In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.preprocessing import LabelEncoder
```

In [2]:
```
df1 = pd.read_csv('C:/Users/Rog Strix/Downloads/2022DA04050/training_dataset/Ball.csv'
df1.head()
```

Out[2]:

	ID	innings	overs	ballnumber	batter	bowler	non-striker	extra_type	batsman_run	extras
0	1312200	1	0	1	YBK Jaiswal	Mohammed Shami	JC Buttler	NaN	0	
1	1312200	1	0	2	YBK Jaiswal	Mohammed Shami	JC Buttler	legbyes	0	
2	1312200	1	0	3	JC Buttler	Mohammed Shami	YBK Jaiswal	NaN	1	
3	1312200	1	0	4	YBK Jaiswal	Mohammed Shami	JC Buttler	NaN	0	
4	1312200	1	0	5	YBK Jaiswal	Mohammed Shami	JC Buttler	NaN	0	

In [3]:
```
df2 = pd.read_csv('C:/Users/Rog Strix/Downloads/2022DA04050/training_dataset/Result.cs
df2.head()
```

Out[3]:

	ID	City	Date	Season	MatchNumber	Team1	Team2	Venue	TossWinne
0	1312200	Ahmedabad	2022-05-29	2022	Final	Rajasthan Royals	Gujarat Titans	Narendra Modi Stadium, Ahmedabad	Rajastha Royal
1	1312199	Ahmedabad	2022-05-27	2022	Qualifier 2	Royal Challengers Bangalore	Rajasthan Royals	Narendra Modi Stadium, Ahmedabad	Rajastha Royal
2	1312198	Kolkata	2022-05-25	2022	Eliminator	Royal Challengers Bangalore	Lucknow Super Giants	Eden Gardens, Kolkata	Lucknov Supe Giant
3	1312197	Kolkata	2022-05-24	2022	Qualifier 1	Rajasthan Royals	Gujarat Titans	Eden Gardens, Kolkata	Gujara Titan
4	1304116	Mumbai	2022-05-22	2022	70	Sunrisers Hyderabad	Punjab Kings	Wankhede Stadium, Mumbai	Sunriser Hyderaba

In [4]:
```
df3 = pd.merge(df1,df2,left_on='ID',right_on='ID',how='outer')
```

In [5]:
```
def get_bowling_team(row):
    if row['TossWinner'] == row['BattingTeam']:
        # if the team that won the toss is batting first, the other team will bowl fir
        return row['Team2']
    else:
        # if the team that won the toss is bowling first, the other team will bat firs
        return row['Team1']

# apply the function to each row of the DataFrame to create a new column called "bowli
df3['bowling_team'] = df3.apply(get_bowling_team, axis=1)
```

In [6]:
```
df = df3.copy()
```

In [7]:
```
df_sum_first_six_overs = df[df["overs"] <= 6].groupby(["ID", "innings","Venue","Battir

# rename the column containing the sum of runs
df_sum_first_six_overs = df_sum_first_six_overs.rename(columns={"runs": "sum_first_six
```

In [8]:
```
df_sum_first_six_overs.rename(columns = {'total_run':'total_runs'}, inplace = True)
```

In [9]:
```
df4 = pd.DataFrame(df_sum_first_six_overs)
```

In [10]:
```
df4
```

Out[10]:

	ID	innings	Venue	BattingTeam	bowling_team	MatchNumber	total_runs
0	335982	1	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	1	68
1	335982	2	M Chinnaswamy Stadium	Royal Challengers Bangalore	Kolkata Knight Riders	1	33
2	335983	1	Punjab Cricket Association Stadium, Mohali	Chennai Super Kings	Chennai Super Kings	2	62
3	335983	2	Punjab Cricket Association Stadium, Mohali	Kings XI Punjab	Kings XI Punjab	2	69
4	335984	1	Feroz Shah Kotla	Rajasthan Royals	Rajasthan Royals	3	44
...	...	...	...	...	...	...	...
1923	1312198	2	Eden Gardens, Kolkata	Lucknow Super Giants	Lucknow Super Giants	Eliminator	67
1924	1312199	1	Narendra Modi Stadium, Ahmedabad	Royal Challengers Bangalore	Royal Challengers Bangalore	Qualifier 2	52
1925	1312199	2	Narendra Modi Stadium, Ahmedabad	Rajasthan Royals	Rajasthan Royals	Qualifier 2	77
1926	1312200	1	Narendra Modi Stadium, Ahmedabad	Rajasthan Royals	Gujarat Titans	Final	54
1927	1312200	2	Narendra Modi Stadium, Ahmedabad	Gujarat Titans	Rajasthan Royals	Final	35

1928 rows × 7 columns

In [11]:

```
def remove_brackets(x):
    return x.strip('[]')

df['Team1Players'] = df['Team1Players'].apply(remove_brackets)
# create Team1players_num series
for i in df["Team1Players"]:
    i = list[i]

df['Team2Players'] = df['Team2Players'].apply(remove_brackets)
# create Team1players_num series
for i in df["Team2Players"]:
    i = list[i]

df['Team1Players'] = df['Team1Players'].apply(lambda x: [y.replace("'", "") for y in x
df['Team2Players'] = df['Team2Players'].apply(lambda x: [y.replace("'", "") for y in x
```

In [12]:
```
df['Team1Players'] = df['Team1Players'].apply(tuple)
df['Team2Players'] = df['Team2Players'].apply(tuple)

# select only the desired columns and create a new DataFrame
new_df = df.loc[:, ['ID', 'Team1Players', 'Team2Players', 'MatchNumber', 'innings','Ve

# drop duplicates from the new DataFrame
new_df.drop_duplicates(inplace=True)

# reset the index of the new DataFrame
new_df.reset_index(drop=True, inplace=True)

# print the new DataFrame
print(new_df)
```

```
           ID                                       Team1Players  \
0     1312200  (YBK Jaiswal, JC Buttler, SV Samson, D Padikka...
1     1312200  (YBK Jaiswal, JC Buttler, SV Samson, D Padikka...
2     1312199  (V Kohli, F du Plessis, RM Patidar, GJ Maxwell...
3     1312199  (V Kohli, F du Plessis, RM Patidar, GJ Maxwell...
4     1312198  (V Kohli, F du Plessis, RM Patidar, GJ Maxwell...
...       ...                                              ...
1923   335984  (G Gambhir, V Sehwag, S Dhawan, MK Tiwary, KD ...
1924   335983  (K Goel, JR Hopes, KC Sangakkara, Yuvraj Singh...
1925   335983  (K Goel, JR Hopes, KC Sangakkara, Yuvraj Singh...
1926   335982  (R Dravid, W Jaffer, V Kohli, JH Kallis, CL Wh...
1927   335982  (R Dravid, W Jaffer, V Kohli, JH Kallis, CL Wh...

                                        Team2Players  MatchNumber  innings  \
0     (WP Saha, Shubman Gill, MS Wade, HH Pandya, DA...        Final        1
1     (WP Saha, Shubman Gill, MS Wade, HH Pandya, DA...        Final        2
2     (YBK Jaiswal, JC Buttler, SV Samson, D Padikka...  Qualifier 2        1
3     (YBK Jaiswal, JC Buttler, SV Samson, D Padikka...  Qualifier 2        2
4     (Q de Kock, KL Rahul, M Vohra, DJ Hooda, MP St...   Eliminator        1
...                                                 ...          ...      ...
1923  (T Kohli, YK Pathan, SR Watson, M Kaif, DS Leh...            3        2
1924  (PA Patel, ML Hayden, MEK Hussey, MS Dhoni, SK...            2        1
1925  (PA Patel, ML Hayden, MEK Hussey, MS Dhoni, SK...            2        2
1926  (SC Ganguly, BB McCullum, RT Ponting, DJ Husse...            1        1
1927  (SC Ganguly, BB McCullum, RT Ponting, DJ Husse...            1        2

                                      Venue
0              Narendra Modi Stadium, Ahmedabad
1              Narendra Modi Stadium, Ahmedabad
2              Narendra Modi Stadium, Ahmedabad
3              Narendra Modi Stadium, Ahmedabad
4                         Eden Gardens, Kolkata
...                                         ...
1923                        Feroz Shah Kotla
1924  Punjab Cricket Association Stadium, Mohali
1925  Punjab Cricket Association Stadium, Mohali
1926                      M Chinnaswamy Stadium
1927                      M Chinnaswamy Stadium

[1928 rows x 6 columns]
```

In [13]:
```
df8 = pd.DataFrame(new_df)
```

In [14]:
```
df8
```

Out[14]:

	ID	Team1Players	Team2Players	MatchNumber	innings	Venue
0	1312200	(YBK Jaiswal, JC Buttler, SV Samson, D Padikka...	(WP Saha, Shubman Gill, MS Wade, HH Pandya, DA...	Final	1	Narendra Modi Stadium, Ahmedabad
1	1312200	(YBK Jaiswal, JC Buttler, SV Samson, D Padikka...	(WP Saha, Shubman Gill, MS Wade, HH Pandya, DA...	Final	2	Narendra Modi Stadium, Ahmedabad
2	1312199	(V Kohli, F du Plessis, RM Patidar, GJ Maxwell...	(YBK Jaiswal, JC Buttler, SV Samson, D Padikka...	Qualifier 2	1	Narendra Modi Stadium, Ahmedabad
3	1312199	(V Kohli, F du Plessis, RM Patidar, GJ Maxwell...	(YBK Jaiswal, JC Buttler, SV Samson, D Padikka...	Qualifier 2	2	Narendra Modi Stadium, Ahmedabad
4	1312198	(V Kohli, F du Plessis, RM Patidar, GJ Maxwell...	(Q de Kock, KL Rahul, M Vohra, DJ Hooda, MP St...	Eliminator	1	Eden Gardens, Kolkata
...	...	...	...	...	...	...
1923	335984	(G Gambhir, V Sehwag, S Dhawan, MK Tiwary, KD ...	(T Kohli, YK Pathan, SR Watson, M Kaif, DS Leh...	3	2	Feroz Shah Kotla
1924	335983	(K Goel, JR Hopes, KC Sangakkara, Yuvraj Singh...	(PA Patel, ML Hayden, MEK Hussey, MS Dhoni, SK...	2	1	Punjab Cricket Association Stadium, Mohali
1925	335983	(K Goel, JR Hopes, KC Sangakkara, Yuvraj Singh...	(PA Patel, ML Hayden, MEK Hussey, MS Dhoni, SK...	2	2	Punjab Cricket Association Stadium, Mohali
1926	335982	(R Dravid, W Jaffer, V Kohli, JH Kallis, CL Wh...	(SC Ganguly, BB McCullum, RT Ponting, DJ Husse...	1	1	M Chinnaswamy Stadium
1927	335982	(R Dravid, W Jaffer, V Kohli, JH Kallis, CL Wh...	(SC Ganguly, BB McCullum, RT Ponting, DJ Husse...	1	2	M Chinnaswamy Stadium

1928 rows × 6 columns

In [15]:
```
new_ipl1 = pd.merge(df8, df4,  how='outer', left_on=['ID','MatchNumber','innings','Ver
```

In [16]:
```
df5 = new_ipl1.copy()
```

In [17]:
```
df5.columns
```

Out[17]:
```
Index(['ID', 'Team1Players', 'Team2Players', 'MatchNumber', 'innings', 'Venue',
       'BattingTeam', 'bowling_team', 'total_runs'],
      dtype='object')
```

In [18]:
```
team_name_mapping = {
    'Deccan Chargers': 'Sunrisers Hyderabad',
    'Delhi Daredevils': 'Delhi Capitals',
```

```
    'Kings XI Punjab': 'Punjab Kings',
    'Rising Pune Supergiants' : 'Chennai Super Kings',
    'Rising Pune Supergiant' : 'Chennai Super Kings' ,
    'Gujarat Lions' : 'Rajasthan Royals'
}
```

In [19]:
```
df5['BattingTeam'].replace(team_name_mapping, inplace=True)
df5['bowling_team'].replace(team_name_mapping, inplace=True)

# Drop Pune Warrior Team details as no franchize brought that team
df5 = df5[(df5['BattingTeam'] != 'Pune Warriors') & (df5['bowling_team'] != 'Pune Warr
df5 = df5[(df5['BattingTeam'] != 'Kochi Tuskers Kerala') & (df5['bowling_team'] != 'Kc
```

In [20]:
```
data = {'Venue': ['Narendra Modi Stadium', 'MA Chidambaram Stadium', 'Arun Jaitley Sta
df_Venues = pd.DataFrame(data)

# Making the Venue list to match with given 2023 Venue list
df5['Venue'] = df5["Venue"].str.split(', ', expand=True)[0]

# Select data of those match with 2023 Venue details
df5 = df5[df5['Venue'].isin(df_Venues['Venue'])]
```

In [21]:
```
venue_dict = {}
index = 0

for team in df5['Venue']:
    if team not in venue_dict:
        venue_dict[team] = index
        index += 1

print(venue_dict)
```

```
{'Narendra Modi Stadium': 0, 'Eden Gardens': 1, 'Wankhede Stadium': 2, 'Brabourne Sta
dium': 3, 'Dr DY Patil Sports Academy': 4, 'Maharashtra Cricket Association Stadium':
5, 'Arun Jaitley Stadium': 6, 'MA Chidambaram Stadium': 7, 'Rajiv Gandhi Internationa
l Stadium': 8, 'Punjab Cricket Association IS Bindra Stadium': 9, 'M.Chinnaswamy Stad
ium': 10, 'Sawai Mansingh Stadium': 11}
```

In [22]:
```
team1_dict = {}
index = 0

for team in df5['BattingTeam']:
    if team not in team1_dict:
        team1_dict[team] = index
        index += 1

print(team1_dict)
```

```
{'Rajasthan Royals': 0, 'Gujarat Titans': 1, 'Royal Challengers Bangalore': 2, 'Luckn
ow Super Giants': 3, 'Sunrisers Hyderabad': 4, 'Punjab Kings': 5, 'Delhi Capitals':
6, 'Mumbai Indians': 7, 'Chennai Super Kings': 8, 'Kolkata Knight Riders': 9}
```

In [23]:
```
team2_dict = {}
index = 0

for team in df5['bowling_team']:
    if team not in team2_dict:
        team2_dict[team] = index
        index += 1
```

```
print(team2_dict)
```

```
{'Gujarat Titans': 0, 'Rajasthan Royals': 1, 'Royal Challengers Bangalore': 2, 'Luckn
ow Super Giants': 3, 'Punjab Kings': 4, 'Sunrisers Hyderabad': 5, 'Delhi Capitals':
6, 'Mumbai Indians': 7, 'Chennai Super Kings': 8, 'Kolkata Knight Riders': 9}
```

In [24]:
```
df5['venue_ids'] = df5['Venue'].apply(lambda x: venue_dict[x])
```

In [25]:
```
df5['team1_ids'] = df5['BattingTeam'].apply(lambda x: team1_dict[x])
```

In [26]:
```
df5['team2_ids'] = df5['bowling_team'].apply(lambda x: team2_dict[x])
```

In [27]:
```
df5.drop(['MatchNumber','Venue','BattingTeam', 'bowling_team'], axis=1 , inplace = Tru
```

In [28]:
```
df5.rename(columns={'venue_ids': 'Venue','team1_ids': 'batting_team', 'team2_ids' : 'b
```

In [29]:
```
df5 = df5[['Venue','innings','batting_team','bowling_team','Team1Players', 'Team2Playe
```

In [30]:
```
df5.head()
```

Out[30]:

	Venue	innings	batting_team	bowling_team	Team1Players	Team2Players	total_runs
0	0	1	0	0	(YBK Jaiswal, JC Buttler, SV Samson, D Padikka...	(WP Saha, Shubman Gill, MS Wade, HH Pandya, DA...	54
1	0	2	1	1	(YBK Jaiswal, JC Buttler, SV Samson, D Padikka...	(WP Saha, Shubman Gill, MS Wade, HH Pandya, DA...	35
2	0	1	2	2	(V Kohli, F du Plessis, RM Patidar, GJ Maxwell...	(YBK Jaiswal, JC Buttler, SV Samson, D Padikka...	52
3	0	2	0	1	(V Kohli, F du Plessis, RM Patidar, GJ Maxwell...	(YBK Jaiswal, JC Buttler, SV Samson, D Padikka...	77
4	1	1	2	2	(V Kohli, F du Plessis, RM Patidar, GJ Maxwell...	(Q de Kock, KL Rahul, M Vohra, DJ Hooda, MP St...	60

In [31]:
```
unique_players1 = df5['Team1Players'].explode().unique().tolist()
unique_players2 = df5['Team2Players'].explode().unique().tolist()

unique_players = list(set(unique_players1 + unique_players2))
```

In [32]:
```
unique_players
```

Out[32]:
```
['DNT Zoysa',
 'Sachin Baby',
 'MP Stoinis',
 'AP Dole',
 'N Pooran',
 'OF Smith',
 'Ankit Soni',
 'P Dogra',
 'MA Agarwal',
 'JJ Bumrah',
 'VVS Laxman',
 'B Sai Sudharsan',
 'MK Pandey',
 'S Chanderpaul',
 'SC Kuggeleijn',
 'SM Curran',
 'Rasikh Salam',
 'S Sreesanth',
 'A Mishra',
 'Y Nagar',
 'Umar Gul',
 'DA Miller',
 'DJ Bravo',
 'Mandeep Singh',
 'M Shahrukh Khan',
 'AJ Turner',
 'MC Juneja',
 'JPR Scantlebury-Searles',
 'AG Paunikar',
 'AR Bawne',
 'S Nadeem',
 'A Nortje',
 'B Akhil',
 'A Kumble',
 'Anureet Singh',
 'RV Pawar',
 'CJ Dala',
 'SO Hetmyer',
 'A Ashish Reddy',
 'J Theron',
 'PSP Handscomb',
 'SJ Srivastava',
 'BB Samantray',
 'M Morkel',
 'NK Patel',
 'D Salunkhe',
 'AA Chavan',
 'SL Malinga',
 'BB Sran',
 'Anmolpreet Singh',
 'C de Grandhomme',
 'B Indrajith',
 'TS Mills',
 'BCJ Cutting',
 'KD Karthik',
 'SW Billings',
 'K Gowtham',
 'PA Reddy',
 'D Padikkal',
 'MF Maharoof',
```

```
'D Wiese',
'AN Ahmed',
'D Brevis',
'DE Bollinger',
'RT Ponting',
'AS Roy',
'SB Styris',
'AS Yadav',
'JR Hopes',
'Fazalhaq Farooqi',
'A Zampa',
'J Arunkumar',
'SK Warne',
'RV Patel',
'Avesh Khan',
'Anand Rajan',
'MS Dhoni',
'JD Unadkat',
'Shakib Al Hasan',
'NLTC Perera',
'AA Jhunjhunwala',
'MR Marsh',
'Mohsin Khan',
'P Parameswaran',
'CL White',
'Iqbal Abdulla',
'SR Watson',
'Mohammed Shami',
'BB McCullum',
'MJ Lumb',
'AC Gilchrist',
'J Suchith',
'MS Wade',
'LPC Silva',
'HH Gibbs',
'MM Ali',
'RP Meredith',
'DA Warner',
'M Jansen',
'C Madan',
'F du Plessis',
'R Parag',
'PP Ojha',
'M Pathirana',
'LI Meriwala',
'TU Deshpande',
'NJ Maddinson',
'Vishnu Vinod',
'YS Chahal',
'C Ganapathy',
'GC Viljoen',
'AD Mathews',
'PA Patel',
'AT Rayudu',
'P Kumar',
'B Stanlake',
'TH David',
'KA Jamieson',
'K Khejroliya',
'YV Takawale',
```

```
'RG More',
'A Nehra',
'Sunny Singh',
'C Munro',
'DJ Malan',
'Mohammad Hafeez',
'AR Patel',
'RA Bawa',
'Harbhajan Singh',
'RR Powar',
'NJ Rimmington',
'DG Nalkande',
'Karanveer Singh',
'KS Williamson',
'A Chopra',
'S Kaushik',
'BA Stokes',
'MK Lomror',
'KK Nair',
'DJ Mitchell',
'MV Boucher',
'CRD Fernando',
'Misbah-ul-Haq',
'AC Voges',
'R Dravid',
'CA Pujara',
'Anuj Rawat',
'Ankit Sharma',
'BMAJ Mendis',
'RE Levi',
'KP Appanna',
'SM Boland',
'SB Wagh',
'BW Hilfenhaus',
'AM Rahane',
'Basil Thampi',
'M Rawat',
'KL Rahul',
'S Aravind',
'HM Amla',
'W Jaffer',
'Ishan Kishan',
'BJ Haddin',
'AC Thomas',
'WD Parnell',
'Swapnil Singh',
'TM Head',
'Yash Dayal',
'LS Livingstone',
'S Kaul',
'PP Chawla',
'VS Malik',
'D Kalyankrishna',
'CA Lynn',
'CJ Jordan',
'JL Denly',
'SK Raina',
'YK Pathan',
'RR Pant',
'RJ Harris',
```

```
'AC Blizzard',
'Shahid Afridi',
'A Chandila',
'DJ Willey',
'P Awana',
'WPUJC Vaas',
'RV Uthappa',
'Harmeet Singh',
'M de Lange',
'RJ Peterson',
'Gurkeerat Singh',
'JJ van der Wath',
'Azhar Mahmood',
'V Shankar',
'DJ Hussey',
'RK Bhui',
'DR Shorey',
'AS Joseph',
'D Pretorius',
'KH Pandya',
'R McLaren',
'KR Sen',
'MN Samuels',
'SK Trivedi',
'Abdul Samad',
'Ramandeep Singh',
'HH Pandya',
'KV Sharma',
'Parvez Rasool',
'Shashank Singh',
'KC Sangakkara',
'TD Paine',
'P Chopra',
'NS Naik',
'Simarjeet Singh',
'JP Duminy',
'MJ Henry',
'Y Venugopal Rao',
'M Manhas',
'JM Kemp',
'BE Hendricks',
'RV Gomez',
'VS Yeligati',
'S Sandeep Warrier',
'A Choudhary',
'CJ Anderson',
'P Ray Barman',
'JC Buttler',
'RG Sharma',
'RM Patidar',
'Z Khan',
'Kamran Khan',
'Jaskaran Singh',
'MM Sharma',
'SA Asnodkar',
'FY Fazal',
'KA Pollard',
'KW Richardson',
'TA Boult',
'HE van der Dussen',
```

```
'Kuldeep Yadav',
'T Stubbs',
'LH Ferguson',
'BR Dunk',
'H Das',
'OC McCoy',
'DJ Harris',
'PK Garg',
'CM Gautam',
'Tilak Varma',
'LA Pomersbach',
'KM Jadhav',
'H Klaasen',
'SP Jackson',
'R Powell',
'DH Yagnik',
'X Thalaivan Sargunam',
'SM Pollock',
'AD Nath',
'CK Kapugedera',
'VY Mahesh',
'TK Curran',
'PV Tambe',
'S Badrinath',
'TL Suman',
'MS Bisla',
'SP Goswami',
'CR Woakes',
'TM Srivastava',
'G Gambhir',
'I Malhotra',
'J Yadav',
'LE Plunkett',
'DB Ravi Teja',
'R Rampaul',
'MS Gony',
'UT Khawaja',
'Jalaj S Saxena',
'YBK Jaiswal',
'Q de Kock',
'WA Mota',
'SP Fleming',
'CH Morris',
'M Markande',
'F Behardien',
'SV Samson',
'N Saini',
'A Manohar',
'Shivam Sharma',
'KMA Paul',
'R Sai Kishore',
'JD Ryder',
'KL Nagarkoti',
'AD Mascarenhas',
'PM Sarvesh Kumar',
'Navdeep Saini',
'U Kaul',
'SP Narine',
'IK Pathan',
'DS Kulkarni',
```

```
'BJ Rohrer',
'P Simran Singh',
'R Vinay Kumar',
'V Pratap Singh',
'R Shukla',
'A Tomar',
'DJ Thornely',
'Lalit Yadav',
'SR Tendulkar',
'Shahbaz Ahmed',
'MC Henriques',
'Abhishek Sharma',
'PC Valthaty',
'KB Arun Karthik',
'CA Ingram',
'S Rana',
'NL McCullum',
'Mohammad Nabi',
'KK Ahmed',
'VR Aaron',
'PP Shaw',
'GD McGrath',
'N Rana',
'SN Khan',
'K Kartikeya',
'MJ McClenaghan',
'PJ Sangwan',
'RS Bopara',
'JA Richardson',
'J Botha',
'CK Langeveldt',
'DP Nannes',
'KS Bharat',
'S Anirudha',
'Harshit Rana',
'DT Patil',
'DB Das',
'SS Prabhudessai',
'VG Arora',
'A Mithun',
'CJ McKay',
'R Dhawan',
'V Kohli',
'HF Gurney',
'AK Markram',
'GC Smith',
'RD Gaikwad',
'T Natarajan',
'SW Tait',
'PJ Cummins',
'JEC Franklin',
'JA Morkel',
'S Badree',
'AD Russell',
'MJ Santner',
'JR Hazlewood',
'C Sakariya',
'AM Salvi',
'L Ngidi',
'TG Southee',
```

```
'GS Sandhu',
'SD Lad',
'Mohammed Siraj',
'HR Shokeen',
'Y Prithvi Raj',
'DPMD Jayawardene',
'DJ Jacobs',
'Kamran Akmal',
'JO Holder',
'NT Ellis',
'R Shepherd',
'DJM Short',
'UA Birla',
'T Mishra',
'MB Parmar',
'J Syed Mohammad',
'R Ninan',
'Gagandeep Singh',
'Washington Sundar',
'N Jagadeesan',
'CV Varun',
'Abdur Razzak',
'S Vidyut',
'PN Mankad',
'OA Shah',
'Harpreet Singh',
'M Prasidh Krishna',
'A Mukund',
'A Uniyal',
'DS Lehmann',
'Shoaib Malik',
'DT Christian',
'SS Iyer',
'RN ten Doeschate',
'Umran Malik',
'L Ronchi',
'AG Murtaza',
'VR Iyer',
'JC Archer',
'KMDN Kulasekara',
'TR Birt',
'EJG Morgan',
'A Badoni',
'M Theekshana',
'AJ Tye',
'Mohammad Asif',
'DL Chahar',
'R Tewatia',
'P Negi',
'MEK Hussey',
'DJ Muthuswami',
'A Symonds',
'S Randiv',
'DJ Hooda',
'PD Collingwood',
'S Dhawan',
'MG Johnson',
'M Vohra',
'TP Sudhindra',
'S Sohal',
```

```
'T Thushara',
'PR Shah',
'KP Pietersen',
'AF Milne',
'E Lewis',
'DP Vijaykumar',
'PVD Chameera',
'Mujeeb Ur Rahman',
'MK Tiwary',
'R Sharma',
'S Dube',
'DW Steyn',
'SE Marsh',
'KK Cooper',
'SE Bond',
'AL Menaria',
'ML Hayden',
'B Laughlin',
'PWH de Silva',
'SB Bangar',
'GB Hogg',
'MA Wood',
'HV Patel',
'JDS Neesham',
'O Thomas',
'KM Asif',
'JJ Roy',
'NV Ojha',
'SC Ganguly',
'B Chipli',
'KJ Abbott',
'SA Abbott',
'IC Pandey',
'RE van der Merwe',
'S Gopal',
'Sohail Tanvir',
'RA Jadeja',
'P Sahu',
'JE Taylor',
'RP Singh',
'Shivam Mavi',
'V Sehwag',
'RS Gavaskar',
'TL Seifert',
'SN Thakur',
'R Sanjay Yadav',
'Ravi Bishnoi',
'Shoaib Akhtar',
'JP Behrendorff',
'M Ashwin',
'AA Kazi',
'S Ladda',
'RW Price',
'CJ Ferguson',
'RR Bhatkal',
'P Suyal',
'Yuvraj Singh',
'AP Tare',
'KS Sharma',
'SB Jakati',
```

```
                'MD Mishra',
                'NM Coulter-Nile',
                'IR Jaggi',
                'P Amarnath',
                'TM Dilshan',
                'SB Joshi',
                'M Ntini',
                'LRPL Taylor',
                'Rashid Khan',
                'SMSM Senanayake',
                'JM Bairstow',
                'AB McDonald',
                'T Taibu',
                'DL Vettori',
                'BJ Hodge',
                'Mustafizur Rahman',
                'Mukesh Choudhary',
                'Virat Singh',
                'BA Bhatt',
                'M Kartik',
                'B Kumar',
                'VRV Singh',
                'PH Solanki',
                'B Lee',
                'DR Sams',
                'IS Sodhi',
                'Shubman Gill',
                'AUK Pathan',
                'AD Hales',
                'Imran Tahir',
                'A Dananjaya',
                'JM Sharma',
                'WP Saha',
                'JDP Oram',
                'S Midhun',
                'S Lamichhane',
                'M Kaif',
                'I Sharma',
                'L Balaji',
                'K Goel',
                'GJ Maxwell',
                'AJ Finch',
                'MA Starc',
                'Joginder Sharma',
                'MJ Guptill',
                'M Muralitharan',
                'Akash Deep',
                'BAW Mendis',
                'RR Sarwan',
                'Sunny Gupta',
                'AM Nayar',
                'B Sumanth',
                'CH Gayle',
                'ND Doshi',
                'R Bhatia',
                'PBB Rajapaksa',
                'UBT Chand',
                'RK Singh',
                'LMP Simmons',
                'MA Khote',
```

```
'DJG Sammy',
'R Sathish',
'AB de Villiers',
'Aman Hakim Khan',
'DR Smith',
'SA Yadav',
'Kartik Tyagi',
'DP Conway',
'Pankaj Singh',
'S Tyagi',
'Tejas Baroka',
'RD Chahar',
'SE Rutherford',
'AB Dinda',
'JP Faulkner',
'AS Raut',
'Bipul Sharma',
'T Shamsi',
'M Vijay',
'A Singh',
'UT Yadav',
'SPD Smith',
'Arshdeep Singh',
'RA Tripathi',
'GJ Bailey',
'Harpreet Brar',
'RR Rossouw',
'MM Patel',
'ST Jayasuriya',
'SS Tiwary',
'GH Vihari',
'JH Kallis',
'AB Agarkar',
'Anirudh Singh',
'JW Hastings',
'K Upadhyay',
'LR Shukla',
'AS Rajpoot',
'K Rabada',
'RR Raje',
'STR Binny',
'T Kohli',
'KC Cariappa',
'SD Chitnis',
'DM Bravo',
'Sandeep Sharma',
'KAJ Roach',
'FA Allen',
'CR Brathwaite',
'S Narwal',
'Salman Butt',
'R Ashwin']
```

In [33]:
```
for name in unique_players:
    df5[name] = df5.apply(lambda row: 1 if name in row['Team1Players'] or name in row[
```

In [34]:
```
df5.drop(['Team1Players','Team2Players'], axis=1 , inplace = True)
```

In [35]:
```
df5.head()
```

Out[35]:

	Venue	innings	batting_team	bowling_team	total_runs	DNT Zoysa	Sachin Baby	MP Stoinis	AP Dole	N Pooran	...
0	0	1	0	0	54	0	0	0	0	0	...
1	0	2	1	1	35	0	0	0	0	0	...
2	0	1	2	2	52	0	0	0	0	0	...
3	0	2	0	1	77	0	0	0	0	0	...
4	1	1	2	2	60	0	0	1	0	0	...

5 rows × 597 columns

In [36]:
```
features = df5.drop(['total_runs'], axis=1)
target = df5['total_runs']
```

In [37]:
```
acc = []
model = []
```

In [38]:
```
from sklearn.model_selection import train_test_split
Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size = 0.2,random
```

In [39]:
```
from sklearn.tree import DecisionTreeClassifier
classifier= DecisionTreeClassifier(criterion='entropy', random_state=0)
classifier.fit(Xtrain, Ytrain)
y_pred=classifier.predict(Xtest)
from sklearn.metrics import accuracy_score
accuracy=accuracy_score(y_pred, Ytest)
print('Decision Tree Model accuracy score: {0:0.4f}'.format(accuracy_score(Ytest, y_pr
```

```
Decision Tree Model accuracy score: 0.0099
```

In [40]:
```
from sklearn.ensemble import RandomForestClassifier
RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)
predicted_values = RF.predict(Xtest)
from sklearn.metrics import accuracy_score
x = accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)
from sklearn.metrics import classification_report
print(classification_report(Ytest,predicted_values))
```

```
RF's Accuracy is:  0.024630541871921183
              precision    recall  f1-score   support

