In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
df = pd.read_csv('deliveries.csv', encoding = "ISO-8859-1", na_filter=False)
```

In [3]:

```
df.head()
```

Out[3]:

	season	match_id	venue	inning	batting_team	bowling_team	batsman	non_striker	bowler	ball	...	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	player_dismissed	di
0	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	SC Ganguly	BB McCullum	P Kumar	0.1	...	0	0	0	1	1		
1	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullum	SC Ganguly	P Kumar	0.2	...	0	0	0	0	0		
2	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullum	SC Ganguly	P Kumar	0.3	...	0	0	0	1	1		
3	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullum	SC Ganguly	P Kumar	0.3	...	0	0	0	0	0		
4	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullum	SC Ganguly	P Kumar	0.4	...	0	0	0	0	0		

5 rows × 23 columns

In [4]:

```
df.tail()
```

Out[4]:

	season	match_id	venue	inning	batting_team	bowling_team	batsman	non_striker	bowler	ball	...	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	player_dismiss
131996	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	RA Jadeja	SR Watson	SL Malinga	19.2	...	0	0	1	0	1	
131997	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	SR Watson	RA Jadeja	SL Malinga	19.3	...	0	0	2	0	2	
131998	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	SR Watson	RA Jadeja	SL Malinga	19.4	...	0	0	1	0	1	SR Wats
131999	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	SN Thakur	RA Jadeja	SL Malinga	19.5	...	0	0	2	0	2	
132000	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	SN Thakur	RA Jadeja	SL Malinga	19.6	...	0	0	0	0	0	SN Tha

5 rows × 23 columns

In [5]:

```
df.shape
```

Out[5]:

```
(132001, 23)
```

In [6]:

```
df.columns
```

Out[6]:

```
Index(['season', 'match_id', 'venue', 'inning', 'batting_team', 'bowling_team',
       'batsman', 'non_striker', 'bowler', 'ball', 'wide_runs', 'bye_runs',
       'legbye_runs', 'noball_runs', 'penalty_runs', 'batsman_runs',
       'extra_runs', 'total_runs', 'player_dismissed', 'dismissal_kind',
       'fielder', 'wickets', 'commentary'],
      dtype='object')
```

In [7]:
```
df.isna().sum()
```

Out[7]:
```
season              0
match_id            0
venue               0
inning              0
batting_team        0
bowling_team        0
batsman             0
non_striker         0
bowler              0
ball                0
wide_runs           0
bye_runs            0
legbye_runs         0
noball_runs         0
penalty_runs        0
batsman_runs        0
extra_runs          0
total_runs          0
player_dismissed    0
dismissal_kind      0
fielder             0
wickets             0
commentary          0
dtype: int64
```

In [8]:
```
df.dtypes
```

Out[8]:
```
season                int64
match_id             object
venue                object
inning                int64
batting_team         object
bowling_team         object
batsman              object
non_striker          object
bowler               object
ball                float64
wide_runs             int64
bye_runs              int64
legbye_runs           int64
noball_runs           int64
penalty_runs          int64
batsman_runs          int64
extra_runs            int64
total_runs            int64
player_dismissed     object
dismissal_kind       object
fielder              object
wickets               int64
commentary           object
dtype: object
```

In [9]:
```
df.batting_team.unique()
```

Out[9]:
```
array(['Kolkata Knight Riders', 'Royal Challengers Bangalore',
       'Chennai Super Kings', 'Kings XI Punjab', 'Rajasthan Royals',
       'Delhi Daredevils', 'Mumbai Indians', 'Deccan Chargers',
       'Sunrisers Hyderabad', 'Delhi Capitals'], dtype=object)
```

In [10]:
```
columns_to_remove = ['batsman', 'non_striker', 'player_dismissed', 'fielder']
df.drop(labels=columns_to_remove, axis=1, inplace=True)
```

In [11]:
```
df.head()
```

Out[11]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismissal_ki
0	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	P Kumar	0.1	0	0	1	0	0	0	1	1	
1	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	P Kumar	0.2	0	0	0	0	0	0	0	0	
2	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	P Kumar	0.3	1	0	0	0	0	0	1	1	
3	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	P Kumar	0.3	0	0	0	0	0	0	0	0	
4	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	P Kumar	0.4	0	0	0	0	0	0	0	0	

In [12]:
```
df.loc[(df.batting_team=='Delhi Daredevils'),'batting_team'] = 'Delhi Capitals'
df.loc[(df.bowling_team=='Delhi Daredevils'),'bowling_team'] = 'Delhi Capitals'
df.loc[(df.batting_team=='Delhi Daredevils')|(df.bowling_team=='Delhi Daredevils')]
```

Out[12]:

season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismissal_kind

In [13]:

```
rcb_spin = ['SB Joshi', 'KP Appanna', 'A Kumble', 'J Arunkumar', 'Abdur Razzak', 'CL White', 'TM Dilshan',
            'KP Pietersen', 'RE van der Merwe', 'S Sriram', 'DL Vettori', 'KB Arun Karthik', 'CH Gayle', 'S Sriram',
            'AA Kazi', 'R Ninan', 'J Syed Mohammad', 'M Muralitharan', 'M Kartik', 'P R Barman',
            'YS Chahal', 'Yuvraj Singh', 'SB Jakati', 'Iqbal Abdulla', 'T Shamsi', 'Parvez Rasool',
            'S Baby', 'TM Head', 'KM Jadhav', 'P Negi', 'S Badree', 'Washington Sundar', 'M Ali', 'M Ashwin']

mi_spin = ['A Dananjaya', 'AG Murtaza', 'A Roy', 'GJ Maxwell', 'Harbhajan Singh', 'J Suchith', 'J Yadav',
           'JP Duminy', 'KH Pandya', 'KV Sharma', 'M Markande', 'N Rana', 'PP Ojha', 'RD Chahar', 'RG Sharma',
           'RJ Peterson', 'RW Price', 'S Gopal', 'SD Chitnis', 'ST Jayasuriya', 'TL Suman', 'VS Yeligati']

kkr_spin = ['BAW Mendis','BJ Hodge', 'DJ Hussey', 'GB Hogg', 'J Botha', 'KC Cariappa', 'M Kartik', 'CH Gayle',
            'Iqbal Abdulla', 'Kuldeep Yadav', 'MB Parmar', 'MK Tiwary', 'Mohammad Hafeez', 'N Rana',
            'PP Chawla', 'RS Gavaskar', 'S Ladda', 'Shakib Al Hasan', 'SMSM Senanayake', 'SP Narine', 'YK Pathan']

csk_spin = ['DJ Hussey', 'F du Plessis', 'Harbhajan Singh', 'Imran Tahir', 'RA Jadeja', 'KV Sharma', 'M Santner',
            'M Muralitharan', 'P Negi', 'R Ashwin', 'S Badree', 'S Randiv', 'S Vidyut', 'SB Jakati', 'SK Raina',
            'PP Chawla', 'RS Gavaskar', 'S Ladda', 'Shakib Al Hasan', 'SMSM Senanayake', 'SP Narine', 'YK Pathan']

dc_spin = ['A Mishra', 'AA Jhunjhunwala', 'Ankit Sharma', 'CL White', 'PP Ojha', 'DB Ravi Teja', 'JP Duminy',
           'LPC Silva', 'R Sharma', 'RG Sharma', 'S Dhawan', 'Shahid Afridi', 'TL Suman', 'Y Venugopal Rao']

dd_spin = ['A Mishra', 'AJ Finch', 'AR Patel', 'BMAJ Mendis', 'DL Vettori', 'GJ Maxwell', 'H Vihari', 'Imran Tahir',
           'J Suchith', 'J Botha', 'J Yadav', 'JP Duminy', 'KP Pietersen', 'MK Tiwary', 'P Negi', 'R Sharma',
           'R Tewatia', 'RE van der Merwe', 'S Ladda', 'S Lamichhane', 'S Nadeem', 'S Sriram', 'Shoaib Malik',
           'TM Dilshan', 'Sunny Gupta', 'V Sehwag', 'Y Venugopal Rao', 'Yuvraj Singh', 'Y Nagar']

srh_spin = ['A Mishra', 'Ankit Sharma', 'Bipul Sharma', 'CL White', 'DJ Hooda', 'KS Williamson', 'KV Sharma',
            'Mohammad Nabi', 'Parvez Rasool', 'Rashid Khan', 'S Nadeem', 'Shakib Al Hasan', 'Y Venugopal Rao', 'YK Pathan',
            'Yuvraj Singh']

rr_spin = ['A Chandila', 'AA Chavan', 'AA Jhunjhunwala', 'AC Voges', 'AJ Finch', 'AL Menaria', 'Ankit Sharma',
           'AS Raut', 'BJ Hodge', 'D Short', 'DJ Hooda', 'D Salunkhe', 'GB Hogg', 'I Sodhi', 'J Botha', 'K Gowtham',
           'L Livingstone', 'LRPL Taylor', 'M Lomror', 'PV Tambe', 'ND Doshi', 'R Parag', 'R Tewatia', 'S Badree',
           'S Gopal', 'RA Jadeja', 'S Midhun', 'SK Warne', 'YK Pathan']

punj_spin = ['AC Gilchrist', 'AR Patel', 'BA Bhatt', 'Bipul Sharma', 'DJ Hussey', 'GJ Maxwell', 'Gurkeerat Singh'
            'H Brar', 'Karanveer Singh', 'KC Cariappa', 'M Ashwin', 'M Kartik', 'M Ur Rahman', 'M Vijay', 'MK Tiwary',
             'P Sahu', 'PP Chawla', 'R Ashwin', 'R Tewatia', 'RR Powar', 'Shivam Sharma', 'SN Khan', 'Swapnil Singh',
             'V Chakravarthy', 'Yuvraj Singh']
```

In [14]:

```
df.loc[(df.bowling_team=='Royal Challengers Bangalore')&(df.bowler.isin(rcb_spin)),'bowler'] = 'spinner'
df.loc[(df.bowling_team=='Delhi Capitals')&(df.bowler.isin(dd_spin)),'bowler'] = 'spinner'
df.loc[(df.bowling_team=='Mumbai Indians')&(df.bowler.isin(mi_spin)),'bowler'] = 'spinner'
df.loc[(df.bowling_team=='Chennai Super Kings')&(df.bowler.isin(csk_spin)),'bowler'] = 'spinner'
df.loc[(df.bowling_team=='Kings XI Punjab')&(df.bowler.isin(punj_spin)),'bowler'] = 'spinner'
df.loc[(df.bowling_team=='Sunrisers Hyderabad')&(df.bowler.isin(srh_spin)),'bowler'] = 'spinner'
df.loc[(df.bowling_team=='Kolkata Knight Riders')&(df.bowler.isin(kkr_spin)),'bowler'] = 'spinner'
df.loc[(df.bowling_team=='Deccan Chargers')&(df.bowler.isin(dc_spin)),'bowler'] = 'spinner'
df.loc[(df.bowling_team=='Rajasthan Royals')&(df.bowler.isin(rr_spin)),'bowler'] = 'spinner'
```

```
In [15]: df.loc[(df.bowling_team=='Royal Challengers Bangalore')&(df.bowler=='spinner')]
```

Out[15]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismis
56	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.1	0	0	0	0	0	1	0	1	
57	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.2	0	0	0	0	0	1	0	1	
58	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.3	0	0	0	0	0	1	0	1	
59	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.4	0	0	0	0	0	0	0	0	
60	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.5	0	0	0	0	0	6	0	6	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
130411	2019	53	Bangalore	1	Sunrisers Hyderabad	Royal Challengers Bangalore	spinner	15.2	0	0	0	0	0	1	0	1	
130412	2019	53	Bangalore	1	Sunrisers Hyderabad	Royal Challengers Bangalore	spinner	15.3	0	0	0	0	0	1	0	1	
130413	2019	53	Bangalore	1	Sunrisers Hyderabad	Royal Challengers Bangalore	spinner	15.4	0	0	0	0	0	1	0	1	
130414	2019	53	Bangalore	1	Sunrisers Hyderabad	Royal Challengers Bangalore	spinner	15.5	0	0	0	0	0	0	0	0	
130415	2019	53	Bangalore	1	Sunrisers Hyderabad	Royal Challengers Bangalore	spinner	15.6	0	0	0	0	0	1	0	1	

5355 rows × 19 columns

In [16]:
```
df.loc[(df.bowling_team=='Delhi Capitals')&(df.bowler=='spinner')]
```

Out[16]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	d
540	2008	3	Delhi	1	Rajasthan Royals	Delhi Capitals	spinner	11.1	0	0	0	0	0	6	0	6	
541	2008	3	Delhi	1	Rajasthan Royals	Delhi Capitals	spinner	11.2	0	0	0	0	0	4	0	4	
542	2008	3	Delhi	1	Rajasthan Royals	Delhi Capitals	spinner	11.3	0	0	0	0	0	1	0	1	
543	2008	3	Delhi	1	Rajasthan Royals	Delhi Capitals	spinner	11.4	0	0	0	0	0	0	0	0	
544	2008	3	Delhi	1	Rajasthan Royals	Delhi Capitals	spinner	11.5	0	0	0	0	0	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
131731	2019	semi-final2	Visakhapatnam	2	Chennai Super Kings	Delhi Capitals	spinner	15.2	0	0	0	0	0	2	0	2	
131732	2019	semi-final2	Visakhapatnam	2	Chennai Super Kings	Delhi Capitals	spinner	15.3	0	0	0	0	0	1	0	1	
131733	2019	semi-final2	Visakhapatnam	2	Chennai Super Kings	Delhi Capitals	spinner	15.4	0	0	0	0	0	4	0	4	
131734	2019	semi-final2	Visakhapatnam	2	Chennai Super Kings	Delhi Capitals	spinner	15.5	0	0	0	0	0	1	0	1	
131735	2019	semi-final2	Visakhapatnam	2	Chennai Super Kings	Delhi Capitals	spinner	15.6	0	0	0	0	0	0	0	0	

5188 rows × 19 columns

In [17]:
```
df.loc[(df.bowling_team=='Mumbai Indians')&(df.bowler=='spinner')]
```

Out[17]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismi
858	2008	4	Mumbai	2	Royal Challengers Bangalore	Mumbai Indians	spinner	7.1	0	0	0	0	0	2	0	2	
859	2008	4	Mumbai	2	Royal Challengers Bangalore	Mumbai Indians	spinner	7.2	0	0	0	0	0	1	0	1	
860	2008	4	Mumbai	2	Royal Challengers Bangalore	Mumbai Indians	spinner	7.3	0	0	0	0	0	1	0	1	
861	2008	4	Mumbai	2	Royal Challengers Bangalore	Mumbai Indians	spinner	7.4	0	0	0	0	0	1	0	1	
862	2008	4	Mumbai	2	Royal Challengers Bangalore	Mumbai Indians	spinner	7.5	0	0	0	0	0	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
131984	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.2	0	0	0	0	0	6	0	6	
131985	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.3	0	0	0	0	0	6	0	6	
131986	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.4	0	0	0	0	0	6	0	6	
131987	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.5	0	0	0	0	0	0	0	0	
131988	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.6	0	0	0	0	0	1	0	1	

5493 rows × 19 columns

In [18]:
```
df.loc[(df.bowling_team=='Chennai Super Kings')&(df.bowler=='spinner')]
```

Out[18]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismi
387	2008	2	Mohali	2	Kings XI Punjab	Chennai Super Kings	spinner	6.1	0	0	0	0	0	1	0	1	
388	2008	2	Mohali	2	Kings XI Punjab	Chennai Super Kings	spinner	6.2	0	0	0	0	0	1	0	1	
389	2008	2	Mohali	2	Kings XI Punjab	Chennai Super Kings	spinner	6.3	0	0	0	0	0	1	0	1	
390	2008	2	Mohali	2	Kings XI Punjab	Chennai Super Kings	spinner	6.4	0	0	0	0	0	1	0	1	
391	2008	2	Mohali	2	Kings XI Punjab	Chennai Super Kings	spinner	6.5	0	0	1	0	0	0	1	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
131854	2019	final	Hyderabad	1	Mumbai Indians	Chennai Super Kings	spinner	16.2	0	0	0	0	0	1	0	1	
131855	2019	final	Hyderabad	1	Mumbai Indians	Chennai Super Kings	spinner	16.3	0	0	0	0	0	1	0	1	
131856	2019	final	Hyderabad	1	Mumbai Indians	Chennai Super Kings	spinner	16.4	0	0	0	0	0	0	0	0	
131857	2019	final	Hyderabad	1	Mumbai Indians	Chennai Super Kings	spinner	16.5	0	0	0	0	0	6	0	6	
131858	2019	final	Hyderabad	1	Mumbai Indians	Chennai Super Kings	spinner	16.6	0	0	0	0	0	1	0	1	

6682 rows × 19 columns

In [19]:
```
df.loc[(df.bowling_team=='Rajasthan Royals')&(df.bowler=='spinner')]
```

Out[19]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismissal
646	2008	3	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	8.1	0	0	0	0	0	0	0	0	
647	2008	3	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	8.2	0	0	0	0	0	1	0	1	
648	2008	3	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	8.3	0	0	0	0	0	4	0	4	
649	2008	3	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	8.4	0	0	0	0	0	0	0	0	
650	2008	3	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	8.5	0	0	0	0	0	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
130313	2019	52	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	15.3	0	0	0	0	0	1	0	1	
130314	2019	52	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	15.4	0	0	0	0	0	1	0	1	
130315	2019	52	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	15.5	0	0	0	0	0	6	0	6	
130316	2019	52	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	15.6	0	0	0	0	0	1	0	1	
130317	2019	52	Delhi	2	Delhi Capitals	Rajasthan Royals	spinner	16.1	0	0	0	0	0	6	0	6	

4538 rows × 19 columns

In [20]:
```
df.loc[(df.bowling_team=='Deccan Chargers')&(df.bowler=='spinner')]
```

Out[20]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismis
1134	2008	5	Kolkata	2	Kolkata Knight Riders	Deccan Chargers	spinner	12.1	1	0	0	0	0	0	1	1	
1135	2008	5	Kolkata	2	Kolkata Knight Riders	Deccan Chargers	spinner	12.1	0	0	0	0	0	0	0	0	
1136	2008	5	Kolkata	2	Kolkata Knight Riders	Deccan Chargers	spinner	12.2	0	0	0	0	0	0	0	0	
1137	2008	5	Kolkata	2	Kolkata Knight Riders	Deccan Chargers	spinner	12.3	0	0	0	0	0	1	0	1	
1138	2008	5	Kolkata	2	Kolkata Knight Riders	Deccan Chargers	spinner	12.4	1	0	0	0	0	0	1	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
51189	2012	53	Hyderabad	2	Royal Challengers Bangalore	Deccan Chargers	spinner	17.2	0	0	0	0	0	1	0	1	
51190	2012	53	Hyderabad	2	Royal Challengers Bangalore	Deccan Chargers	spinner	17.3	0	0	0	0	0	2	0	2	
51191	2012	53	Hyderabad	2	Royal Challengers Bangalore	Deccan Chargers	spinner	17.4	0	0	0	0	0	2	0	2	
51192	2012	53	Hyderabad	2	Royal Challengers Bangalore	Deccan Chargers	spinner	17.5	0	0	0	0	0	1	0	1	
51193	2012	53	Hyderabad	2	Royal Challengers Bangalore	Deccan Chargers	spinner	17.6	0	0	0	0	0	0	0	0	

2217 rows × 19 columns

```
In [21]: df.loc[(df.bowling_team=='Kings XI Punjab')&(df.bowler=='spinner')]
```

Out[21]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismissal
287	2008	2	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	10.1	0	0	0	0	0	6	0	6	
288	2008	2	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	10.2	0	0	0	0	0	1	0	1	
289	2008	2	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	10.3	0	0	0	0	0	2	0	2	
290	2008	2	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	10.4	0	0	0	0	0	0	0	0	
291	2008	2	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	10.5	0	0	0	0	0	6	0	6	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
130668	2019	54	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	17.2	0	0	0	0	0	1	0	1	
130669	2019	54	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	17.3	0	0	0	0	0	1	0	1	
130670	2019	54	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	17.4	0	0	0	0	0	0	0	0	
130671	2019	54	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	17.5	0	0	0	0	0	0	0	0	
130672	2019	54	Mohali	1	Chennai Super Kings	Kings XI Punjab	spinner	17.6	0	0	0	0	0	1	0	1	

5721 rows × 19 columns

In [22]:
```
df.loc[(df.bowling_team=='Kolkata Knight Riders')&(df.bowler=='spinner')]
```

Out[22]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismiss
977	2008	5	Kolkata	1	Deccan Chargers	Kolkata Knight Riders	spinner	6.1	0	0	0	0	0	1	0	1	
978	2008	5	Kolkata	1	Deccan Chargers	Kolkata Knight Riders	spinner	6.2	0	0	0	0	0	0	0	0	
979	2008	5	Kolkata	1	Deccan Chargers	Kolkata Knight Riders	spinner	6.3	0	0	0	0	0	1	0	1	
980	2008	5	Kolkata	1	Deccan Chargers	Kolkata Knight Riders	spinner	6.4	0	0	0	0	0	4	0	4	
981	2008	5	Kolkata	1	Deccan Chargers	Kolkata Knight Riders	spinner	6.5	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
131008	2019	55	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	spinner	13.2	0	0	0	0	0	6	0	6	
131009	2019	55	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	spinner	13.3	0	0	0	0	0	1	0	1	
131010	2019	55	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	spinner	13.4	0	0	0	0	0	0	0	0	
131011	2019	55	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	spinner	13.5	0	0	0	0	0	0	0	0	
131012	2019	55	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	spinner	13.6	0	0	0	0	0	4	0	4	

6949 rows × 19 columns

In [23]:
```
df.loc[(df.bowling_team=='Sunrisers Hyderabad')&(df.bowler=='spinner')]
```

Out[23]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	d
53441	2013	5	Hyderabad	1	Royal Challengers Bangalore	Sunrisers Hyderabad	spinner	4.1	0	0	0	0	0	5	0	5	
53442	2013	5	Hyderabad	1	Royal Challengers Bangalore	Sunrisers Hyderabad	spinner	4.2	0	0	0	0	0	0	0	0	
53443	2013	5	Hyderabad	1	Royal Challengers Bangalore	Sunrisers Hyderabad	spinner	4.3	0	0	0	0	0	0	0	0	
53444	2013	5	Hyderabad	1	Royal Challengers Bangalore	Sunrisers Hyderabad	spinner	4.4	0	0	0	0	0	4	0	4	
53445	2013	5	Hyderabad	1	Royal Challengers Bangalore	Sunrisers Hyderabad	spinner	4.5	0	0	0	0	0	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
131485	2019	eliminator	Visakhapatnam	2	Delhi Capitals	Sunrisers Hyderabad	spinner	15.2	0	0	0	0	0	0	0	0	
131486	2019	eliminator	Visakhapatnam	2	Delhi Capitals	Sunrisers Hyderabad	spinner	15.3	0	0	0	0	0	0	0	0	
131487	2019	eliminator	Visakhapatnam	2	Delhi Capitals	Sunrisers Hyderabad	spinner	15.4	0	0	0	0	0	1	0	1	
131488	2019	eliminator	Visakhapatnam	2	Delhi Capitals	Sunrisers Hyderabad	spinner	15.5	0	0	0	0	0	6	0	6	
131489	2019	eliminator	Visakhapatnam	2	Delhi Capitals	Sunrisers Hyderabad	spinner	15.6	0	0	0	0	0	2	0	2	

3235 rows × 19 columns

In [24]:
```
l = ['spinner']
df.loc[~df.bowler.isin(l),'bowler'] = 'pacer'
```

In [25]:
```
df.loc[df.bowler=='pacer']
```

Out[25]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismi
0	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.1	0	0	1	0	0	0	1	1	
1	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.2	0	0	0	0	0	0	0	0	
2	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	1	0	0	0	0	0	1	1	
3	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	0	0	0	0	0	0	0	0	
4	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.4	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
131996	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.2	0	0	0	0	0	1	0	1	
131997	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.3	0	0	0	0	0	2	0	2	
131998	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.4	0	0	0	0	0	1	0	1	
131999	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.5	0	0	0	0	0	2	0	2	
132000	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.6	0	0	0	0	0	0	0	0	

86623 rows × 19 columns

In [26]:

```
df.loc[df.bowler=='spinner']
```

Out[26]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismi
56	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.1	0	0	0	0	0	1	0	1	
57	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.2	0	0	0	0	0	1	0	1	
58	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.3	0	0	0	0	0	1	0	1	
59	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.4	0	0	0	0	0	0	0	0	
60	2008	1	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	spinner	9.5	0	0	0	0	0	6	0	6	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
131984	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.2	0	0	0	0	0	6	0	6	
131985	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.3	0	0	0	0	0	6	0	6	
131986	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.4	0	0	0	0	0	6	0	6	
131987	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.5	0	0	0	0	0	0	0	0	
131988	2019	final	Hyderabad	2	Chennai Super Kings	Mumbai Indians	spinner	17.6	0	0	0	0	0	1	0	1	

45378 rows × 19 columns

In [27]:

```
playoff = ['3rd Place Playoff', 'eliminator', 'final', 'semi-final1', 'semi-final2', 'semi-final']
df.loc[df.match_id.isin(playoff), 'match_id'] = 'playoffs'
```

In [28]:
```
df.loc[df.match_id=='playoffs']
```

Out[28]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismi
12801	2008	playoffs	Mumbai	1	Rajasthan Royals	Delhi Capitals	pacer	0.1	0	0	0	0	0	0	0	0	
12802	2008	playoffs	Mumbai	1	Rajasthan Royals	Delhi Capitals	pacer	0.2	1	0	0	0	0	0	1	1	
12803	2008	playoffs	Mumbai	1	Rajasthan Royals	Delhi Capitals	pacer	0.2	0	0	0	0	0	0	0	0	
12804	2008	playoffs	Mumbai	1	Rajasthan Royals	Delhi Capitals	pacer	0.3	0	0	0	0	0	0	0	0	
12805	2008	playoffs	Mumbai	1	Rajasthan Royals	Delhi Capitals	pacer	0.4	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
131996	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.2	0	0	0	0	0	1	0	1	
131997	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.3	0	0	0	0	0	2	0	2	
131998	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.4	0	0	0	0	0	1	0	1	
131999	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.5	0	0	0	0	0	2	0	2	
132000	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.6	0	0	0	0	0	0	0	0	

9186 rows × 19 columns

In [29]:
```
p = ['playoffs']
df.loc[~df.match_id.isin(p),'match_id'] = 'round-robin'
```

```
In [30]: df.loc[df.match_id=='round-robin']
```

Out[30]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs	dismis
0	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.1	0	0	1	0	0	0	1	1	
1	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.2	0	0	0	0	0	0	0	0	
2	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	1	0	0	0	0	0	1	1	
3	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	0	0	0	0	0	0	0	0	
4	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.4	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
131021	2019	round-robin	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	pacer	15.3	0	0	0	0	0	0	0	0	
131022	2019	round-robin	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	pacer	15.4	0	0	0	0	0	4	0	4	
131023	2019	round-robin	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	pacer	15.5	0	0	0	0	0	1	0	1	
131024	2019	round-robin	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	pacer	15.6	0	0	0	0	0	1	0	1	
131025	2019	round-robin	Mumbai	2	Mumbai Indians	Kolkata Knight Riders	pacer	16.1	0	0	0	0	0	6	0	6	

122815 rows × 19 columns

In [31]:
```
df.rename(columns={"match_id": "match_type", "bowler": "bowler_type"})
```

Out[31]:

	season	match_type	venue	inning	batting_team	bowling_team	bowler_type	ball	wide_runs	bye_runs	legbye_runs	noball_runs	penalty_runs	batsman_runs	extra_runs	total_runs
0	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.1	0	0	1	0	0	0	1	1
1	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.2	0	0	0	0	0	0	0	0
2	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	1	0	0	0	0	0	1	1
3	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	0	0	0	0	0	0	0	0
4	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.4	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
131996	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.2	0	0	0	0	0	1	0	1
131997	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.3	0	0	0	0	0	2	0	2
131998	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.4	0	0	0	0	0	1	0	1
131999	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.5	0	0	0	0	0	2	0	2
132000	2019	playoffs	Hyderabad	2	Chennai Super Kings	Mumbai Indians	pacer	19.6	0	0	0	0	0	0	0	0

132001 rows × 19 columns

In [32]:
```
df.drop(labels='penalty_runs', axis=1, inplace=True)
df.drop(labels=1614, axis=0, inplace=True)
```

In [33]:
```
df.head()
```

Out[33]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	batsman_runs	extra_runs	total_runs	dismissal_kind	wickets	c
0	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.1	0	0	1	0	0	1	1		0	
1	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.2	0	0	0	0	0	0	0		0	
2	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	1	0	0	0	0	1	1		0	
3	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	0	0	0	0	0	0	0		0	
4	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.4	0	0	0	0	0	0	0		0	r

In [34]:
```
import re
def lower_comm(x):
    return x.lower()
df['commentary'] = df['commentary'].apply(lower_comm)
```

In [35]:
```
full_toss_length = ['full toss', 'beamer', 'full-toss', 'fulltoss']
yorker_length = ['yorker', 'yorker length', 'yorker-ish', 'blockhole', 'block-hole', 'block hole', 'toe-crusher', 'toe crusher']
full_length = ['full', 'full and straight', 'full length', 'fullish', 'full-ish', 'fuller', 'very full', 'full ball', 'full delivery', 'forward',
'front foot', 'flicked off the pads', 'glanced off the pads', 'flicks off the pad', 'off the pads', 'off the pad',
               'clips', 'clipped', 'pads', 'on the pad', 'into the pads', 'cover drive', 'front', 'beaten', 'beaten on the drive','pitches it up'
, 'full on', 'slog', 'slogs', 'flat and quick',
               'onto the pads', 'aimed at leg stump', 'aimed at leg-stump', 'pitched at leg stump', 'floated up', 'floats', 'floating', 'flights'
, 'flight', 'inside out', 'in the slot', 'slot delivery', 'long on', 'long-on',
               'flighted', 'flighting', 'tossed up', 'overpitched', 'overpitching', 'overpitches', 'pitched up', 'half-volley', 'half volley', 'l
oopy', 'looping', 'reverse sweep', 'down the ground',
               'reverse-sweep', 'sweep', 'sweeping', 'paddled', 'paddle', 'paddles', 'slog swept', 'slog-sweep', 'off-drive', 'straight-drive',
'worked', 'works', 'meeting the pitch', 'lunges forward', 'cover-drive', 'cover drive',
               'off drive', 'slog-sweeps', 'straight drive', 'on-drive', 'on drive', 'helicopter shot', 'helicopter', 'over cover', 'over extra c
over', 'pitch of the ball', 'to the pitch', 'slower full', 'slower, full']
good_length = ['length ball', 'good length', 'length delivery', 'at a length', ', length,', 'length', 'peach', 'corridor of uncertainty', 'punche
d', 'googly', 'cramped', 'cramps', 'cramping', 'thigh pad', 'cramp',
               'goodish', 'good-ish', 'nudges', 'driven on the up', 'slower length ball', 'slower, good length', 'on the up', 'length outside',
'length on', 'tight length']
short_length = ['short', 'short ball', 'short delivery', 'bouncer', 'pulled', 'hooked', 'pulling', 'hooking', 'pull', 'hook',
                'back of a length', 'back of length', 'back of good length', 'back foot', 'back-foot', 'heaved', 'hoicked',
                'short of good length', 'short length', 'shortish', 'short-ish', 'cuts', 'cut', 'cutting it', 'cutting the', 'dragged back', 'dra
gs his length back',
                'back-of-a-length', 'holding back the length']
```

```
In [36]: conditions  = [df.commentary.str.contains('|'.join(full_toss_length)), df.commentary.str.contains('|'.join(yorker_length)), df.commentary.str.contains('|'.join(full_length)), df.commentary.str.contains('|'.join(good_length)), df.commentary.str.contains('|'.join(short_length))]
         choices = ["full toss", "yorker", "full", "good", "short"]

         df["ball_length"] = np.select(conditions, choices, default=np.nan)
         df.head(50)
```

Out[36]:

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	batsman_runs	extra_runs	total_runs	dismissal_kind	wickets
0	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.1	0	0	1	0	0	1	1		0
1	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.2	0	0	0	0	0	0	0		0
2	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	1	0	0	0	0	1	1		0
3	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	0	0	0	0	0	0	0		0
4	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.4	0	0	0	0	0	0	0		0
5	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.5	0	0	0	0	0	0	0		0
6	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.6	0	0	1	0	0	1	1		0
7	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	1.1	0	0	0	0	0	0	0		0
8	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	1.2	0	0	0	0	4	0	4		0
9	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	1.3	0	0	0	0	4	0	4		0
10	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	1.4	0	0	0	0	6	0	6		0
11	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	1.5	0	0	0	0	4	0	4		0
12	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	1.6	0	0	0	0	0	0	0		0
13	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	2.1	0	0	0	0	0	0	0		0
14	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	2.2	0	0	0	0	0	0	0		0
15	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	2.3	0	0	1	0	0	1	1		0
16	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	2.4	0	0	0	0	4	0	4		0
17	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	2.5	0	0	0	0	1	0	1		0
18	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	2.6	0	0	0	0	0	0	0		0
19	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	3.1	5	0	0	0	0	5	5		0
20	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	3.1	0	0	0	0	6	0	6		0

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	batsman_runs	extra_runs	total_runs	dismissal_kind	wickets
21	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	3.2	0	0	1	0	0	1	1		0
22	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	3.3	0	0	0	0	4	0	4		0
23	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	3.4	0	0	0	0	0	0	0		0
24	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	3.5	0	0	0	0	1	0	1		0
25	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	3.6	0	0	0	0	6	0	6		0
26	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	4.1	0	0	0	0	4	0	4		0
27	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	4.2	0	0	0	0	1	0	1		0
28	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	4.3	0	0	0	0	4	0	4		0
29	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	4.4	0	0	0	0	0	0	0		0
30	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	4.5	0	0	0	0	1	0	1		0
31	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	4.6	0	0	0	0	0	0	0		0
32	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	5.1	0	0	0	0	1	0	1		0
33	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	5.2	0	0	0	0	0	0	0	caught	1
34	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	5.3	0	0	0	0	0	0	0		1
35	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	5.4	0	0	0	0	0	0	0		1
36	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	5.5	0	0	0	0	0	0	0		1
37	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	5.6	0	0	0	0	0	0	0		1
38	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	6.1	0	0	0	0	1	0	1		1
39	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	6.2	0	0	0	0	1	0	1		1
40	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	6.3	0	0	0	0	1	0	1		1
41	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	6.4	0	0	0	0	2	0	2		1
42	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	6.5	0	0	0	0	1	0	1		1

	season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	batsman_runs	extra_runs	total_runs	dismissal_kind	wickets
43	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	6.6	0	0	0	0	1	0	1		1
44	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	7.1	0	0	0	0	0	0	0		1
45	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	7.2	0	0	0	0	1	0	1		1
46	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	7.3	0	0	0	0	1	0	1		1
47	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	7.4	0	0	0	0	1	0	1		1
48	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	7.5	0	0	0	0	1	0	1		1
49	2008	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	7.6	0	0	0	0	1	0	1		1

In [37]:
```
df['batsman_runs'] = df['batsman_runs'].astype('int64')
df['total_runs'] = df['total_runs'].astype('int64')
df['extra_runs'] = df['extra_runs'].astype('int64')
df['noball_runs'] = df['noball_runs'].astype('int64')
df['legbye_runs'] = df['legbye_runs'].astype('int64')
df['bye_runs'] = df['bye_runs'].astype('int64')
df['wide_runs'] = df['wide_runs'].astype('int64')

def event_pred(row):
    if row['batsman_runs']== 0 and row['extra_runs']== 0 and row['total_runs'] == 0 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return '0 Runs'
    if row['batsman_runs']==1 and row['extra_runs']==0 and row['total_runs'] == 1 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return '1 Run'
    if row['batsman_runs']==2 and row['extra_runs']==0 and row['total_runs'] == 2 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return '2 Runs'
    if row['batsman_runs']==3 and row['extra_runs']==0 and row['total_runs'] == 3 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return '3 Runs'
    if row['batsman_runs']==4 and row['extra_runs']==0 and row['total_runs'] == 4 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return '4 Runs!'
    if row['batsman_runs']==5 and row['extra_runs']==0 and row['total_runs'] == 5 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return '5 Runs'
    if row['batsman_runs']==6 and row['extra_runs']==0 and row['total_runs'] == 6 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return '6 Runs!'
    if row['wide_runs']>0 and row['extra_runs']>0 and row['total_runs']>0 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return 'Extras (wide)'
    if row['bye_runs']>0 and row['extra_runs']>0 and row['total_runs']>0 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return 'Extras (bye)'
    if row['legbye_runs']>0 and row['extra_runs']>0 and row['total_runs']>0 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return 'Extras (legbye)'
    if row['noball_runs']>0 and row['extra_runs']>0 and row['batsman_runs']==0 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return 'Extras (noball) + 0 Batsman Runs'
    if row['noball_runs']>0 and row['extra_runs']>0 and row['batsman_runs']==1 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return 'Extras (noball) + 1 Batsman Run'
    if row['noball_runs']>0 and row['extra_runs']>0 and row['batsman_runs']==2 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return 'Extras (noball) + 2 Batsman Runs'
    if row['noball_runs']>0 and row['extra_runs']>0 and row['batsman_runs']==3 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return 'Extras (noball) + 3 Batsman Runs'
    if row['noball_runs']>0 and row['extra_runs']>0 and row['batsman_runs']==4 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return 'Extras (noball) + 4 Batsman Runs!'
    if row['noball_runs']>0 and row['extra_runs']>0 and row['batsman_runs']==6 and row['dismissal_kind']!='bowled' and row['dismissal_kind']!='caught' and row['dismissal_kind']!='caught and bowled' and row['dismissal_kind']!='hit wicket' and row['dismissal_kind']!='lbw' and row['dismissal_kind']!='obstructing the field' and row['dismissal_kind']!='retired hurt' and row['dismissal_kind']!='run out' and row['dismissal_kind']!='stumped':
        return 'Extras (noball) + 6 Batsman Runs!'
    if row['dismissal_kind']=='bowled':
        return 'WICKET (Bowled)!'
    if row['dismissal_kind']=='caught':
        return 'WICKET (Caught)!'
    if row['dismissal_kind']=='caught and bowled':
        return 'WICKET (Caught & Bowled)!'
    if row['dismissal_kind']=='hit wicket':
        return 'WICKET (Hit Wicket)!'
    if row['dismissal_kind']=='lbw':
        return 'WICKET (LBW - Leg Before Wicket)!'
    if row['dismissal_kind']=='obstructing the field':
        return 'WICKET (Obstructing the Field)!'
    if row['dismissal_kind']=='retired hurt':
        return 'Retired Hurt'
    if row['dismissal_kind']=='run out':
        return 'WICKET (Run Out)!'
    if row['dismissal_kind']=='stumped':
        return 'WICKET (Stumped)!'
```

In [38]:
```
df['event'] = df.apply(lambda row: event_pred(row), axis=1)
```

In [39]:
```
df.loc[df.event.isnull()]
```

Out[39]:

season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	batsman_runs	extra_runs	total_runs	dismissal_kind	wickets	comn

In [40]:
```
df.loc[(df.ball_length=='nan') & (df.bowler=='spinner'), 'ball_length'] = 'good'
```

In [41]:
```
df.loc[(df.ball_length=='nan') & (df.bowler=='spinner')]
```

Out[41]:

season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	batsman_runs	extra_runs	total_runs	dismissal_kind	wickets	comn

In [42]:
```
df.loc[(df.ball_length=='nan') & (df.event=='Extras (wide)'), 'ball_length'] = 'good'
```

In [43]:
```
df.loc[(df.ball_length=='nan') & (df.event=='Extras (legbye)'), 'ball_length'] = 'full'
```

In [44]:
```
df.loc[(df.ball_length=='nan') & (df.event=='0 Runs') | (df.event=='1 Run') | (df.event=='2 Runs') | (df.event=='3 Runs'), 'ball_length'] = 'good'
```

In [45]:
```
df.loc[(df.ball_length=='nan') & (df.event=='4 Runs!') |  (df.event=='6 Runs!'), 'ball_length'] = 'full'
```

In [46]:
```
df.loc[(df.ball_length=='nan') & (df.event=='5 Runs') | (df.event=='Extras (bye)') | (df.event=='Extras (noball) + 0 Batsman Runs'), 'ball_length'] = 'good'
```

In [47]:
```
df.loc[(df.ball_length=='nan') & (df.event=='Extras (noball) + 6 Batsman Runs!') | (df.event=='Extras (noball) + 4 Batsman Runs!') | (df.event=='Extras (noball) + 1 Batsman Run') | (df.event=='Extras (noball) + 2 Batsman Runs'), 'ball_length'] = 'full'
```

In [48]:
```
df.loc[(df.ball_length=='nan') & (df.event=='WICKET (Caught)!') | (df.event=='WICKET (Run Out)!') | (df.event=='WICKET (Hit Wicket)!'), 'ball_length'] = 'good'
df.loc[(df.ball_length=='nan') & (df.event=='WICKET (Bowled)!') | (df.event=='WICKET (LBW - Leg Before Wicket)!') | (df.event=='WICKET (Caught & Bowled)!') | (df.event=='WICKET (Stumped)!'), 'ball_length'] = 'full'
df.loc[(df.ball_length=='nan')]
```

Out[48]:

season	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wide_runs	bye_runs	legbye_runs	noball_runs	batsman_runs	extra_runs	total_runs	dismissal_kind	wickets	comn

In [49]:
```
df.dtypes
```

Out[49]:
```
season              int64
match_id           object
venue              object
inning              int64
batting_team       object
bowling_team       object
bowler             object
ball              float64
wide_runs           int64
bye_runs            int64
legbye_runs         int64
noball_runs         int64
batsman_runs        int64
extra_runs          int64
total_runs          int64
dismissal_kind     object
wickets             int64
commentary         object
ball_length        object
event              object
dtype: object
```

In [50]:

```
df['match_id'] = df['match_id'].astype('category')
df['venue'] = df['venue'].astype('category')
df['inning'] = df['inning'].astype('category')
df['batting_team'] = df['batting_team'].astype('category')
df['bowling_team'] = df['bowling_team'].astype('category')
df['bowler'] = df['bowler'].astype('category')
df['ball_length'] = df['ball_length'].astype('category')

df['event'] = df['event'].astype('object')

df['ball'] = df['ball'].astype('float64')

df['wickets'] = df['wickets'].astype('int64')

df.drop(labels='commentary', axis=1, inplace=True)
df.drop(labels='wide_runs', axis=1, inplace=True)
df.drop(labels='bye_runs', axis=1, inplace=True)
df.drop(labels='legbye_runs', axis=1, inplace=True)
df.drop(labels='noball_runs', axis=1, inplace=True)
df.drop(labels='batsman_runs', axis=1, inplace=True)
df.drop(labels='extra_runs', axis=1, inplace=True)
df.drop(labels='total_runs', axis=1, inplace=True)
df.drop(labels='dismissal_kind', axis=1, inplace=True)
df.drop(labels='season', axis=1, inplace=True)

df.head()
```

Out[50]:

	match_id	venue	inning	batting_team	bowling_team	bowler	ball	wickets	ball_length	event
0	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.1	0	full	Extras (legbye)
1	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.2	0	good	0 Runs
2	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	0	good	Extras (wide)
3	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.3	0	good	0 Runs
4	round-robin	Bangalore	1	Kolkata Knight Riders	Royal Challengers Bangalore	pacer	0.4	0	good	0 Runs

In [51]:

```
df.dtypes
```

Out[51]:

```
match_id        category
venue           category
inning          category
batting_team    category
bowling_team    category
bowler          category
ball             float64
wickets            int64
ball_length     category
event             object
dtype: object
```

In [52]:

```
df.describe()
```

Out[52]:

	ball	wickets
count	132000.000000	132000.00000
mean	9.547066	2.37997
std	5.675279	2.06105
min	0.100000	0.00000
25%	4.500000	1.00000
50%	9.400000	2.00000
75%	14.400000	4.00000
max	19.600000	10.00000

In [53]:

```
from sklearn.preprocessing import minmax_scale
df[['ball','wickets']] = minmax_scale(df[['ball','wickets']])

encoded_df = pd.get_dummies(data=df, columns=['match_id', 'venue', 'inning', 'batting_team', 'bowling_team', 'bowler', 'ball_length'])
```

In [54]:

```
encoded_df.columns
```

Out[54]:

```
Index(['ball', 'wickets', 'event', 'match_id_playoffs', 'match_id_round-robin',
       'venue_Ahmedabad', 'venue_Bangalore', 'venue_Chennai', 'venue_Cuttack',
       'venue_Delhi', 'venue_Dharamsala', 'venue_Hyderabad', 'venue_Indore',
       'venue_Kolkata', 'venue_Mohali', 'venue_Mumbai', 'venue_Nagpur',
       'venue_Pune', 'venue_Raipur', 'venue_Rajasthan', 'venue_Ranchi',
       'venue_Visakhapatnam', 'inning_1', 'inning_2',
       'batting_team_Chennai Super Kings', 'batting_team_Deccan Chargers',
       'batting_team_Delhi Capitals', 'batting_team_Kings XI Punjab',
       'batting_team_Kolkata Knight Riders', 'batting_team_Mumbai Indians',
       'batting_team_Rajasthan Royals',
       'batting_team_Royal Challengers Bangalore',
       'batting_team_Sunrisers Hyderabad', 'bowling_team_Chennai Super Kings',
       'bowling_team_Deccan Chargers', 'bowling_team_Delhi Capitals',
       'bowling_team_Kings XI Punjab', 'bowling_team_Kolkata Knight Riders',
       'bowling_team_Mumbai Indians', 'bowling_team_Rajasthan Royals',
       'bowling_team_Royal Challengers Bangalore',
       'bowling_team_Sunrisers Hyderabad', 'bowler_pacer', 'bowler_spinner',
       'ball_length_full', 'ball_length_full toss', 'ball_length_good',
       'ball_length_short', 'ball_length_yorker'],
      dtype='object')
```

In [55]:

```
encoded_df.shape
```

Out[55]:

```
(132000, 49)
```

In [56]:
```
encoded_df = encoded_df[['ball', 'wickets', 'match_id_playoffs',
       'match_id_round-robin', 'venue_Ahmedabad', 'venue_Bangalore',
       'venue_Chennai', 'venue_Cuttack', 'venue_Delhi',
       'venue_Dharamsala', 'venue_Hyderabad', 'venue_Indore', 'venue_Kolkata',
       'venue_Mohali', 'venue_Mumbai', 'venue_Nagpur', 'venue_Pune',
       'venue_Raipur', 'venue_Rajasthan', 'venue_Ranchi',
       'venue_Visakhapatnam', 'inning_1', 'inning_2',
       'batting_team_Chennai Super Kings', 'batting_team_Deccan Chargers',
       'batting_team_Delhi Capitals', 'batting_team_Kings XI Punjab',
       'batting_team_Kolkata Knight Riders', 'batting_team_Mumbai Indians',
       'batting_team_Rajasthan Royals',
       'batting_team_Royal Challengers Bangalore',
       'batting_team_Sunrisers Hyderabad', 'bowling_team_Chennai Super Kings',
       'bowling_team_Deccan Chargers', 'bowling_team_Delhi Capitals',
       'bowling_team_Kings XI Punjab', 'bowling_team_Kolkata Knight Riders',
       'bowling_team_Mumbai Indians', 'bowling_team_Rajasthan Royals',
       'bowling_team_Royal Challengers Bangalore',
       'bowling_team_Sunrisers Hyderabad', 'bowler_pacer', 'bowler_spinner',
       'ball_length_full', 'ball_length_full toss', 'ball_length_good',
       'ball_length_short', 'ball_length_yorker', 'event']]
```

In [57]:
```
encoded_df['event'].unique()
```

Out[57]:
```
array(['Extras (legbye)', '0 Runs', 'Extras (wide)', '4 Runs!', '6 Runs!',
       '1 Run', 'WICKET (Caught)!', '2 Runs', 'Extras (bye)',
       'WICKET (Bowled)!', 'WICKET (Run Out)!',
       'Extras (noball) + 4 Batsman Runs!',
       'WICKET (LBW - Leg Before Wicket)!', 'Retired Hurt',
       'WICKET (Stumped)!', 'WICKET (Caught & Bowled)!',
       'Extras (noball) + 0 Batsman Runs', '5 Runs', '3 Runs',
       'Extras (noball) + 1 Batsman Run',
       'Extras (noball) + 6 Batsman Runs!', 'WICKET (Hit Wicket)!',
       'Extras (noball) + 2 Batsman Runs',
       'WICKET (Obstructing the Field)!'], dtype=object)
```

In [58]:
```
X = encoded_df.iloc[:,0:48].values
X.shape
```

Out[58]:
```
(132000, 48)
```

In [59]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
encoded_df['event'] = encoder.fit_transform(encoded_df['event'])
Y = pd.get_dummies(encoded_df['event']).values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,Y,test_size=0.213530303,random_state=0)
```

In [60]:
```
import tensorflow as tf
from tensorflow.keras import models, layers, optimizers
```

In [61]:
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(512, input_shape=(48,), activation="relu"),
    tf.keras.layers.Dropout(0.4),
    tf.keras.layers.Dense(256, activation="relu"),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(24, activation="softmax")
])

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense (Dense)                (None, 512)               25088     
_________________________________________________________________
dropout (Dropout)            (None, 512)               0         
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328    
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0         
_________________________________________________________________
dense_2 (Dense)              (None, 24)                6168      
=================================================================
Total params: 162,584
Trainable params: 162,584
Non-trainable params: 0
_________________________________________________________________
```

In [62]:
```
model.compile(loss=tf.keras.losses.CategoricalCrossentropy(),
              optimizer=optimizers.Adam(lr=0.0001),
              metrics=['accuracy'])
```

In [63]:

```
X_train = np.asarray(X_train)
y_train = np.asarray(y_train)
model.fit(X_train,y_train,batch_size=16,epochs=50)
```

```
Epoch 1/50
6489/6489 [==============================] - 18s 3ms/step - loss: 1.4228 - accuracy: 0.5250
Epoch 2/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.3473 - accuracy: 0.5361
Epoch 3/50
6489/6489 [==============================] - 18s 3ms/step - loss: 1.3327 - accuracy: 0.5396
Epoch 4/50
6489/6489 [==============================] - 16s 3ms/step - loss: 1.3254 - accuracy: 0.5414
Epoch 5/50
6489/6489 [==============================] - 14s 2ms/step - loss: 1.3199 - accuracy: 0.5431
Epoch 6/50
6489/6489 [==============================] - 18s 3ms/step - loss: 1.3158 - accuracy: 0.5438
Epoch 7/50
6489/6489 [==============================] - 14s 2ms/step - loss: 1.3130 - accuracy: 0.5444
Epoch 8/50
6489/6489 [==============================] - 16s 2ms/step - loss: 1.3104 - accuracy: 0.5454
Epoch 9/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.3080 - accuracy: 0.5457
Epoch 10/50
6489/6489 [==============================] - 14s 2ms/step - loss: 1.3055 - accuracy: 0.5465
Epoch 11/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.3036 - accuracy: 0.5470
Epoch 12/50
6489/6489 [==============================] - 18s 3ms/step - loss: 1.3018 - accuracy: 0.5474
Epoch 13/50
6489/6489 [==============================] - 19s 3ms/step - loss: 1.3010 - accuracy: 0.5475
Epoch 14/50
6489/6489 [==============================] - 22s 3ms/step - loss: 1.2973 - accuracy: 0.5483
Epoch 15/50
6489/6489 [==============================] - 20s 3ms/step - loss: 1.2976 - accuracy: 0.5484 0s - loss: 1.2978 - accuracy
Epoch 16/50
6489/6489 [==============================] - 14s 2ms/step - loss: 1.2948 - accuracy: 0.5489
Epoch 17/50
6489/6489 [==============================] - 14s 2ms/step - loss: 1.2939 - accuracy: 0.5474 0s - loss: 1.2936 - accuracy: 0.
Epoch 18/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.2929 - accuracy: 0.5486
Epoch 19/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.2908 - accuracy: 0.5491
Epoch 20/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.2893 - accuracy: 0.5498
Epoch 21/50
6489/6489 [==============================] - 16s 2ms/step - loss: 1.2886 - accuracy: 0.5497
Epoch 22/50
6489/6489 [==============================] - 14s 2ms/step - loss: 1.2869 - accuracy: 0.5498
Epoch 23/50
6489/6489 [==============================] - 14s 2ms/step - loss: 1.2861 - accuracy: 0.5494
Epoch 24/50
6489/6489 [==============================] - 17s 3ms/step - loss: 1.2849 - accuracy: 0.5503 0s - los
Epoch 25/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.2837 - accuracy: 0.5507
Epoch 26/50
6489/6489 [==============================] - 17s 3ms/step - loss: 1.2824 - accuracy: 0.5496
Epoch 27/50
6489/6489 [==============================] - 17s 3ms/step - loss: 1.2820 - accuracy: 0.5513
Epoch 28/50
6489/6489 [==============================] - 16s 2ms/step - loss: 1.2809 - accuracy: 0.5510
Epoch 29/50
6489/6489 [==============================] - 18s 3ms/step - loss: 1.2798 - accuracy: 0.5514
Epoch 30/50
6489/6489 [==============================] - 14s 2ms/step - loss: 1.2782 - accuracy: 0.5510
Epoch 31/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.2775 - accuracy: 0.5515
Epoch 32/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.2764 - accuracy: 0.5518
Epoch 33/50
6489/6489 [==============================] - 16s 2ms/step - loss: 1.2765 - accuracy: 0.5516
Epoch 34/50
6489/6489 [==============================] - 17s 3ms/step - loss: 1.2737 - accuracy: 0.5524
Epoch 35/50
6489/6489 [==============================] - 14s 2ms/step - loss: 1.2746 - accuracy: 0.5511
Epoch 36/50
6489/6489 [==============================] - 16s 2ms/step - loss: 1.2740 - accuracy: 0.5515
Epoch 37/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.2729 - accuracy: 0.5527
Epoch 38/50
6489/6489 [==============================] - 16s 2ms/step - loss: 1.2724 - accuracy: 0.5531
Epoch 39/50
6489/6489 [==============================] - 20s 3ms/step - loss: 1.2705 - accuracy: 0.5530
Epoch 40/50
6489/6489 [==============================] - 18s 3ms/step - loss: 1.2699 - accuracy: 0.5524
Epoch 41/50
6489/6489 [==============================] - 19s 3ms/step - loss: 1.2696 - accuracy: 0.5524
Epoch 42/50
6489/6489 [==============================] - 18s 3ms/step - loss: 1.2682 - accuracy: 0.5527
Epoch 43/50
6489/6489 [==============================] - 21s 3ms/step - loss: 1.2677 - accuracy: 0.5527
Epoch 44/50
6489/6489 [==============================] - 19s 3ms/step - loss: 1.2669 - accuracy: 0.5537
Epoch 45/50
6489/6489 [==============================] - 21s 3ms/step - loss: 1.2665 - accuracy: 0.5542
Epoch 46/50
6489/6489 [==============================] - 18s 3ms/step - loss: 1.2654 - accuracy: 0.5530
Epoch 47/50
6489/6489 [==============================] - 17s 3ms/step - loss: 1.2647 - accuracy: 0.5537
Epoch 48/50
6489/6489 [==============================] - 15s 2ms/step - loss: 1.2637 - accuracy: 0.5534
Epoch 49/50
6489/6489 [==============================] - 20s 3ms/step - loss: 1.2641 - accuracy: 0.5539
Epoch 50/50
6489/6489 [==============================] - 18s 3ms/step - loss: 1.2629 - accuracy: 0.5542
```

```
Out[63]: <tensorflow.python.keras.callbacks.History at 0x7fa20571cd00>
```

In [64]:

```
y_pred=model.predict(X_test)
y_test_class=np.argmax(y_test,axis=1)
y_pred_class=np.argmax(y_pred,axis=1)
```

In [65]:

```
from sklearn.metrics import classification_report,confusion_matrix
print(classification_report(y_test_class,y_pred_class))
print(confusion_matrix(y_test_class,y_pred_class))
```

```
              precision    recall  f1-score   support

           0       0.49      0.55      0.52      8468
           1       0.61      0.95      0.74     10312
           2       0.00      0.00      0.00      1735
           3       0.00      0.00      0.00        92
           4       0.42      0.13      0.20      3357
           5       0.00      0.00      0.00         6
           6       0.37      0.33      0.35      1365
           7       0.00      0.00      0.00        59
           8       0.00      0.00      0.00       443
           9       0.00      0.00      0.00        36
          10       0.00      0.00      0.00        46
          11       0.00      0.00      0.00        14
          12       0.00      0.00      0.00        11
          13       0.00      0.00      0.00         9
          14       0.00      0.00      0.00       858
          15       0.00      0.00      0.00         2
          16       1.00      0.00      0.01       248
          17       0.00      0.00      0.00        32
          18       0.29      0.01      0.01       840
          19       0.00      0.00      0.00         1
          20       0.00      0.00      0.00        75
          21       0.00      0.00      0.00         1
          22       0.00      0.00      0.00       126
          23       0.00      0.00      0.00        50

    accuracy                           0.55     28186
   macro avg       0.13      0.08      0.08     28186
weighted avg       0.45      0.55      0.47     28186

[[4689 3155    0    0  347    0  270    0    0    0    0    0    0    0
     0    0    0    0    7    0    0    0    0    0]
 [ 482 9830    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [  76 1659    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [  16   76    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [2103  532    0    0  449    0  272    0    0    0    0    0    0    0
     0    0    0    0    1    0    0    0    0    0]
 [   2    4    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [ 792    0    0    0  126    0  444    0    0    0    0    0    0    0
     0    0    0    0    3    0    0    0    0    0]
 [   3   56    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [ 278   77    0    0   41    0   46    0    0    0    0    0    0    0
     0    0    0    0    1    0    0    0    0    0]
 [   5   31    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [  27    0    0    0    4    0   15    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [   9    0    0    0    1    0    4    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [   9    0    0    0    0    0    2    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [   2    1    0    0    4    0    2    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [ 355  429    0    0   38    0   36    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [   2    0    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [ 168   52    0    0    8    0   16    0    0    0    0    0    0    0
     0    0    1    0    3    0    0    0    0    0]
 [  27    0    0    0    1    0    4    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [ 501  201    0    0   56    0   76    0    0    0    0    0    0    0
     0    0    0    0    6    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [  62    0    0    0    5    0    8    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [   0    1    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [  13  113    0    0    0    0    0    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]
 [  41    0    0    0    0    0    9    0    0    0    0    0    0    0
     0    0    0    0    0    0    0    0    0    0]]
```

```
/Applications/anaconda3/lib/python3.8/site-packages/sklearn/metrics/_classification.py:1221: UndefinedMetricWarning: Precision and F-score are
ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]: