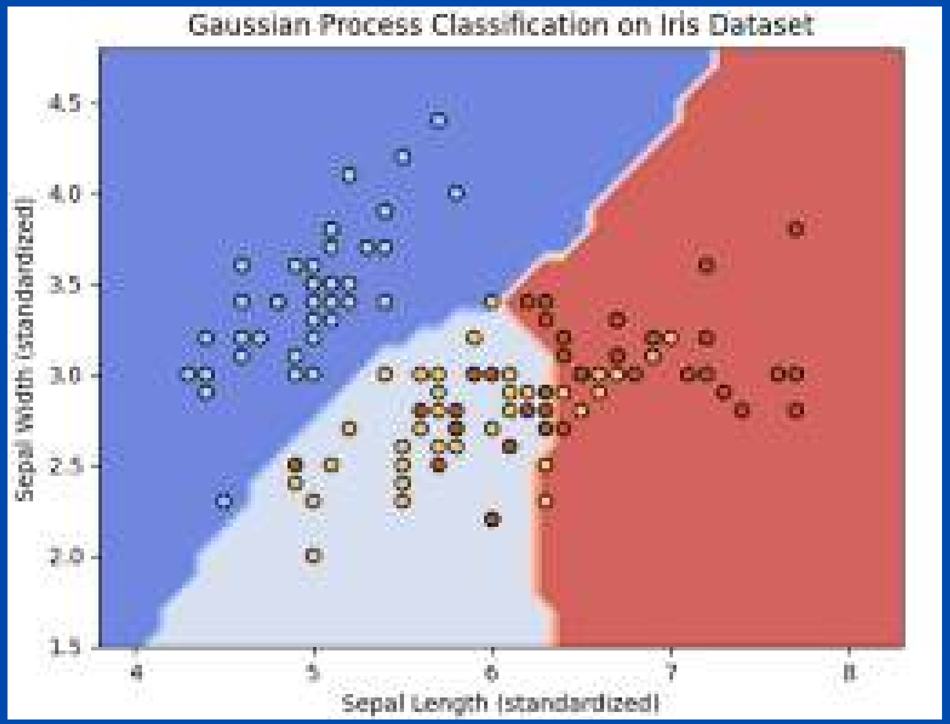
GAUSSIAN PROCESSES CLASSIFICATION EVALUATION

After Adding XG, XGA

Event Classification

View chart: Table

Cutoff	Cutoff Source	Target Name	Response	Event	Value	Training Frequ...	Validation Freq...	Test Frequency	Training Percen...	Validation Perc...	Test Percentage
0.3900	KS	target	CORRECT	1	True Positive	236	116	.	74.9206	73.4177	.
0.3900	KS	target	INCORRECT	1	False Negative	79	42	.	25.0794	26.5823	.
0.3900	KS	target	CORRECT	0	True Negative	378	184	.	72.9730	71.0425	.
0.3900	KS	target	INCORRECT	0	False Positive	140	75	.	27.0270	28.9575	.
0.4200	KS	target	CORRECT	1	True Positive	.	.	38	.	.	71.6981
0.4200	KS	target	INCORRECT	1	False Negative	.	.	15	.	.	28.3019
0.4200	KS	target	CORRECT	0	True Negative	.	.	62	.	.	72.0930
0.4200	KS	target	INCORRECT	0	False Positive	.	.	24	.	.	27.9070
0.5000	Default	target	CORRECT	1	True Positive	173	84	25	54.9206	53.1646	47.1698
0.5000	Default	target	INCORRECT	1	False Negative	142	74	28	45.0794	46.8354	52.8302
0.5000	Default	target	CORRECT	0	True Negative	451	226	74	87.0656	87.2587	86.0465
0.5000	Default	target	INCORRECT	0	False Positive	67	33	12	12.9344	12.7413	13.9535



USING GAUSSIAN PROCESSES CLASSIFICATION MODEL ON UNSEEN DATA

After Adding XG, XGA

Model exposed to 417 Match in Vaildation Set

Predicted **316** Match True

Model exposed to 139 Match in Testing Set

Predicted **101** Match True

Impact of XG, XGA on Improving Models Performance

BEFORE Using XG, XGA	AFTER USING XG, XGA
Random Forest Model	**Random Forest Model**
246 True Match from 417 Match (Vaildation Set) 74 True Match from 139 Match (Testing Set)	325 True Match from 417 Match (Vaildation Set) 101 True Match from 139 Match (Testing Set)
---	Expecteing 77 True MatcheMore + (Vaildation Set) Expecteing 27 True Match More + (Testing Set)
Gaussian Processes Classification	**Gaussian Processes Classification**
255 True Match from 417 Match (Vaildation Set) 85 True Match from 139 Match (Testing Set)	316 True Match from 417 Match (Vaildation Set) 101 True Match from 139 Match (Testing Set)
--	Expecteing 61 True Match More + (Vaildation Set) Expecteing 16 True Match More + (Testing Set)

THANKS FOR YOUR TIME