colab.research.google.com

ML_MiniProject_60009210185.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

```
[173] import numpy as np
      import pandas as pd
      import seaborn as sns
      import matplotlib.pyplot as plt
```

Reading the Dataset

```
[174] df= pd.read_csv("/content/ml_dataset.csv")
```

```
[26] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6080 entries, 0 to 6079
Data columns (total 21 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  6080 non-null   int64
 1   HomeTeam    6080 non-null   object
 2   AwayTeam    6080 non-null   object
 3   FTHG        6080 non-null   int64
 4   FTAG        6080 non-null   int64
 5   FTR         6080 non-null   object
 6   HTHG        6080 non-null   int64
 7   HTAG        6080 non-null   int64
 8   HTR         6080 non-null   object
 9   HS          6080 non-null   int64
 10  AS          6080 non-null   int64
 11  HST         6080 non-null   int64
 12  AST         6080 non-null   int64
 13  HF          6080 non-null   int64
 14  AF          6080 non-null   int64
 15  HC          6080 non-null   int64
 16  AC          6080 non-null   int64
 17  HY          6080 non-null   int64
 18  AY          6080 non-null   int64
 19  HR          6080 non-null   int64
 20  AR          6080 non-null   int64
dtypes: int64(17), object(4)
memory usage: 997.6+ KB
```

```
[175] df.describe()
```

	Unnamed: 0	FTHG	FTAG	HTHG	HTAG	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	H
count	6080.00000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.00000
mean	3039.50000	1.544737	1.065789	0.686842	0.476316	12.792105	9.378947	6.163158	4.531579	12.978947	13.944737	6.289474	4.647368	1.347368	1.794737	0.08421
std	1755.28915	1.283850	1.027799	0.842582	0.693612	4.853118	3.604910	2.987889	2.236617	4.159616	4.377285	2.997683	2.739197	1.258912	1.307968	0.28704
min	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	2.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	1519.75000	1.000000	0.000000	0.000000	0.000000	9.000000	7.000000	4.000000	3.000000	10.000000	11.000000	4.000000	3.000000	0.000000	1.000000	0.00000
50%	3039.50000	1.000000	1.000000	0.000000	0.000000	12.000000	9.000000	6.000000	4.000000	13.000000	13.000000	6.000000	4.000000	1.000000	2.000000	0.00000
75%	4559.25000	2.000000	2.000000	1.000000	1.000000	16.000000	12.000000	8.000000	6.000000	16.000000	17.000000	8.000000	6.000000	2.000000	3.000000	0.00000
max	6079.00000	6.000000	4.000000	5.000000	4.000000	33.000000	24.000000	19.000000	14.000000	25.000000	28.000000	17.000000	14.000000	7.000000	7.000000	2.00000

0s completed at 23:51

colab.research.google.com

ML_MiniProject_60009210185.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

[175] df.describe()

	Unnamed: 0	FTHG	FTAG	HTHG	HTAG	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	H
count	6080.00000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.000000	6080.00000
mean	3039.50000	1.544737	1.065789	0.686842	0.476316	12.792105	9.378947	6.163158	4.531579	12.978947	13.944737	6.289474	4.647368	1.347368	1.794737	0.08421
std	1755.28915	1.283850	1.027799	0.842582	0.693612	4.853118	3.604910	2.987889	2.236617	4.159616	4.377285	2.997683	2.739197	1.258912	1.307968	0.28704
min	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	2.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	1519.75000	1.000000	0.000000	0.000000	0.000000	9.000000	7.000000	4.000000	3.000000	10.000000	11.000000	4.000000	3.000000	0.000000	1.000000	0.00000
50%	3039.50000	1.000000	1.000000	0.000000	0.000000	12.000000	9.000000	6.000000	4.000000	13.000000	13.000000	6.000000	4.000000	1.000000	2.000000	0.00000
75%	4559.25000	2.000000	2.000000	1.000000	1.000000	16.000000	12.000000	8.000000	6.000000	16.000000	17.000000	8.000000	6.000000	2.000000	3.000000	0.00000
max	6079.00000	6.000000	4.000000	5.000000	4.000000	33.000000	24.000000	19.000000	14.000000	25.000000	28.000000	17.000000	14.000000	7.000000	7.000000	2.00000

ML_MiniProject_60009210185.ipynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

RAM Disk

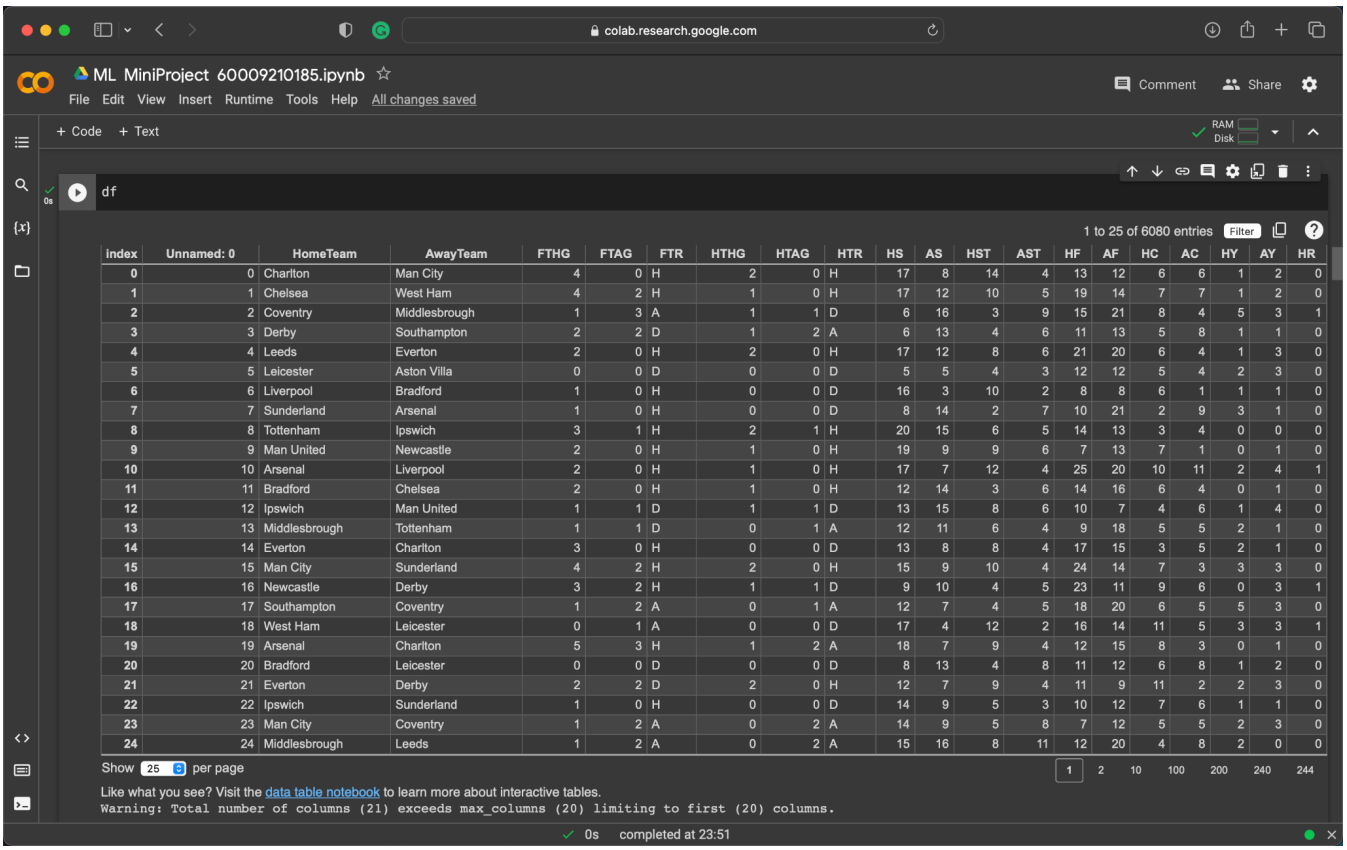
Searching for Null values

```
df_null = pd.DataFrame(df.isnull().sum(),columns=["null values"])
df_null
```

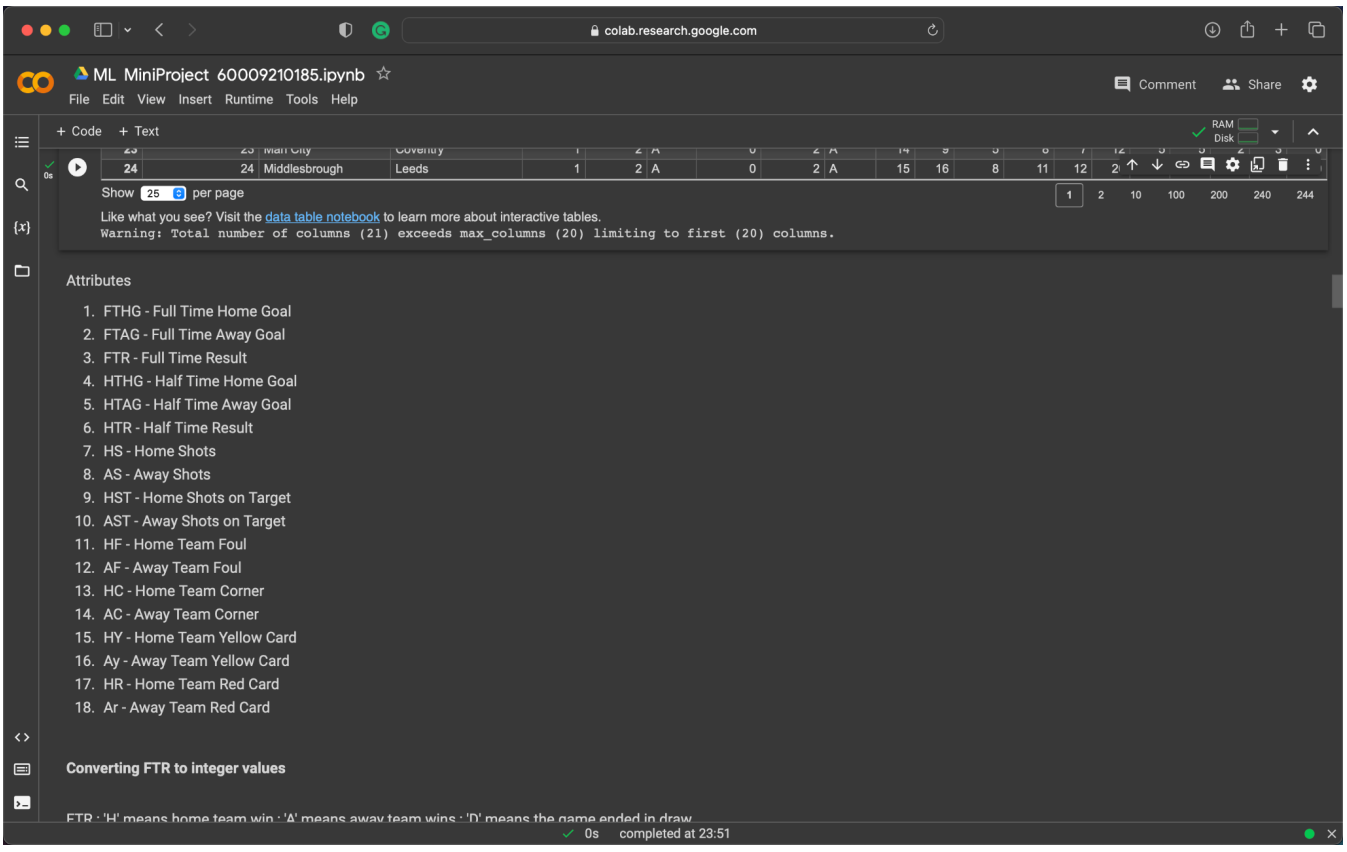
	null values
Unnamed: 0	0
HomeTeam	0
AwayTeam	0
FTHG	0
FTAG	0
FTR	0
HTHG	0
HTAG	0
HTR	0
HS	0
AS	0
HST	0
AST	0
HF	0
AF	0
HC	0
AC	0
HY	0
AY	0
HR	0
AR	0

Since, there are no null values in this dataset we need not remove entries to clean the dataset.

0s completed at 23:51



index	Unnamed: 0	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	HR
0	0	Charlton	Man City	4	0	H	2	0	H	17	8	14	4	13	12	6	6	1	2	0
1	1	Chelsea	West Ham	4	2	H	1	0	H	17	12	10	5	19	14	7	7	1	2	0
2	2	Coventry	Middlesbrough	1	3	A	1	1	D	6	16	3	9	15	21	8	4	5	3	1
3	3	Derby	Southampton	2	2	D	1	2	A	6	13	4	6	11	13	5	8	1	1	0
4	4	Leeds	Everton	2	0	H	2	0	H	17	12	8	6	21	20	6	4	1	3	0
5	5	Leicester	Aston Villa	0	0	D	0	0	D	5	5	4	3	12	12	5	4	2	3	0
6	6	Liverpool	Bradford	1	0	H	0	0	D	16	3	10	2	8	8	6	1	1	1	0
7	7	Sunderland	Arsenal	1	0	H	0	0	D	8	14	2	7	10	21	2	9	3	1	0
8	8	Tottenham	Ipswich	3	1	H	2	1	H	20	15	6	5	14	13	3	4	0	0	0
9	9	Man United	Newcastle	2	0	H	1	0	H	19	9	9	6	7	13	7	1	0	1	0
10	10	Arsenal	Liverpool	2	0	H	1	0	H	17	7	12	4	25	20	10	11	2	4	1
11	11	Bradford	Chelsea	2	0	H	1	0	H	12	14	3	6	14	16	6	4	0	1	0
12	12	Ipswich	Man United	1	1	D	1	1	D	13	15	8	6	10	7	4	6	1	4	0
13	13	Middlesbrough	Tottenham	1	1	D	0	1	A	12	11	6	4	9	18	5	5	2	1	0
14	14	Everton	Charlton	3	0	H	0	0	D	13	8	8	4	17	15	3	5	2	1	0
15	15	Man City	Sunderland	4	2	H	2	0	H	15	9	10	4	24	14	7	3	3	3	0
16	16	Newcastle	Derby	3	2	H	1	1	D	9	10	4	5	23	11	9	6	0	3	1
17	17	Southampton	Coventry	1	2	A	0	1	A	12	7	4	5	18	20	6	5	5	3	0
18	18	West Ham	Leicester	0	1	A	0	0	D	17	4	12	2	16	14	11	5	3	3	1
19	19	Arsenal	Charlton	5	3	H	1	2	A	18	7	9	4	12	15	8	3	0	1	0
20	20	Bradford	Leicester	0	0	D	0	0	D	8	13	4	8	11	12	6	8	1	2	0
21	21	Everton	Derby	2	2	D	2	0	H	12	7	9	4	11	9	11	2	2	3	0
22	22	Ipswich	Sunderland	1	0	H	0	0	D	14	9	5	3	10	12	7	6	1	1	0
23	23	Man City	Coventry	1	2	A	0	2	A	14	9	5	8	7	12	5	5	2	3	0
24	24	Middlesbrough	Leeds	1	2	A	0	2	A	15	16	8	11	12	20	4	8	2	0	0



16. Ay - Away Team Yellow Card
17. HR - Home Team Red Card
18. Ar - Away Team Red Card

Converting FTR to integer values

FTR : 'H' means home team win ; 'A' means away team wins ; 'D' means the game ended in draw

```
df = df.replace({'H':1,'A':-1,'D':0})
df
```

	Unnamed: 0	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	HS	...	HST	AST	HF	AF	HC	AC	HY	AY	HR	AR
0	0	Charlton	Man City	4	0	1	2	0	1	17	...	14	4	13	12	6	6	1	2	0	0
1	1	Chelsea	West Ham	4	2	1	1	0	1	17	...	10	5	19	14	7	7	1	2	0	0
2	2	Coventry	Middlesbrough	1	3	-1	1	1	0	6	...	3	9	15	21	8	4	5	3	1	0
3	3	Derby	Southampton	2	2	0	1	2	-1	6	...	4	6	11	13	5	8	1	1	0	0
4	4	Leeds	Everton	2	0	1	2	0	1	17	...	8	6	21	20	6	4	1	3	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6075	6075	Man City	Chelsea	1	2	-1	1	1	0	3	...	1	3	22	18	8	7	4	2	0	0
6076	6076	Middlesbrough	West Ham	2	1	1	2	1	1	19	...	7	5	13	15	5	6	0	0	0	0
6077	6077	Newcastle	Aston Villa	3	0	1	2	0	1	9	...	5	1	10	14	5	1	0	5	1	1
6078	6078	Southampton	Arsenal	3	2	1	0	1	-1	11	...	7	5	17	10	7	5	1	2	0	0
6079	6079	Tottenham	Man United	3	1	1	1	1	0	9	...	3	6	13	15	3	6	0	2	0	0

6080 rows × 21 columns

colab.research.google.com

ML MiniProject 60009210185.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

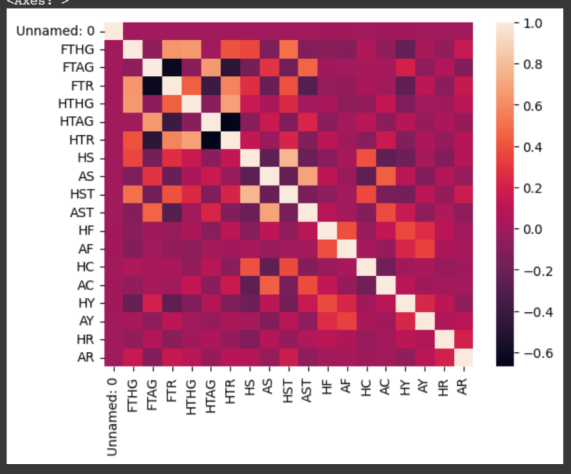
+ Code + Text

RAM Disk

Now we find the correlation between the FULL TIME RESULT (FTR) and other attributes

```
corr_matrix = df.corr()
sns.heatmap(corr_matrix)
```

```
<ipython-input-30-8e2c7b3d5a34>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to
  corr_matrix = df.corr()
<Axes: >
```



From this Heatmap we can see that these attributes affect Full Time Result the least:

1. HS - Home shots
2. AS - Away shots
3. HST - Home shots on Target
4. AST - Away shots on target
5. HF - Home fouls
6. AF - Away fouls
7. HC - Home corners
8. AC - Away corners
9. HY - Home yellow cards
10. AY - Away yellow cards
11. HR - Home red cards
12. AR - Away red cards

We will remove unneccesary attributes from the dataset

High number of attributes in the dataset can prove to be harmful in several ways:

1. Overfitting - When there are too many features in a dataset, it is more likely that the model will learn noise or irrelevant information, rather than the actual patterns that are useful for prediction. This can lead to overfitting, where the model performs well on the training data but poorly on the test data.
2. Increased complexity - With a large number of features, the model becomes more complex and harder to interpret.
3. Computational costs - The more features there are, the longer it will take to train the model and make predictions

colab.research.google.com

ML MiniProject 60009210185.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

We will be selecting these following attributes as they are the only ones needed for Predicitng the match winner:

1. HomeTeam
2. AwayTeam
3. FTHG - Full time home goals
4. FTAG - Full time away goals
5. HTHG - Half time away goals
6. HTAG - Half time home goals
7. FTR - Full time result

```
updated_df = df.loc[:,['HomeTeam','AwayTeam','FTHG','FTAG','HTHG','HTAG','FTR']]
updated_df
```

	HomeTeam	AwayTeam	FTHG	FTAG	HTHG	HTAG	FTR
0	Charlton	Man City	4	0	2	0	1
1	Chelsea	West Ham	4	2	1	0	1
2	Coventry	Middlesbrough	1	3	1	1	-1
3	Derby	Southampton	2	2	1	2	0
4	Leeds	Everton	2	0	2	0	1
...	...	...	...	...	...	...	...
6075	Man City	Chelsea	1	2	1	1	-1
6076	Middlesbrough	West Ham	2	1	2	1	1
6077	Newcastle	Aston Villa	3	0	2	0	1
6078	Southampton	Arsenal	3	2	0	1	1
6079	Tottenham	Man United	3	1	1	1	1

6080 rows × 7 columns

0s completed at 23:51

colab.research.google.com

ML MiniProject 60009210185.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

We calculate the strength of each team

```
[81] #we create a new dataframe containing sum of all goals scored (home & away) and their average
     table = pd.DataFrame(columns=('HGS','AGS','HAS','AAS','HGC','AGC','HDS','ADS','Team'))
```

1. HGS - Home goals scored
2. AGS - Away goals scored
3. HAS - Home attacking strength
4. AAS - Away attacking strength
5. HGC - Home goals conceded
6. AGC - Away goals conceded
7. HDS - Home defensive strength
8. ADS - Away defensive strength

```
[90] res_home = df.groupby('HomeTeam')
     res_away = df.groupby("AwayTeam")
```

```
     avg_home_scored = df.FTHG.sum() / 6080.0 #there are 6080 matches
     avg_away_scored = df.FTAG.sum() / 6080.0
     avg_home_conceded = avg_away_scored
     avg_away_conceded = avg_home_scored
```

colab.research.google.com

ML MiniProject 60009210185.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
table.HGS = res_home.FTHG.sum().values
table.HGC = res_home.FTAG.sum().values
table.AGS = res_away.FTAG.sum().values
table.AGC = res_away.FTHG.sum().values
#19 Home matches for each team each season and 16 seasons therefore 304 home matches and 304 away matches
table.HAS = (table.HGS / 304.0) / avg_home_scored
table.AAS = (table.AGS / 304.0) / avg_away_scored
table.HDS = (table.HGC / 304.0) / avg_home_conceded
table.ADS = (table.AGC / 304.0) / avg_away_conceded
table.Team = res_home.HomeTeam.all().values

table
```

colab.research.google.com

ML MiniProject 60009210185.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
table
```

	HGS	AGS	HAS	AAS	HGC	AGC	HDS	ADS	Team
0	720	288	1.533220	0.888889	208	400	0.641975	0.851789	Arsenal
1	432	304	0.919932	0.938272	320	368	0.987654	0.783646	Aston Villa
2	320	160	0.681431	0.493827	464	656	1.432099	1.396934	Bradford
3	496	304	1.056218	0.938272	304	608	0.938272	1.294719	Charlton
4	704	384	1.499148	1.185185	320	400	0.987654	0.851789	Chelsea
5	224	352	0.477002	1.086420	368	640	1.135802	1.362862	Coventry
6	368	224	0.783646	0.691358	384	560	1.185185	1.192504	Derby
7	464	256	0.988075	0.790123	432	512	1.333333	1.090290	Everton
8	496	416	1.056218	1.283951	240	432	0.740741	0.919932	Ipswich
9	576	448	1.226576	1.382716	336	352	1.037037	0.749574	Leeds
10	448	176	0.954003	0.543210	368	448	1.135802	0.954003	Leicester
11	640	496	1.362862	1.530864	224	400	0.691358	0.851789	Liverpool
12	320	336	0.681431	1.037037	496	544	1.530864	1.158433	Man City
13	784	480	1.669506	1.481481	192	304	0.592593	0.647359	Man United
14	288	416	0.613288	1.283951	368	336	1.135802	0.715503	Middlesbrough
15	416	288	0.885860	0.888889	272	528	0.839506	1.124361	Newcastle
16	432	208	0.919932	0.641975	352	416	1.086420	0.885860	Southampton
17	384	352	0.817717	1.086420	256	400	0.790123	0.851789	Sunderland
18	496	256	1.056218	0.790123	256	608	0.790123	1.294719	Tottenham
19	384	336	0.817717	1.037037	320	480	0.987654	1.022147	West Ham

0s completed at 23:51

colab.research.google.com

ML_MiniProject_60009210185.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

```
table.set_index('Team')
```

Team	HGS	AGS	HAS	AAS	HGC	AGC	HDS	ADS
Arsenal	720	288	1.533220	0.888889	208	400	0.641975	0.851789
Aston Villa	432	304	0.919932	0.938272	320	368	0.987654	0.783646
Bradford	320	160	0.681431	0.493827	464	656	1.432099	1.396934
Charlton	496	304	1.056218	0.938272	304	608	0.938272	1.294719
Chelsea	704	384	1.499148	1.185185	320	400	0.987654	0.851789
Coventry	224	352	0.477002	1.086420	368	640	1.135802	1.362862
Derby	368	224	0.783646	0.691358	384	560	1.185185	1.192504
Everton	464	256	0.988075	0.790123	432	512	1.333333	1.090290
Ipswich	496	416	1.056218	1.283951	240	432	0.740741	0.919932
Leeds	576	448	1.226576	1.382716	336	352	1.037037	0.749574
Leicester	448	176	0.954003	0.543210	368	448	1.135802	0.954003
Liverpool	640	496	1.362862	1.530864	224	400	0.691358	0.851789
Man City	320	336	0.681431	1.037037	496	544	1.530864	1.158433
Man United	784	480	1.669506	1.481481	192	304	0.592593	0.647359
Middlesbrough	288	416	0.613288	1.283951	368	336	1.135802	0.715503
Newcastle	416	288	0.885860	0.888889	272	528	0.839506	1.124361
Southampton	432	208	0.919932	0.641975	352	416	1.086420	0.885860
Sunderland	384	352	0.817717	1.086420	256	400	0.790123	0.851789
Tottenham	496	256	1.056218	0.790123	256	608	0.790123	1.294719
West Ham	384	336	0.817717	1.037037	320	480	0.987654	1.022147

Total Team Strength can be found by subtracting defensive strength from attacking strength and taking the average of Home Team Strength (HTS) and Away Team Strength (ATS)

```
[155] table['HTS']=table['HAS']-table['HDS']
      table['ATS']=table['AAS']-table['ADS']
      table['TS']=(table['HTS']+table['ATS'])/2
```

```
[117] best_home_team1 = table[['Team','HGS']]
      best_home_team1.set_index('Team')
      best_home_team1 = best_home_team1.sort_values(by='HGS', ascending =False)
```

0s completed at 23:51

colab.research.google.com

ML_MiniProject_60009210185.ip...

File Edit View Insert Runtime Tools Help All changes saved

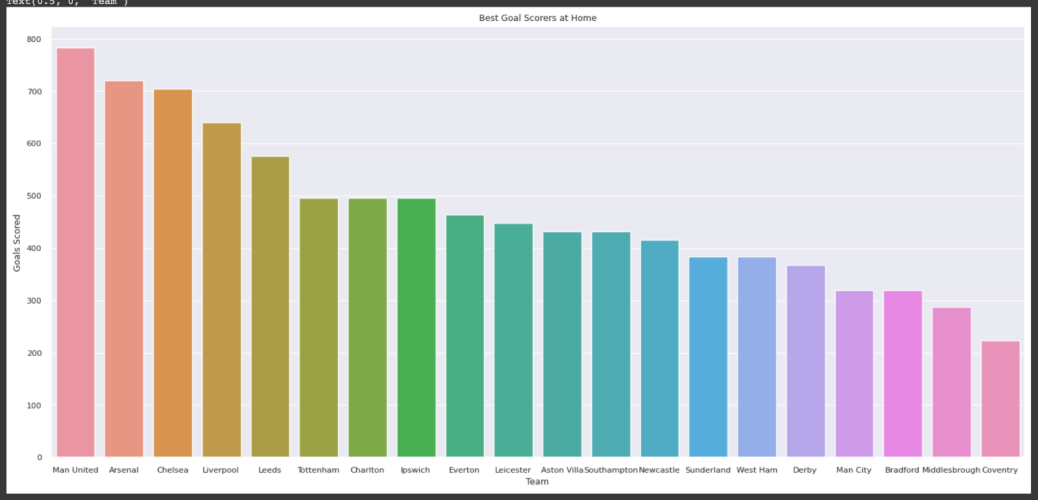
Comment Share

+ Code + Text

The best goalscoring teams at home represented through bar chart

```
plt.figure(figsize=(18,8))
sns.set(font_scale=0.75)
sns.barplot(x='Team', y='HGS', data=best_home_team1,)
plt.title("Best Goal Scorers at Home")
plt.ylabel("Goals Scored")
plt.xlabel("Team")
```

```
Text(0.5, 0, 'Team')
```



Observation

0s completed at 23:51

Observation

Man United and Arsenal are top scoring teams at home while Middlesbrough are in the bottom two at home.

colab.research.google.com

ML MiniProiect 60009210185.ip...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

[120] best_away_team1 = best_away_team1.sort_values(by='AGS', ascending=False)

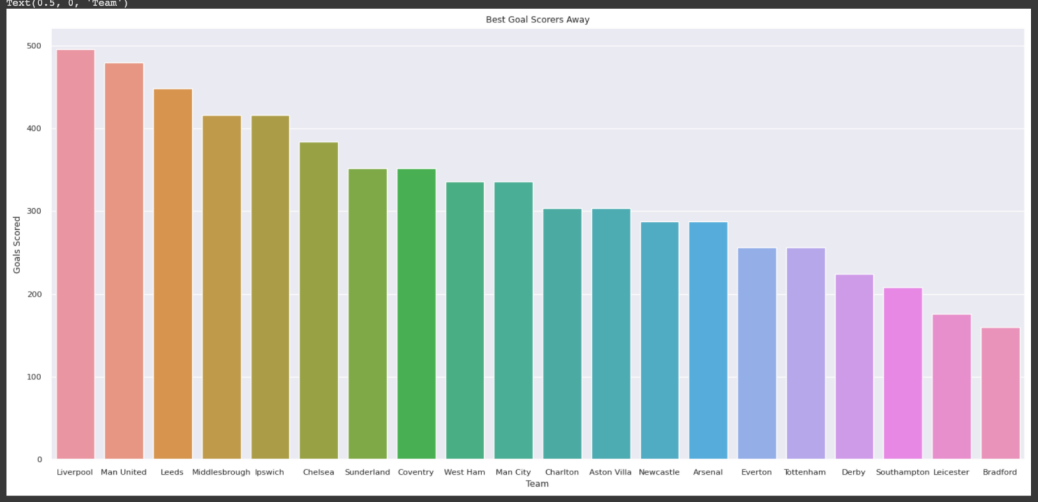
0s

The best goalscoring teams away from home represented through bar chart

```
plt.figure(figsize=(18,8))
sns.set(font_scale=0.75)
sns.barplot(x='Team', y='AGS', data=best_away_team1,)
plt.title("Best Goal Scorers Away")
plt.ylabel("Goals Scored")
plt.xlabel("Team")
```

1s

```
Text(0.5, 0, 'Team')
```



0s completed at 23:51

colab.research.google.com

ML MiniProject 60009210185.ip...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Observation

Liverpool and Man United at the top with Middlesbrough close behind while Arsenal are way behind compared to their form at home.

Conclusion

Man United and Liverpool have a strong goalscoring record at home and away while Arsenal seem to struggle on enemy grounds and Middlesbrough vice versa to the Arsenal's record.

```
[146] best_strength_home = table[['Team','HTS']]
      best_strength_home.set_index('Team')
      best_strength_home = best_strength_home.sort_values(by='HTS', ascending =False)
```

```
[149] best_strength_away = table[['Team','ATS']]
      best_strength_away.set_index('Team')
      best_strength_away = best_strength_away.sort_values(by='ATS', ascending =False)
```

colab.research.google.com

ML MiniProject 60009210185.ip...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

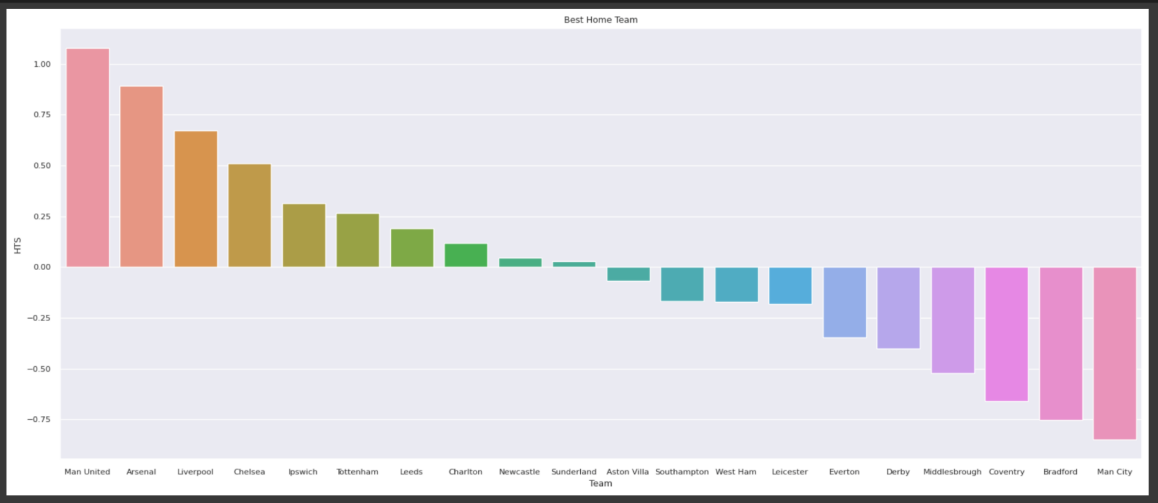
+ Code + Text

RAM Disk

```
[149] best_strength_away.set_index('Team')
      best_strength_away = best_strength_away.sort_values(by='ATS', ascending =False)
```

Home Team Strength

```
plt.figure(figsize=(20,8))
sns.set(font_scale=0.75)
plt.title("Best Home Team")
plt.ylabel("Home Team Strength")
plt.xlabel("Team")
sns.barplot(x='Team', y='HTS', data=best_strength_home)
plt.show()
```



0s completed at 23:51

colab.research.google.com

ML MiniProject 60009210185.ip...

File Edit View Insert Runtime Tools Help All changes saved

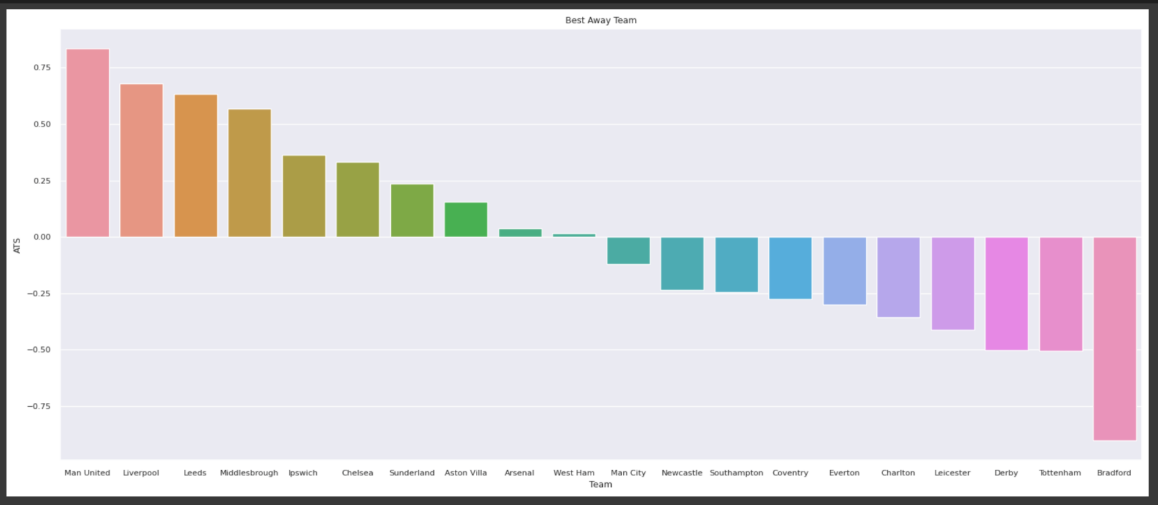
Comment Share

+ Code + Text

RAM Disk

Away Team Strength

```
plt.figure(figsize=(20,8))
sns.set(font_scale=0.75,style='darkgrid', rc={'axes.facecolor':'#EAEAF2'})
plt.title("Best Away Team")
plt.ylabel("Away Team Strength")
plt.xlabel("Team")
sns.barplot(x='Team', y='ATS', data=best_strength_away)
plt.show()
```




```
[156] new_df = table[['Team','HTS','ATS']]
```

0s completed at 23:51

colab.research.google.com

ML MiniProject 60009210185.ip...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

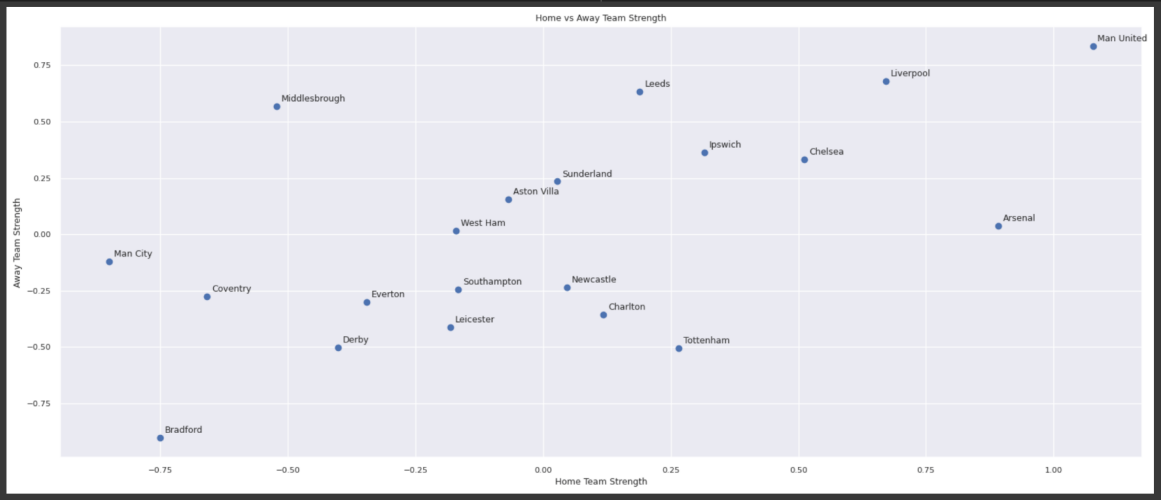
+ Code + Text

RAM Disk

Comparing the home strength and away strength

```
plt.figure(figsize=(20,8))
plt.scatter(x='HTS', y='ATS', data=new_df)
plt.title("Home vs Away Team Strength")
plt.ylabel("Away Team Strength")
plt.xlabel("Home Team Strength")
for i, row in new_df.iterrows():
    plt.annotate(row['Team'], xy=(row['HTS'], row['ATS']), xytext=(5, 5), textcoords='offset points')

plt.show()
```



0s completed at 23:51

Observation

We notice that Middlesbrough and Man City have an incredible record away from if compared to their strength at home while Arsenal fall behind their rivals Liverpool and Man United when it comes to getting a result on enemy grounds