           7       0.00      0.00      0.00         1
          10       0.00      0.00      0.00         0
          15       0.00      0.00      0.00         0
          20       0.00      0.00      0.00         1
          26       0.00      0.00      0.00         1
          27       0.00      0.00      0.00         1
          29       0.00      0.00      0.00         0
          30       0.00      0.00      0.00         1
          31       0.00      0.00      0.00         0
          32       0.00      0.00      0.00         1
          33       0.00      0.00      0.00         1
          34       0.00      0.00      0.00         3
          35       0.00      0.00      0.00         4
          36       0.00      0.00      0.00         1
          37       0.00      0.00      0.00         5
          38       0.00      0.00      0.00         8
          39       0.00      0.00      0.00         3
          40       0.00      0.00      0.00         6
          41       0.00      0.00      0.00         4
          42       0.00      0.00      0.00         4
          43       0.12      0.09      0.11        11
          44       0.00      0.00      0.00         2
          45       0.00      0.00      0.00         2
          46       0.00      0.00      0.00         6
          47       0.00      0.00      0.00         5
          48       0.00      0.00      0.00         5
          49       0.11      0.10      0.11        10
          50       0.00      0.00      0.00         5
          51       0.00      0.00      0.00         6
          52       0.00      0.00      0.00         4
          53       0.00      0.00      0.00         4
          54       0.00      0.00      0.00        10
          55       0.00      0.00      0.00         8
          56       0.33      0.25      0.29         4
          57       0.00      0.00      0.00         6
          58       0.00      0.00      0.00         7
          59       0.00      0.00      0.00         2
          60       0.20      0.33      0.25         3
          61       0.00      0.00      0.00         4
          62       0.00      0.00      0.00         1
          63       0.00      0.00      0.00         7
          64       0.00      0.00      0.00         6
          65       0.00      0.00      0.00         3
          66       0.00      0.00      0.00         1
          67       0.00      0.00      0.00         6
          68       0.00      0.00      0.00         3
          69       0.00      0.00      0.00         3
          70       0.00      0.00      0.00         1
          71       0.00      0.00      0.00         4
          72       0.00      0.00      0.00         1
          73       0.00      0.00      0.00         2
          74       1.00      0.50      0.67         2
          75       0.00      0.00      0.00         3
          76       0.00      0.00      0.00         1
          77       0.00      0.00      0.00         3
          78       0.00      0.00      0.00         0
          79       0.00      0.00      0.00         1
```

```
          80       0.00      0.00      0.00         2
          82       0.00      0.00      0.00         2
          83       0.00      0.00      0.00         1
          84       0.00      0.00      0.00         0
          87       0.00      0.00      0.00         0
          89       0.00      0.00      0.00         0
         107       0.00      0.00      0.00         1

    accuracy                           0.02       203
   macro avg       0.03      0.02      0.02       203
weighted avg       0.03      0.02      0.03       203
```

```
C:\Users\Rog Strix\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:131
8: 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 be
havior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Rog Strix\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:131
8: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in
labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Rog Strix\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:131
8: 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 be
havior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Rog Strix\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:131
8: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in
labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Rog Strix\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:131
8: 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 be
havior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Rog Strix\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:131
8: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in
labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [41]:

```python
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
lr = LinearRegression()
lr.fit(Xtrain, Ytrain)

# make predictions on the testing set
y_pred = lr.predict(Xtest)

# evaluate the model by calculating the root mean squared error (RMSE)
rmse = mean_squared_error(Ytest, y_pred, squared=False)

print(f"RMSE: {rmse}")
```

```
RMSE: 31.897952362185627
```

In [42]:

```python
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

svm = SVC()
svm.fit(Xtrain, Ytrain)
```

```
# make predictions on the testing set
y_pred = svm.predict(Xtest)

# evaluate the model by calculating the accuracy
accuracy = accuracy_score(Ytest, y_pred)

print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.014778325123152709
```

In [43]:
```
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error

gb = GradientBoostingRegressor()
gb.fit(Xtrain, Ytrain)

# make predictions on the testing set
y_pred = gb.predict(Xtest)

# evaluate the model by calculating the mean squared error
mse = mean_squared_error(Ytest, y_pred)

print(f"Mean Squared Error: {mse}")
```

```
Mean Squared Error: 163.3462499088171
```

In [44]:
```
from sklearn.metrics import mean_squared_error
from sklearn.svm import SVR
from sklearn.linear_model import BayesianRidge
import xgboost as xgb
```

In [45]:
```
svr = SVR()
svr.fit(Xtrain, Ytrain)
y_pred = svr.predict(Xtest)
mse = mean_squared_error(Ytest, y_pred)
print(f"SVR Mean Squared Error: {mse}")
```

```
SVR Mean Squared Error: 178.96308070437175
```

In [46]:
```
bayesian = BayesianRidge()
bayesian.fit(Xtrain, Ytrain)
y_pred = bayesian.predict(Xtest)
mse = mean_squared_error(Ytest, y_pred)
print(f"Bayesian Regression Mean Squared Error: {mse}")
```

```
Bayesian Regression Mean Squared Error: 179.07394291552848
```

In [47]:
```
dtrain = xgb.DMatrix(Xtrain, label=Ytrain)
dtest = xgb.DMatrix(Xtest, label=Ytest)
params = {
    "objective": "reg:squarederror",
    "learning_rate": 0.5,
    "max_depth": 2,
    "min_child_weight": 3,
    "gamma": 100.0
}
xgb_model = xgb.train(params, dtrain)
y_pred = xgb_model.predict(dtest)
mse = mean_squared_error(Ytest, y_pred)
print(f"XGBoost Mean Squared Error: {mse}")
```

```
XGBoost Mean Squared Error: 162.98827268228766
```

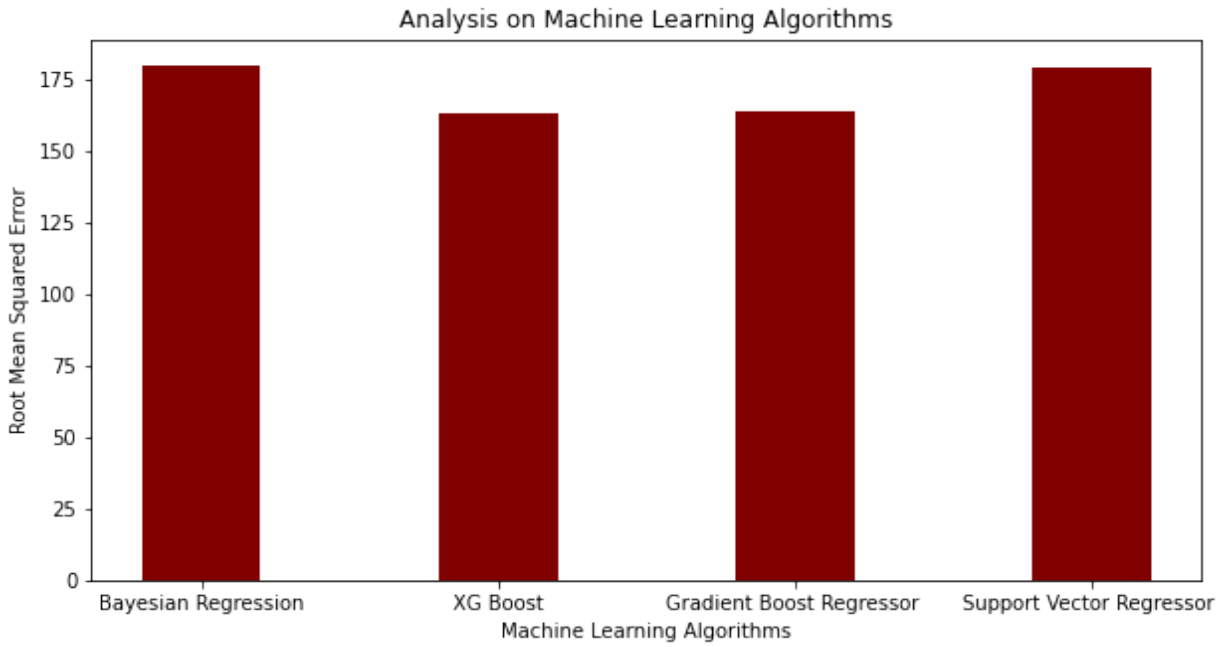
In [48]:
```
import numpy as np
import matplotlib.pyplot as plt


# creating the dataset
data = {'Bayesian Regression': 179.74, 'XG Boost': 162.99, 'Gradient Boost Regressor '
        'Support Vector Regressor ':178.96}
Machine_Learning_Algorithms = list(data.keys())
Root_Mean_Squared_Error = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(Machine_Learning_Algorithms, Root_Mean_Squared_Error, color ='maroon',
        width = 0.4)

plt.xlabel("Machine Learning Algorithms")
plt.ylabel("Root Mean Squared Error")
plt.title("Analysis on Machine Learning Algorithms")
plt.show()
```



In [49]:
```
df11 = pd.DataFrame(columns=df5.columns)
```

In [50]:
```
df11
```

Out[50]:

Venue	innings	batting_team	bowling_team	total_runs	DNT Zoysa	Sachin Baby	MP Stoinis	AP Dole	N Pooran	...

0 rows × 597 columns

In [51]:
```
df10 = pd.read_csv('C:/Users/Rog Strix/Downloads/2022DA04050/training_dataset/test_fil
```

In [52]:
```
df10
```

Out[52]:

	venue	innings	batting_team	bowling_team	batsmen	bowlers
0	Narendra Modi Stadium	1	Gujarat Titans	Rajasthan Royals	WP Saha, Shubman Gill, MS Wade	TA Boult, Sandeep Sharma, A Zampa
1	Narendra Modi Stadium	2	Rajasthan Royals	Gujarat Titans	YBK Jaiswal, SV Samson	HH Pandya, Mohammed Shami

In [53]:
```
df10['venue_ids'] = df10['venue'].apply(lambda x: venue_dict[x])
```

In [54]:
```
df10['batting_team_ids'] = df10['batting_team'].apply(lambda x: team1_dict[x])
```

In [55]:
```
df10['bowling_team_ids'] = df10['bowling_team'].apply(lambda x: team2_dict[x])
```

In [56]:
```
unique_players1 = df10['batsmen'].explode().unique().tolist()
unique_players2 = df10['bowlers'].explode().unique().tolist()

unique_players_test = list(set(unique_players1 + unique_players2))
```

In [57]:
```
unique_players_test
```

Out[57]:
```
['WP Saha, Shubman Gill, MS Wade',
 'HH Pandya, Mohammed Shami',
 'TA Boult, Sandeep Sharma, A Zampa',
 'YBK Jaiswal, SV Samson']
```

In [58]:
```
unique_players = []
for team_players in df10[['batsmen', 'bowlers']].values:
    for player in team_players:
        unique_players.extend([p.strip() for p in player.split(',')])
unique_players = list(set(unique_players))
```

In [59]:
```
unique_players
```

Out[59]:
```
['YBK Jaiswal',
 'Mohammed Shami',
 'Sandeep Sharma',
 'SV Samson',
 'TA Boult',
 'A Zampa',
 'WP Saha',
 'Shubman Gill',
 'MS Wade',
 'HH Pandya']
```

In [60]:
```
for name in unique_players:
    df10[name] = df10.apply(lambda row: 1 if name in row['batsmen'] or name in row['bo
```

In [61]:
```
df10.drop(['batsmen','bowlers','venue','batting_team','bowling_team'], axis=1 , inplac
```

In [62]:
```
df10.head()
```

Out[62]:

	innings	venue_ids	batting_team_ids	bowling_team_ids	YBK Jaiswal	Mohammed Shami	Sandeep Sharma	SV Samson	E
0	1	0	1	1	0	0	1	0	
1	2	0	0	0	1	1	0	1	

In [63]:
```
df10.rename(columns={'venue_ids': 'Venue','batting_team_ids': 'batting_team', 'bowling
```

In [64]:
```
for col in df10.columns:
    if col not in df11.columns:
        df11[col] = 0
```

In [65]:
```
df_combined = pd.concat([df11, df10], axis=0)
```

In [66]:
```
df_combined.fillna(0, inplace=True)
```

In [67]:
```
df_combined.drop('total_runs', axis=1, inplace=True)
```

In [68]:
```
y_pred=RF.predict(df_combined)
```

In [69]:
```
y_pred
```

Out[69]:
```
array([54, 54], dtype=int64)
```

In [ ]: