In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
train = pd.read_csv("C:/Users/khali/Desktop/Kaggle_prjt/train.csv")
test = pd.read_csv("C:/Users/khali/Desktop/Kaggle_prjt/test.csv")
train_target_scores = pd.read_csv('C:/Users/khali/Desktop/Kaggle_prjt/train_target_and_scores.csv')
```

```
C:\Users\khali\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3444: DtypeWarning: Columns (7) hav
e mixed types.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

In [3]:
```python
train_df=train.copy()
```

In [4]:
```python
train_df.shape
```

Out[4]:
```
(110938, 190)
```

In [5]:
```python
train_df.head()
```

Out[5]:

	id	target	home_team_name	away_team_name	match_date	league_name	league_id	is_cup	home_team_coach_id	away_team_c
0	11906497	away	Newell's Old Boys	River Plate	2019-12-01 00:45:00	Superliga	636	False	468196.0	4
1	11984383	home	Real Estelí	Deportivo Las Sabanas	2019-12-01 01:00:00	Primera Division	752	False	516788.0	221
2	11983301	draw	UPNFM	Marathón	2019-12-01 01:00:00	Liga Nacional	734	False	2510608.0	4
3	11983471	away	León	Morelia	2019-12-01 01:00:00	Liga MX	743	False	1552508.0	4
4	11883005	home	Cobán Imperial	Iztapa	2019-12-01 01:00:00	Liga Nacional	705	False	429958.0	4

5 rows × 190 columns

In [6]:
```python
train_df.columns
```

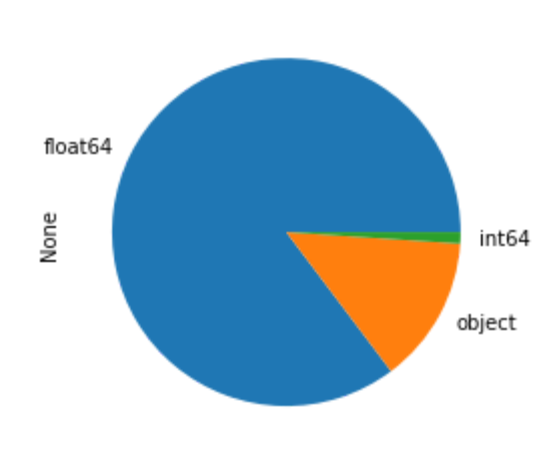
Out[6]:
```
Index(['id', 'target', 'home_team_name', 'away_team_name', 'match_date',
       'league_name', 'league_id', 'is_cup', 'home_team_coach_id',
       'away_team_coach_id',
       ...
       'away_team_history_league_id_1', 'away_team_history_league_id_2',
       'away_team_history_league_id_3', 'away_team_history_league_id_4',
       'away_team_history_league_id_5', 'away_team_history_league_id_6',
       'away_team_history_league_id_7', 'away_team_history_league_id_8',
       'away_team_history_league_id_9', 'away_team_history_league_id_10'],
      dtype='object', length=190)
```

In [7]:
```python
train_df.dtypes.value_counts()
```

Out[7]:
```
float64    162
object      26
int64        2
dtype: int64
```

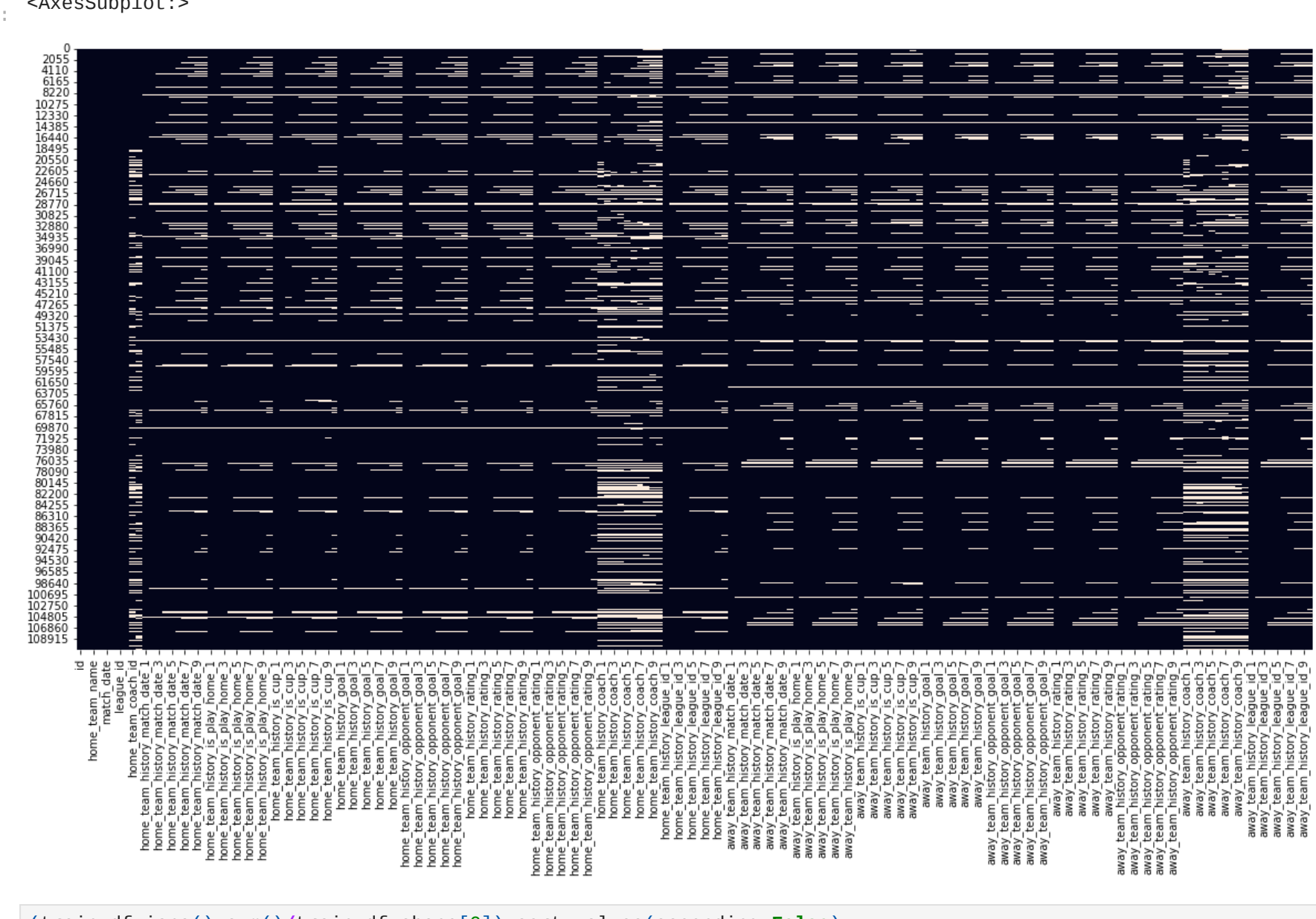
In [8]:
```python
train_df.dtypes.value_counts().plot.pie()
```

Out[8]:
```
<AxesSubplot:ylabel='None'>
```



In [9]:
```python
plt.figure(figsize=(20,10))
sns.heatmap(train_df.isna(), cbar=False)
```

Out[9]:
```
<AxesSubplot:>
```



In [10]:
```python
(train_df.isna().sum()/train_df.shape[0]).sort_values(ascending=False)
```

Out[10]:
```
away_team_history_coach_10    0.244542
home_team_history_coach_10    0.240251
away_team_history_coach_9     0.235717
home_team_history_coach_9     0.231958
away_team_history_coach_8     0.227325
                                ...
league_name                   0.000009
target                        0.000000
match_date                    0.000000
league_id                     0.000000
id                            0.000000
Length: 190, dtype: float64
```

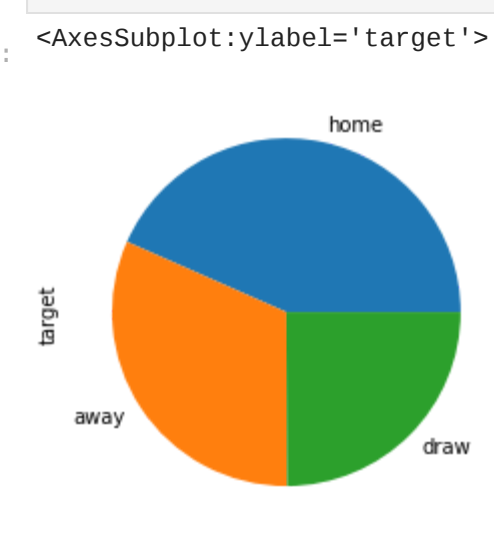
peu de valeurs manquantes la variables qui a le plus de valuers manquantes est 25 %

In [11]:
```python
train_df['target'].value_counts(normalize=True)
```

Out[11]:
```
home    0.433693
away    0.317060
draw    0.249247
Name: target, dtype: float64
```

In [12]:
```python
train_df['target'].value_counts(normalize=True).plot.pie()
```

Out[12]:
```
<AxesSubplot:ylabel='target'>
```



In [13]:
```python
train_df.isnull().sum()
```

Out[13]:
```
id                                 0
target                             0
home_team_name                     1
away_team_name                     1
match_date                         0
                               ...
away_team_history_league_id_6   8426
away_team_history_league_id_7   9867
away_team_history_league_id_8  11295
away_team_history_league_id_9  12762
away_team_history_league_id_10 14216
Length: 190, dtype: int64
```

In [14]:
```python
#tous les lignes des 2 colonees
a = train_df.loc[:, ['home_team_name', 'league_id']]
#nunique(): Compter le nombre d'éléments distincts dans l'axe spécifié.
print("Le nombre d'equipe domicile est :" ,a.home_team_name.nunique())
```

```
Le nombre d'equipe domicile est : 9813
```

In [15]:
```python
ext = train_df.loc[:, ['away_team_name', 'league_id']]
print("Le nombre d'equipe exterieur est :" ,ext.away_team_name.nunique())
```

```
Le nombre d'equipe exterieur est : 9892
```

In [16]:
```python
teams_No_leagues = a.groupby('home_team_name')['league_id'].count()
teams_No_leagues.sort_values(ascending=False)
```

Out[16]:
```
home_team_name
Al Ittihad         91
River Plate        71
Rangers            64
Liverpool          63
Al Ahli            62
                   ..
Peñarol W           1
Ugyen Academy       1
Eibar U19           1
Eichede             1
Derby County U19    1
Name: league_id, Length: 9813, dtype: int64
```

est-il possible qu'une equipe joue dans 9 league differentes??

In [17]:
```python
train_df = pd.concat([train, test], axis=0)
b= train_df[['league_id', 'league_name']]
print(b.groupby('league_id')['league_name'].nunique().sort_values(ascending=False))
```

```
league_id
199     2
1763    2
334     2
749     2
1695    2
       ..
1128    1
1146    1
1147    1
1148    1
2439    1
Name: league_name, Length: 1000, dtype: int64
```

On remarque qu'il y a des id qui sont affecte a 2 differet league name

In [18]:
```python
b.groupby('league_id')['league_name'].nunique().value_counts()
```

Out[18]:
```
1    976
2     24
Name: league_name, dtype: int64
```

On remarque que 24 league id ont 2 league name different

In [19]:
```python
print('---------------------------------')
print(b.groupby('league_name')['league_id'].nunique().sort_values(ascending=False))
```

```
---------------------------------
league_name
Premier League      35
Super Cup           28
Primera Division     9
Women's Cup          8
Super League         7
                    ..
Esiliiga B           1
Estonian Cup         1
Ettan: North         1
Ettan: South         1
Yokary Liga          1
Name: league_id, Length: 851, dtype: int64
```

In [20]:
```python
b.groupby('league_name')['league_id'].nunique().value_counts()
```

Out[20]:
```
1     800
2      25
3       9
4       6
6       4
7       2
5       1
35      1
9       1
28      1
8       1
Name: league_id, dtype: int64
```

On remarque que certains league name ont plus que 35 id c'est pour cette raison qu'on doit negliger league id et utiliser que la colonne league_name

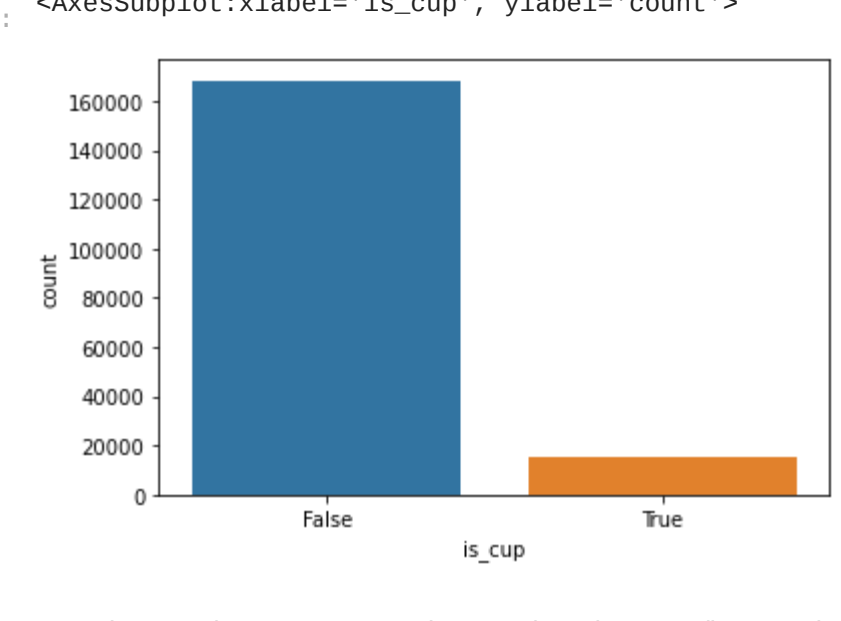
In [21]:
```python
(train_df.is_cup).value_counts(normalize=True)
```

Out[21]:
```
False    0.91622
True     0.08378
Name: is_cup, dtype: float64
```

In [22]:
```python
sns.countplot(train_df.is_cup)
```

```
C:\Users\khali\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variabl
e as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[22]:
```
<AxesSubplot:xlabel='is_cup', ylabel='count'>
```



91% des matches ne sont pas des matches de coupe(league,champions league.....)

In [23]:
```python
pd.crosstab(train_df['league_name'], train_df['target'])
```

Out[23]:

target	away	draw	home
league_name			
1. Deild	30	16	43
1. Deild Women	23	16	29
1. Division	91	78	147
1. Division Women	10	3	13
1. HNL	81	59	112
...	...	...	...
Wsl 2 Women	42	27	48
Ybc Levain Cup	29	13	18
Ykkonen	48	29	55
Youth League	148	94	225
Yokary Liga	25	10	33

727 rows × 3 columns

In [24]:
```python
pd.crosstab(train_df['match_date'], train_df['target'])
```

Out[24]:

target	away	draw	home
match_date			
2019-12-01 00:45:00	1	0	0
2019-12-01 01:00:00	2	2	3
2019-12-01 02:00:00	0	0	1
2019-12-01 03:00:00	0	0	1
2019-12-01 03:05:00	1	0	0
...	...	...	...
2021-04-30 21:30:00	0	0	1
2021-04-30 22:00:00	5	3	5
2021-04-30 22:30:00	0	1	0
2021-04-30 23:00:00	4	3	2
2021-05-01 00:00:00	1	0	1

20269 rows × 3 columns

In [25]:
```python
pd.crosstab(train_df['home_team_history_goal_1'], train_df['target'])
```

Out[25]:

target	away	draw	home
home_team_history_goal_1			
0.0	11112	8491	13076
1.0	11853	9275	15701
2.0	6905	5569	9969
3.0	2956	2550	5097
4.0	1177	987	2218
5.0	430	330	883
6.0	175	141	371
7.0	61	47	171
8.0	27	18	67
9.0	14	11	41
10.0	5	0	15
11.0	2	5	10
12.0	1	0	9
13.0	0	1	3
14.0	1	1	1
15.0	0	0	1
18.0	0	0	1

In [26]:
```python
pd.crosstab(train_df['is_cup'], train_df['target'])
```

Out[26]:

target	away	draw	home
is_cup			
False	31931	25707	44218
True	3242	1944	3895

In [27]:
```python
pd.crosstab(train_df['is_cup'], train_df['home_team_name'])
```

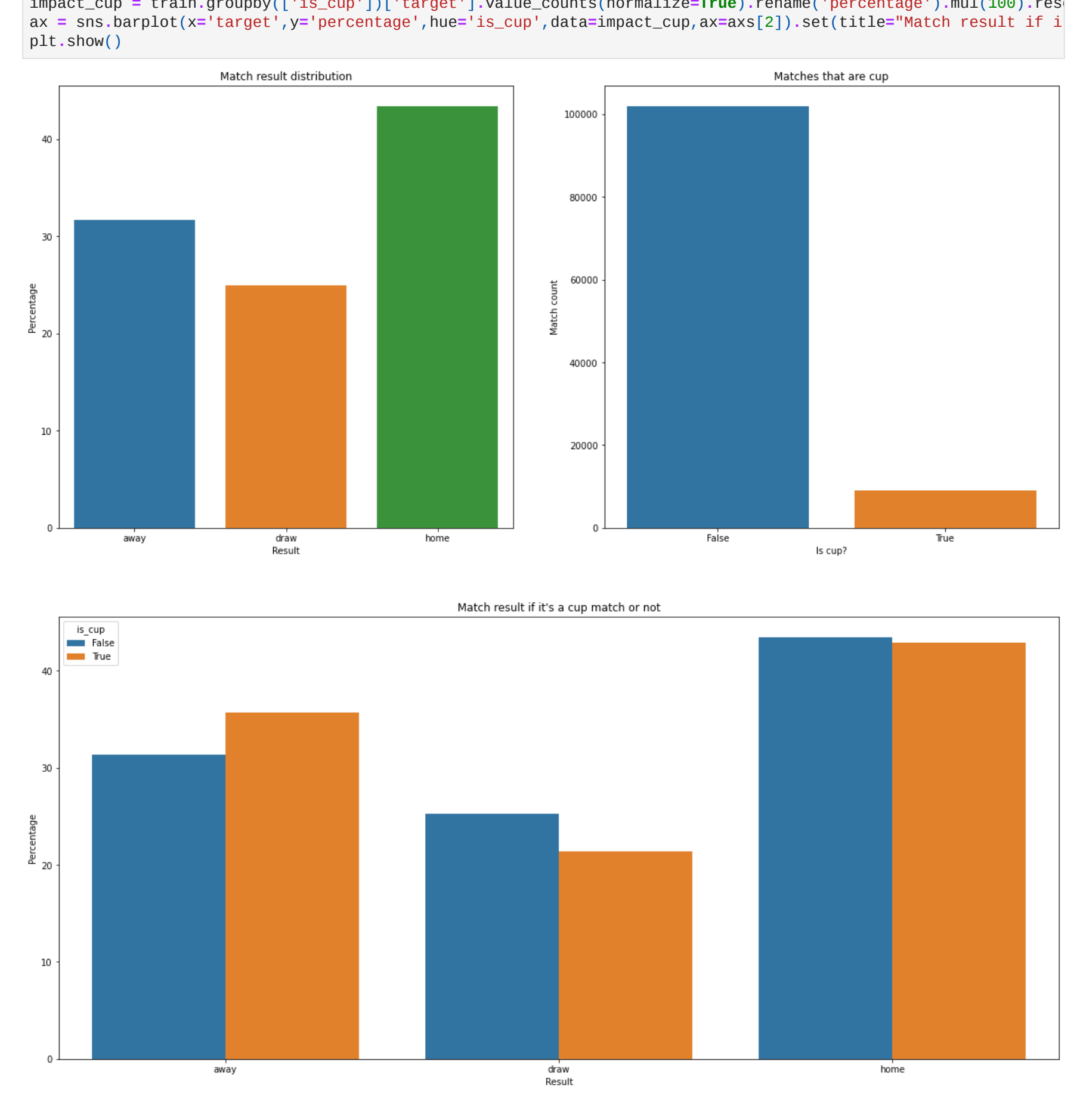
Out[27]:

home_team_name	FC Köln	07 Vestur	1. FC Köln	1. FC M'gladbach	1. FC Merseburg	1. FC Union Berlin	1. FSV Mainz 05	1. Maj Ruma	1. SC Feucht	12 Horas	...	Žarkovo	Žarnovica	Ždírec nad Doubravou	Ž
is_cup															
False	2	17	4	7	9	5	5	6	7	0	...	30	10	7	
True	0	1	0	0	0	1	0	0	0	3	...	0	0	0	

2 rows × 11480 columns

In [28]:
```python
import matplotlib.gridspec as gridspec
plt.figure(figsize=(20,20))
g = gridspec.GridSpec(2,2)
ax1= plt.subplot(g[0,0])
ax2= plt.subplot(g[0,1])
ax3= plt.subplot(g[1,:])
axs=[ax1,ax2,ax3]

# Count for each type of Result
cnt_target = train['target'].value_counts(normalize=True).rename('Percentage').mul(100).reset_index().sort_valu
sns.barplot(x='index',y='Percentage',data=cnt_target,ax=axs[0]).set(title="Match result distribution",ylabel='P
# Matches if is cup or not
sns.countplot(x='is_cup',data=train,ax=axs[1]).set(title="Matches that are cup",ylabel='Match count',xlabel='Is
# Result per home and away team
impact_cup = train.groupby(['is_cup'])['target'].value_counts(normalize=True).rename('percentage').mul(100).res
ax = sns.barplot(x='target',y='percentage',hue='is_cup',data=impact_cup,ax=axs[2]).set(title="Match result if i
plt.show()
```



Preprocessing

In [29]:
```python
train.drop(train.filter(regex='coach').columns, axis=1, inplace = True)
test.drop(test.filter(regex='coach').columns, axis=1, inplace = True)
```

In [30]:
```python
train = train.drop('league_id', axis=1)
test = test.drop('league_id', axis=1)
```

In [31]:
```python
for col in train.filter(regex='date', axis=1).columns:
    train[col] = pd.to_datetime(train[col])
    test[col] = pd.to_datetime(test[col])

# date based features
for i in range(1, 11):
    train[f'home_team_history_match_days_ago_{i}'] = (train['match_date'] - train[f'home_team_history_match_dat
    test[f'home_team_history_match_days_ago_{i}'] = (test['match_date'] - test[f'home_team_history_match_date_(
    train[f'away_team_history_match_days_ago_{i}'] = (train['match_date'] - train[f'away_team_history_match_dat
    test[f'away_team_history_match_days_ago_{i}'] = (test['match_date'] - test[f'away_team_history_match_date_(
```

In [32]:
```python
for i in range(1, 11):
    train.drop(f'home_team_history_match_date_{i}', axis=1 , inplace=True)
    train.drop(f'away_team_history_match_date_{i}', axis=1 , inplace=True)
    test.drop(f'home_team_history_match_date_{i}', axis=1 , inplace=True)
    test.drop(f'away_team_history_match_date_{i}', axis=1 , inplace=True)
```

In [33]:
```python
train.head()
```

Out[33]:

	id	target	home_team_name	away_team_name	match_date	league_name	is_cup	home_team_history_is_play_home_1	home_tear
0	11906497	away	Newell's Old Boys	River Plate	2019-12-01 00:45:00	Superliga	False	0.0	
1	11984383	home	Real Estelí	Deportivo Las Sabanas	2019-12-01 01:00:00	Primera Division	False	1.0	
2	11983301	draw	UPNFM	Marathón	2019-12-01 01:00:00	Liga Nacional	False	0.0	
3	11983471	away	León	Morelia	2019-12-01 01:00:00	Liga MX	False	0.0	
4	11883005	home	Cobán Imperial	Iztapa	2019-12-01 01:00:00	Liga Nacional	False	0.0	

5 rows × 167 columns

In [34]:
```python
test.head()
```

Out[34]:

	id	home_team_name	away_team_name	match_date	league_name	is_cup	home_team_history_is_play_home_1	home_team_histor
0	17761448	12 de Octubre	Sportivo Luqueño	2021-05-01 00:15:00	Division 1	False	0.0	
1	17695487	Necaxa	Atlas	2021-05-01 00:30:00	Liga MX	False	0.0	
2	17715496	Sertãozinho	EC São Bernardo	2021-05-01 01:00:00	Paulista A2	False	1.0	
3	17715493	RB Brasil	XV de Piracicaba	2021-05-01 01:00:00	Paulista A2	False	0.0	
4	17715492	Taubaté	Monte Azul	2021-05-01 01:00:00	Paulista A2	False	0.0	

5 rows × 166 columns

In [35]:
```python
train = train.drop('match_date', axis=1)
test = test.drop('match_date', axis=1)
```

In [36]:
```python
all_data_na = (train.isnull().sum() / len(train)) * 100
all_data_na = all_data_na.drop(all_data_na[all_data_na == 0].index).sort_values(ascending=False)[:30]
missing_data = pd.DataFrame({'Missing Ratio' :all_data_na})
print(len(all_data_na))
missing_data.head(30)
```

```
30
```

Out[36]:

	Missing Ratio
away_team_history_is_cup_10	13.357912
away_team_history_opponent_rating_10	12.820675
away_team_history_rating_10	12.820675
away_team_history_match_days_ago_10	12.814365
away_team_history_league_id_10	12.814365
away_team_history_is_play_home_10	12.814365
away_team_history_opponent_goal_10	12.814365
away_team_history_goal_10	12.814365
home_team_history_is_cup_10	12.787323
home_team_history_rating_10	12.228452
home_team_history_opponent_rating_10	12.228452
home_team_history_opponent_goal_10	12.222142
home_team_history_league_id_10	12.222142
home_team_history_goal_10	12.222142
home_team_history_is_play_home_10	12.222142
home_team_history_match_days_ago_10	12.222142
away_team_history_is_cup_9	11.993185
away_team_history_rating_9	11.512737
away_team_history_opponent_rating_9	11.512737
away_team_history_goal_9	11.503723
away_team_history_league_id_9	11.503723
away_team_history_match_days_ago_9	11.503723
away_team_history_opponent_goal_9	11.503723
away_team_history_is_play_home_9	11.503723
home_team_history_is_cup_9	11.474878
home_team_history_opponent_rating_9	10.943951
home_team_history_rating_9	10.943951
home_team_history_is_play_home_9	10.939444
home_team_history_goal_9	10.939444
home_team_history_opponent_goal_9	10.939444

In [37]:
```python
all_data_na = (test.isnull().sum() / len(test)) * 100
all_data_na = all_data_na.drop(all_data_na[all_data_na == 0].index).sort_values(ascending=False)[:30]
missing_data = pd.DataFrame({'Missing Ratio' :all_data_na})
print(len(all_data_na))
missing_data.head(30)
```

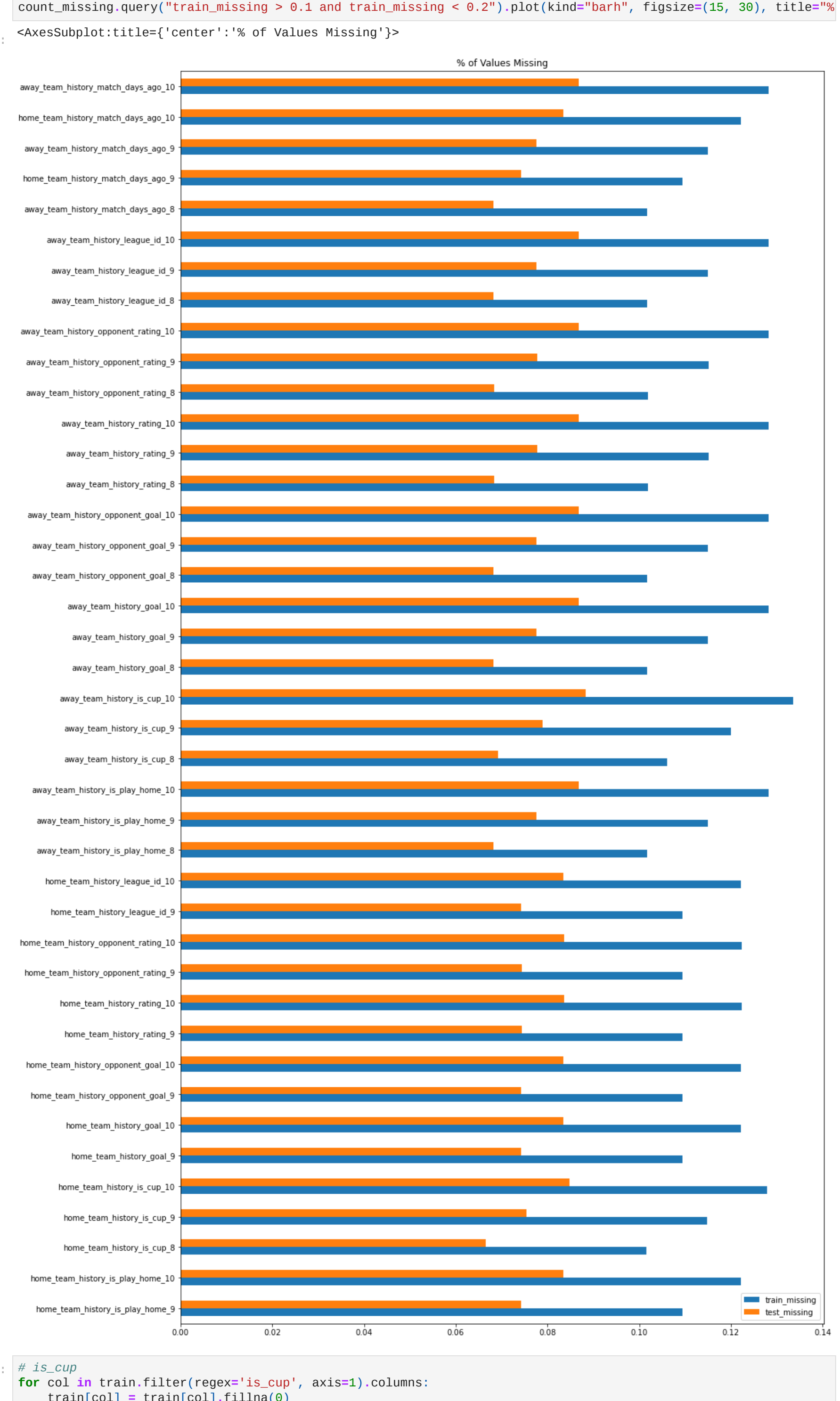
```
30
```

Out[37]:

	Missing Ratio
away_team_history_is_cup_10	8.828100
away_team_history_rating_10	8.689194
away_team_history_opponent_rating_10	8.689194
away_team_history_match_days_ago_10	8.675441
away_team_history_is_play_home_10	8.675441
away_team_history_opponent_goal_10	8.675441
away_team_history_goal_10	8.675441
away_team_history_league_id_10	8.675441
home_team_history_is_cup_10	8.480147
home_team_history_rating_10	8.361871
home_team_history_opponent_rating_10	8.361871
home_team_history_league_id_10	8.339866
home_team_history_opponent_goal_10	8.339866
home_team_history_is_play_home_10	8.339866
home_team_history_goal_10	8.339866
home_team_history_match_days_ago_10	8.339866
away_team_history_is_cup_9	7.902518
away_team_history_rating_9	7.781491
away_team_history_opponent_rating_9	7.781491
away_team_history_is_play_home_9	7.764987
away_team_history_goal_9	7.764987
away_team_history_opponent_goal_9	7.764987
away_team_history_match_days_ago_9	7.764987
away_team_history_league_id_9	7.764987
home_team_history_is_cup_9	7.538062
home_team_history_opponent_rating_9	7.447291
home_team_history_rating_9	7.447291
home_team_history_goal_9	7.429412
home_team_history_league_id_9	7.429412
home_team_history_match_days_ago_9	7.429412

In [38]:
```python
count_missing = pd.DataFrame([train.isna().mean(),test.isna().mean()]).T
count_missing = count_missing.rename(columns={0: "train_missing", 1: "test_missing"})
count_missing.query("train_missing > 0.1 and train_missing < 0.2").plot(kind="barh", figsize=(15, 30), title="%
```

Out[38]:
```
<AxesSubplot:title={'center':'% of Values Missing'}>
```



In [39]:
```python
# is_cup
for col in train.filter(regex='is_cup', axis=1).columns:
    train[col] = train[col].fillna(0)
# rating
for col in train.filter(regex='rating', axis=1).columns:
    train[col] = train[col].fillna(train[col].mean())
# is play home
for col in train.filter(regex='is_play_home', axis=1).columns:
    train[col] = train[col].fillna(0.5)
# league
for col in train.filter(regex='league', axis=1).columns:
    train[col] = train[col].fillna(0)
# goal
for col in train.filter(regex='goal', axis=1).columns:
    train[col] = train[col].fillna(train[col].mean())
#days
for col in train.filter(regex='days', axis=1).columns:
    train[col] = train[col].fillna(train[col].mean())

# ------------------------------------------test-----------------------------------------------

# is_cup
for col in test.filter(regex='is_cup', axis=1).columns:
    test[col] = test[col].fillna(0)
# rating
for col in test.filter(regex='rating', axis=1).columns:
    test[col] = test[col].fillna(test[col].mean())
# is play home
for col in test.filter(regex='is_play_home', axis=1).columns:
    test[col] = test[col].fillna(0.5)
# league
for col in test.filter(regex='league', axis=1).columns:
    test[col] = test[col].fillna(0)
# goal
for col in test.filter(regex='goal', axis=1).columns:
    test[col] = test[col].fillna(test[col].mean())
#days
for col in test.filter(regex='days', axis=1).columns:
    test[col] = test[col].fillna(test[col].mean())
```

In [40]:
```python
all_data_na = (train.isnull().sum() / len(train)) * 100
all_data_na = all_data_na.drop(all_data_na[all_data_na == 0].index).sort_values(ascending=False)[:30]
missing_data = pd.DataFrame({'Missing Ratio' :all_data_na})
print(len(all_data_na))
missing_data.head(30)
```

```
2
```

Out[40]:

	Missing Ratio
home_team_name	0.000901
away_team_name	0.000901

In [41]:
```python
train[train['home_team_name'].isnull()]
```

Out[41]:

	id	target	home_team_name	away_team_name	league_name	is_cup	home_team_history_is_play_home_1	home_team_history
33562	16622092	away	NaN	NaN	0	0	0.0	

1 rows × 166 columns

In [42]:
```python
train = train.drop([33562])
```

In [43]:
```python
all_data_na = (train.isnull().sum() / len(train)) * 100
all_data_na = all_data_na.drop(all_data_na[all_data_na == 0].index).sort_values(ascending=False)[:30]
missing_data = pd.DataFrame({'Missing Ratio' :all_data_na})
print(len(all_data_na))
missing_data.head(30)
```

```
0
```

Out[43]:

Missing Ratio

In [44]:
```python
train['target'] = train_target_scores['target'].map({'home': 0, 'draw': 1, 'away': 2})
train.head()
```

Out[44]:

	id	target	home_team_name	away_team_name	league_name	is_cup	home_team_history_is_play_home_1	home_team_history_is_p
0	11906497	2	Newell's Old Boys	River Plate	Superliga	False	0.0	
1	11984383	0	Real Estelí	Deportivo Las Sabanas	Primera Division	False	1.0	
2	11983301	1	UPNFM	Marathón	Liga Nacional	False	0.0	
3	11983471	2	León	Morelia	Liga MX	False	0.0	
4	11883005	0	Cobán Imperial	Iztapa	Liga Nacional	False	0.0	

5 rows × 166 columns

In [45]:
```python
test.head()
```

Out[45]:

	id	home_team_name	away_team_name	league_name	is_cup	home_team_history_is_play_home_1	home_team_history_is_play_hor
0	17761448	12 de Octubre	Sportivo Luqueño	Division 1	False	0.0	
1	17695487	Necaxa	Atlas	Liga MX	False	0.0	
2	17715496	Sertãozinho	EC São Bernardo	Paulista A2	False	1.0	
3	17715493	RB Brasil	XV de Piracicaba	Paulista A2	False	0.0	
4	17715492	Taubaté	Monte Azul	Paulista A2	False	0.0	

5 rows × 165 columns

In [46]:
```python
test.columns
```

```
Out[46]: Index(['id', 'home_team_name', 'away_team_name', 'league_name', 'is_cup',
       'home_team_history_is_play_home_1', 'home_team_history_is_play_home_2',
       'home_team_history_is_play_home_3', 'home_team_history_is_play_home_4',
       'home_team_history_is_play_home_5',
       ...
       'home_team_history_match_days_ago_6',
       'away_team_history_match_days_ago_6',
       'home_team_history_match_days_ago_7',
       'away_team_history_match_days_ago_7',
       'home_team_history_match_days_ago_8',
       'away_team_history_match_days_ago_8',
       'home_team_history_match_days_ago_9',
       'away_team_history_match_days_ago_9',
       'home_team_history_match_days_ago_10',
       'away_team_history_match_days_ago_10'],
      dtype='object', length=165)
```

In [47]:
```python
train['is_cup'] = train['is_cup'].map({False: 0, True: 1})
test['is_cup'] = test['is_cup'].map({False: 0, True: 1})
```

In [48]:
```python
# Add Column of sum of home_team_history_goal & away_team_history_goal
train['home_team_history_goal']=train[['home_team_history_goal_1','home_team_history_goal_2','home_team_history_
train['away_team_history_goal']=train[['away_team_history_goal_1','away_team_history_goal_2','away_team_history_
# Add Column of sum of home_team_opponent_history_goal & away_team_opponent_history_goal
train['home_team_history_opponent_goal']=train[['home_team_history_opponent_goal_1','home_team_history_opponent_
train['away_team_history_opponent_goal']=train[['away_team_history_opponent_goal_1','away_team_history_opponent_
# Add Column of sum of home_team_history_rating & away_team_history_rating
train['home_team_history_rating']=train[['home_team_history_rating_1','home_team_history_rating_2','home_team_h
train['away_team_history_rating']=train[['away_team_history_rating_1','away_team_history_rating_2','away_team_h
# Add Column of sum of home_team_history_opponent_rating & away_team_history_opponent_rating
train['home_team_history_opponent_rating']=train[['home_team_history_opponent_rating_1','home_team_history_oppo
train['away_team_history_opponent_rating']=train[['away_team_history_opponent_rating_1','away_team_history_oppo


# Delete Columns
for i in range(1,11):
    del train[f"home_team_history_goal_{i}"]
    del train[f"away_team_history_goal_{i}"]
    del train[f"home_team_history_opponent_goal_{i}"]
    del train[f"away_team_history_opponent_goal_{i}"]
    del train[f"home_team_history_rating_{i}"]
    del train[f"away_team_history_rating_{i}"]
    del train[f"home_team_history_opponent_rating_{i}"]
    del train[f"away_team_history_opponent_rating_{i}"]

train
```

Out[48]:

	id	target	home_team_name	away_team_name	league_name	is_cup	home_team_history_is_play_home_1	home_team_histor
0	11906497	2	Newell's Old Boys	River Plate	Superliga	0	0.0	
1	11984383	0	Real Estelí	Deportivo Las Sabanas	Primera Division	0	1.0	
2	11983301	1	UPNFM	Marathón	Liga Nacional	0	0.0	
3	11983471	2	León	Morelia	Liga MX	0	0.0	
4	11883005	0	Cobán Imperial	Iztapa	Liga Nacional	0	0.0	
...	...	...	...	...	...	...	...	
110933	18030016	1	Zamora Fútbol Club	Hermanos Colmenares	Primera Division	0	0.0	
110934	18030096	2	Royal Pari	Oriente Petrolero	Liga De Futbol Prof	0	0.0	
110935	17715497	1	São Bernardo	Água Santa	Paulista A2	0	0.0	
110936	17944153	2	Everton	La Serena	Primera Division	0	0.0	
110937	17786297	0	Colón	Arsenal de Sarandi	Superliga	0	0.0	

110937 rows × 94 columns

In [49]:
```python
# Add Column of sum of home_team_history_goal & away_team_history_goal
test['home_team_history_goal']=test[['home_team_history_goal_1','home_team_history_goal_2','home_team_history_g
test['away_team_history_goal']=test[['away_team_history_goal_1','away_team_history_goal_2','away_team_history_g
# Add Column of sum of home_team_opponent_history_goal & away_team_opponent_history_goal
test['home_team_history_opponent_goal']=test[['home_team_history_opponent_goal_1','home_team_history_opponent_g
test['away_team_history_opponent_goal']=test[['away_team_history_opponent_goal_1','away_team_history_opponent_g
# Add Column of sum of home_team_history_rating & away_team_history_rating
test['home_team_history_rating']=test[['home_team_history_rating_1','home_team_history_rating_2','home_team_his
test['away_team_history_rating']=test[['away_team_history_rating_1','away_team_history_rating_2','away_team_his
# Add Column of sum of home_team_history_opponent_rating & away_team_history_opponent_rating
test['home_team_history_opponent_rating']=test[['home_team_history_opponent_rating_1','home_team_history_oppone
test['away_team_history_opponent_rating']=test[['away_team_history_opponent_rating_1','away_team_history_oppone


# Delete Columns
for i in range(1,11):
    del test[f"home_team_history_goal_{i}"]
    del test[f"away_team_history_goal_{i}"]
    del test[f"home_team_history_opponent_goal_{i}"]
    del test[f"away_team_history_opponent_goal_{i}"]
    del test[f"home_team_history_rating_{i}"]
    del test[f"away_team_history_rating_{i}"]
    del test[f"home_team_history_opponent_rating_{i}"]
    del test[f"away_team_history_opponent_rating_{i}"]

test
```

Out[49]:

	id	home_team_name	away_team_name	league_name	is_cup	home_team_history_is_play_home_1	home_team_history_is_play
0	17761448	12 de Octubre	Sportivo Luqueño	Division 1	0	0.0	
1	17695487	Necaxa	Atlas	Liga MX	0	0.0	
2	17715496	Sertãozinho	EC São Bernardo	Paulista A2	0	1.0	
3	17715493	RB Brasil	XV de Piracicaba	Paulista A2	0	0.0	
4	17715492	Taubaté	Monte Azul	Paulista A2	0	0.0	
...	...	...	...	...	...	...	
72706	18450246	Cerro	Defensor Sporting	Segunda Division	0	1.0	
72707	18164889	Boca Juniors	Newell's Old Boys	Superliga	0	0.0	
72708	18449018	Mexico W	Canada W	Friendly International Women	0	1.0	
72709	17958831	Flamengo	Ceará	Serie A	0	0.0	
72710	18441629	Academia Puerto Cabello	Portuguesa	Primera Division	0	0.0	

72711 rows × 93 columns

In [50]:
```python
test['target'] = train_target_scores['target'].map({'home': 0, 'draw': 1, 'away': 2})
test.head()
```

Out[50]:

	id	home_team_name	away_team_name	league_name	is_cup	home_team_history_is_play_home_1	home_team_history_is_play_hor
0	17761448	12 de Octubre	Sportivo Luqueño	Division 1	0	0.0	
1	17695487	Necaxa	Atlas	Liga MX	0	0.0	
2	17715496	Sertãozinho	EC São Bernardo	Paulista A2	0	1.0	
3	17715493	RB Brasil	XV de Piracicaba	Paulista A2	0	0.0	
4	17715492	Taubaté	Monte Azul	Paulista A2	0	0.0	

5 rows × 94 columns

In [51]:
```python
# Delete Columns history_league_id_{i}
for i in range(1,11):
    del train[f"home_team_history_league_id_{i}"]
    del train[f"away_team_history_league_id_{i}"]
    del test[f"home_team_history_league_id_{i}"]
    del test[f"away_team_history_league_id_{i}"]

test
```

Out[51]:

	id	home_team_name	away_team_name	league_name	is_cup	home_team_history_is_play_home_1	home_team_history_is_play
0	17761448	12 de Octubre	Sportivo Luqueño	Division 1	0	0.0	
1	17695487	Necaxa	Atlas	Liga MX	0	0.0	
2	17715496	Sertãozinho	EC São Bernardo	Paulista A2	0	1.0	
3	17715493	RB Brasil	XV de Piracicaba	Paulista A2	0	0.0	
4	17715492	Taubaté	Monte Azul	Paulista A2	0	0.0	
...	...	...	...	...	...	...	
72706	18450246	Cerro	Defensor Sporting	Segunda Division	0	1.0	
72707	18164889	Boca Juniors	Newell's Old Boys	Superliga	0	0.0	
72708	18449018	Mexico W	Canada W	Friendly International Women	0	1.0	
72709	17958831	Flamengo	Ceará	Serie A	0	0.0	
72710	18441629	Academia Puerto Cabello	Portuguesa	Primera Division	0	0.0	

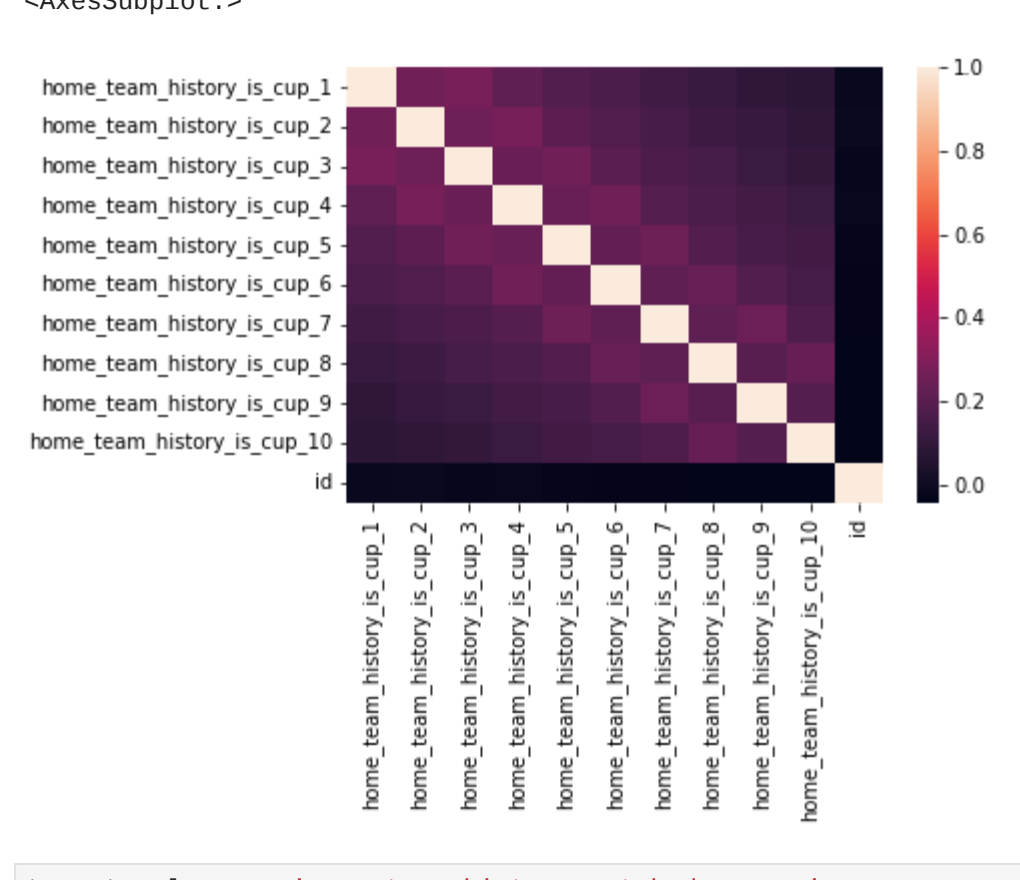
72711 rows × 74 columns

In [52]:
```python
del test['home_team_name']
del test['away_team_name']
del test['league_name']
```

In [53]:
```python
target_columns = 'home_team_history_is_cup'
###
cols = []
for col in train.filter(regex=target_columns, axis=1).columns:
    cols.append(col)
print(cols)
corr = pd.concat([train[cols],train_target_scores], axis=1).corr()
sns.heatmap(corr)
```

```
['home_team_history_is_cup_1', 'home_team_history_is_cup_2', 'home_team_history_is_cup_3', 'home_team_history_i
s_cup_4', 'home_team_history_is_cup_5', 'home_team_history_is_cup_6', 'home_team_history_is_cup_7', 'home_team_
history_is_cup_8', 'home_team_history_is_cup_9', 'home_team_history_is_cup_10']
```

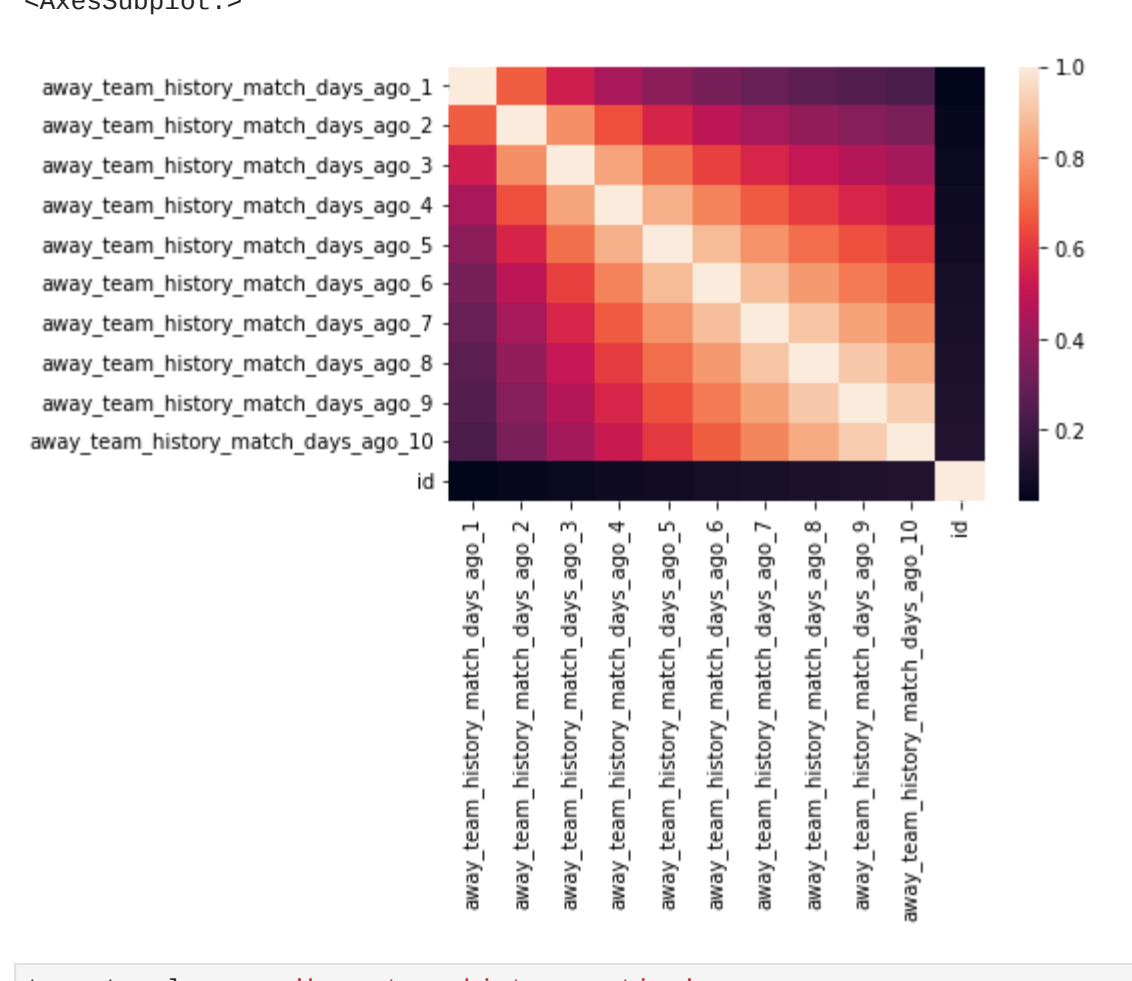
Out[53]: <AxesSubplot:>



In [54]:
```python
target_columns = 'away_team_history_match_days_ago'
###
cols = []
for col in train.filter(regex=target_columns, axis=1).columns:
    cols.append(col)
print(cols)
corr = pd.concat([train[cols],train_target_scores], axis=1).corr()
sns.heatmap(corr)
```

```
['away_team_history_match_days_ago_1', 'away_team_history_match_days_ago_2', 'away_team_history_match_days_ago_
3', 'away_team_history_match_days_ago_4', 'away_team_history_match_days_ago_5', 'away_team_history_match_days_a
go_6', 'away_team_history_match_days_ago_7', 'away_team_history_match_days_ago_8', 'away_team_history_match_day
s_ago_9', 'away_team_history_match_days_ago_10']
```

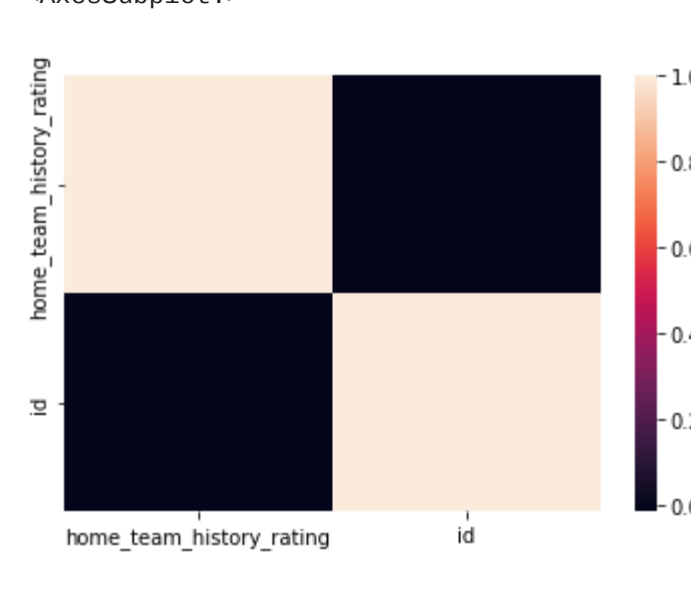
Out[54]: <AxesSubplot:>



In [55]:
```python
target_columns = 'home_team_history_rating'
###
cols = []
for col in train.filter(regex=target_columns, axis=1).columns:
    cols.append(col)
print(cols)
corr = pd.concat([train[cols],train_target_scores], axis=1).corr()
sns.heatmap(corr)
```

```
['home_team_history_rating']
```

Out[55]: <AxesSubplot:>



Modeling

In [56]:
```python
def preprocessing(train):

    rating_features = [x for x in  train if 'rating' in x]

    #Make X and y
    X = train[rating_features]
    y = train['target']

    print(y.value_counts())

    return X, y
```

In [57]:
```python
X_train, y_train = preprocessing(train)
```

```
0    48113
2    35173
1    27651
Name: target, dtype: int64
```

In [58]:
```python
X__test, y__test = preprocessing(test)
```

```
0    31932
2    23202
1    17577
Name: target, dtype: int64
```

In [59]:
```python
# Creating and training our model
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score, confusion_matrix, classification_report
from sklearn.model_selection import learning_curve

model = LogisticRegression(C=0.001,penalty='l2',solver='lbfgs')
```

In [60]:
```python
def evaluation(model):

    model.fit(X_train, y_train)
    ypred = model.predict(X__test)

    print(confusion_matrix(y__test, ypred))
    print(classification_report(y__test, ypred))
```

In [61]:
```python
  evaluation(model)
```

```
[[22482    29  9421]
 [12517    15  5045]
 [16351    19  6832]]
              precision    recall  f1-score   support

           0       0.44      0.70      0.54     31932
           1       0.24      0.00      0.00     17577
           2       0.32      0.29      0.31     23202

    accuracy                           0.40     72711
   macro avg       0.33      0.33      0.28     72711
weighted avg       0.35      0.40      0.34     72711
```

In [62]:
```python
prediction = model.predict_proba(X__test)
prediction
```

```
Out[62]: array([[0.36062054, 0.29191273, 0.34746673],
       [0.36067623, 0.29138757, 0.34793619],
       [0.38929289, 0.29875665, 0.31195046],
       ...,
       [0.17119731, 0.19384419, 0.63495849],
       [0.5564164 , 0.25661563, 0.18696798],
       [0.35804471, 0.29991891, 0.34203639]])
```

In [63]:
```python
df_sub = pd.DataFrame({
    'id': test['id'],
    'away': prediction[:, 0],
    'draw': prediction[:, 1],
    'home': prediction[:, 2]
})
df_sub
```

Out[63]:

	id	away	draw	home
0	17761448	0.360621	0.291913	0.347467
1	17695487	0.360676	0.291388	0.347936
2	17715496	0.389293	0.298757	0.311950
3	17715493	0.241205	0.293148	0.465647
4	17715492	0.478397	0.285097	0.236507
...	...	...	...	...
72706	18450246	0.375724	0.301352	0.322923
72707	18164889	0.498282	0.280137	0.221581
72708	18449018	0.171197	0.193844	0.634958
72709	17958831	0.556416	0.256616	0.186968
72710	18441629	0.358045	0.299919	0.342036

72711 rows × 4 columns

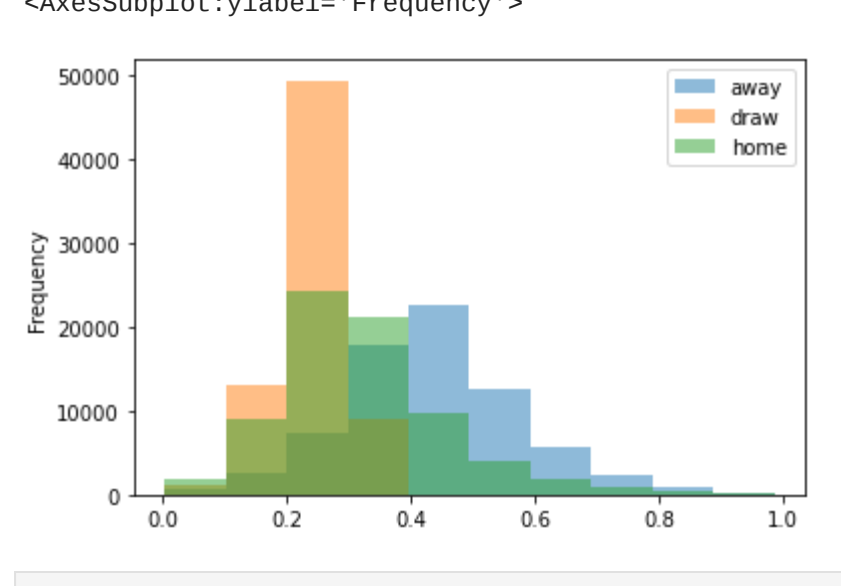
In [64]:
```python
pd.concat([
    train['target'].value_counts(normalize=True).sort_index().rename('observed'),
    pd.Series(np.argmax(df_sub.iloc[:, 1:].values, axis=1), name='predicted').value_counts(normalize=True).sort_
], axis=1)
```

Out[64]:

	observed	predicted
0	0.433697	0.706221
1	0.249250	0.000866
2	0.317054	0.292913

In [65]:
```python
df_sub.iloc[:, 1:].plot.hist(alpha=0.5)
```

Out[65]: <AxesSubplot:ylabel='Frequency'>



In [66]:
```python
df_sub[['id', 'home', 'away', 'draw']].to_csv('submission.csv', index=False)
df_sub.head()
```

Out[66]:

	id	away	draw	home
0	17761448	0.360621	0.291913	0.347467
1	17695487	0.360676	0.291388	0.347936
2	17715496	0.389293	0.298757	0.311950
3	17715493	0.241205	0.293148	0.465647
4	17715492	0.478397	0.285097	0.236507

In [ ]: