CRICKET PLAYER PERFORMANCE PREDICTION USING MACHINE LEARNING

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

In [2]:
```
df = pd.read_csv('cricket_player.csv')
df.head()
```

Out[2]:

	ID	Player_ID	Player	Country	Opposition	Bat1	Runs	BF	SR	4s	6s	Ground	Start Date	Match_ID	Team Runs	Target	Overs	RPO	I
0	6	49619	Oshane Thomas	WestIndies	v India	0*	0	8	0.00	0	0	Dhaka	10 Jan 2010	ODI # 2941	NaN	NaN	NaN	NaN	I
1	10	49619	Oshane Thomas	WestIndies	v England	1*	1	3	33.33	0	0	Nottingham	6 Jul 2011	ODI # 3169	NaN	NaN	NaN	NaN	I
2	11	49619	Oshane Thomas	WestIndies	v Australia	0*	0	2	0.00	0	0	Pallekele	10 Aug 2011	ODI # 3175	NaN	NaN	NaN	NaN	I
3	12	49619	Oshane Thomas	WestIndies	v Pakistan	0	0	2	0.00	0	0	Dubai (DSC)	11 Nov 2011	ODI # 3212	NaN	NaN	NaN	NaN	I
4	14	49619	Oshane Thomas	WestIndies	v Pakistan	0	0	4	0.00	0	0	Dhaka	15 Mar 2012	ODI # 3260	NaN	NaN	NaN	NaN	I

In [3]:
```
df.shape
```

Out[3]:
```
(8818, 20)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8818 entries, 0 to 8817
Data columns (total 20 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   ID          8818 non-null   int64  
 1   Player_ID   8818 non-null   int64  
 2   Player      8818 non-null   object 
 3   Country     8818 non-null   object 
 4   Opposition  8818 non-null   object 
 5   Bat1        8818 non-null   object 
 6   Runs        8818 non-null   int64  
 7   BF          8818 non-null   int64  
 8   SR          8818 non-null   float64
 9   4s          8818 non-null   int64  
 10  6s          8818 non-null   int64  
 11  Ground      8818 non-null   object 
 12  Start Date  8818 non-null   object 
 13  Match_ID    8818 non-null   object 
 14  Team Runs   6192 non-null   object 
 15  Target      3086 non-null   float64
 16  Overs       6192 non-null   float64
 17  RPO         6192 non-null   object 
 18  Inns        6192 non-null   float64
 19  Result      6192 non-null   object 
dtypes: float64(4), int64(6), object(10)
memory usage: 1.3+ MB
```

In [5]:
```
null = df.isnull().sum()
null_perect = (null/len(df))*100
null_perect
```

Out[5]:
```
ID             0.000000
Player_ID      0.000000
Player         0.000000
Country        0.000000
Opposition     0.000000
Bat1           0.000000
Runs           0.000000
BF             0.000000
SR             0.000000
4s             0.000000
6s             0.000000
Ground         0.000000
Start Date     0.000000
Match_ID       0.000000
Team Runs     29.779995
Target        65.003402
Overs         29.779995
RPO           29.779995
Inns          29.779995
Result        29.779995
dtype: float64
```

In [6]:
```
df.dropna(inplace=True)
```

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

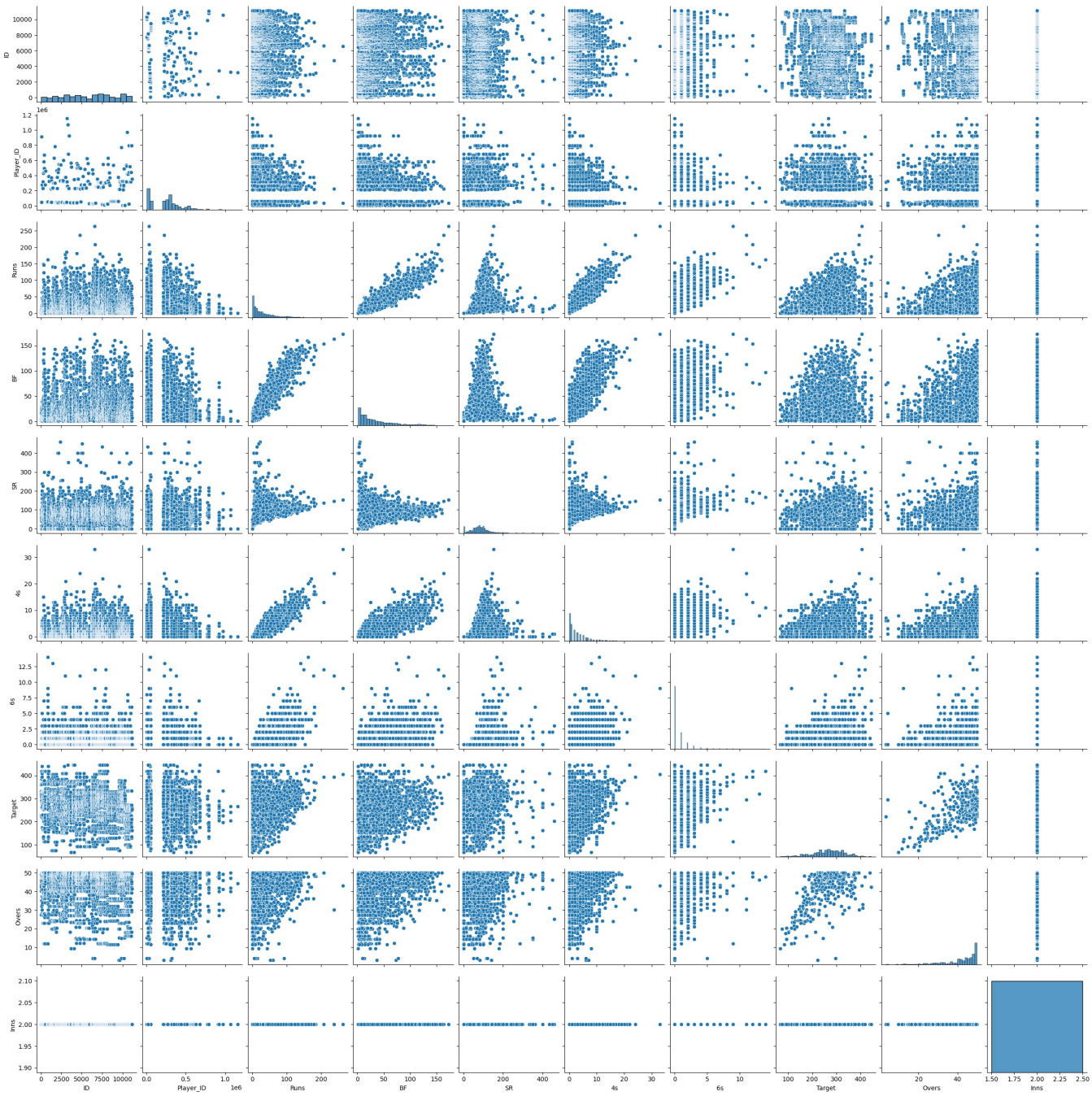
Out[7]:
```
ID            0
Player_ID     0
Player        0
Country       0
Opposition    0
Bat1          0
Runs          0
BF            0
SR            0
4s            0
6s            0
Ground        0
Start Date    0
Match_ID      0
Team Runs     0
Target        0
Overs         0
RPO           0
Inns          0
Result        0
dtype: int64
```

In [8]:
```
df.describe()
```

Out[8]:

	ID	Player_ID	Runs	BF	SR	4s	6s	Target	Overs	Inns
count	3086.000000	3.086000e+03	3086.000000	3086.000000	3086.000000	3086.000000	3086.000000	3086.000000	3086.000000	3086.0
mean	5813.602074	2.539303e+05	30.366818	33.863253	82.964332	2.786455	0.604990	266.442320	42.007323	2.0
std	3153.114936	1.889530e+05	33.430199	32.897672	51.638874	3.384409	1.261252	66.641748	8.760664	0.0
min	18.000000	5.334000e+03	0.000000	1.000000	0.000000	0.000000	0.000000	68.000000	3.200000	2.0
25%	3113.250000	4.975800e+04	6.000000	9.000000	53.330000	0.000000	0.000000	227.000000	39.100000	2.0
50%	5961.500000	2.707495e+05	19.000000	23.000000	80.000000	2.000000	0.000000	270.000000	44.500000	2.0
75%	8329.750000	3.481440e+05	44.000000	49.000000	104.812500	4.000000	1.000000	311.000000	48.500000	2.0
max	11145.000000	1.158100e+06	264.000000	173.000000	460.000000	33.000000	14.000000	445.000000	50.000000	2.0

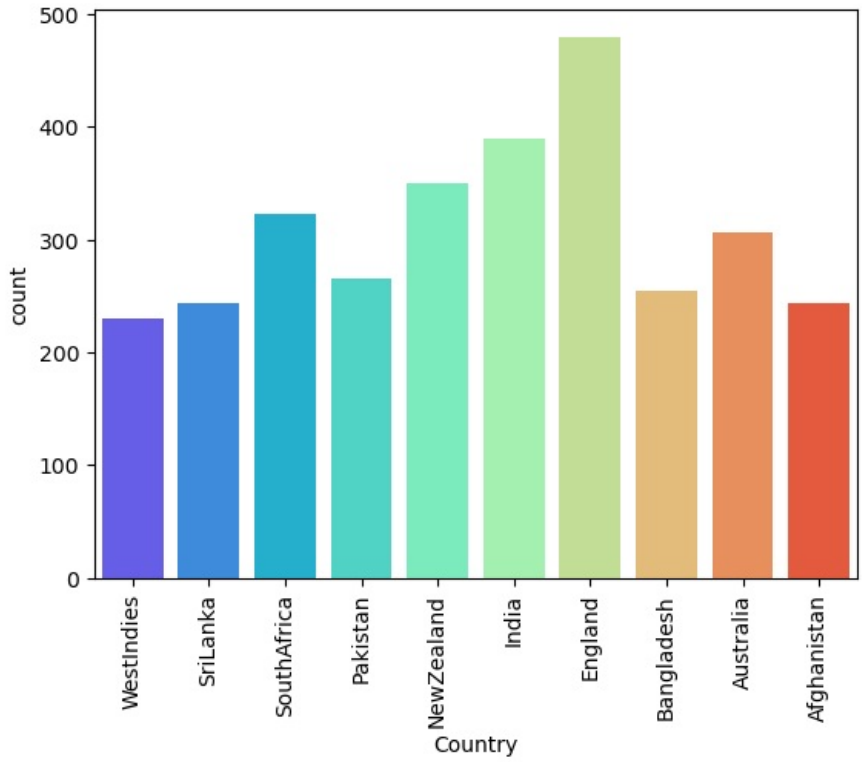
In [9]:
```
sns.pairplot(df)
plt.show()
```



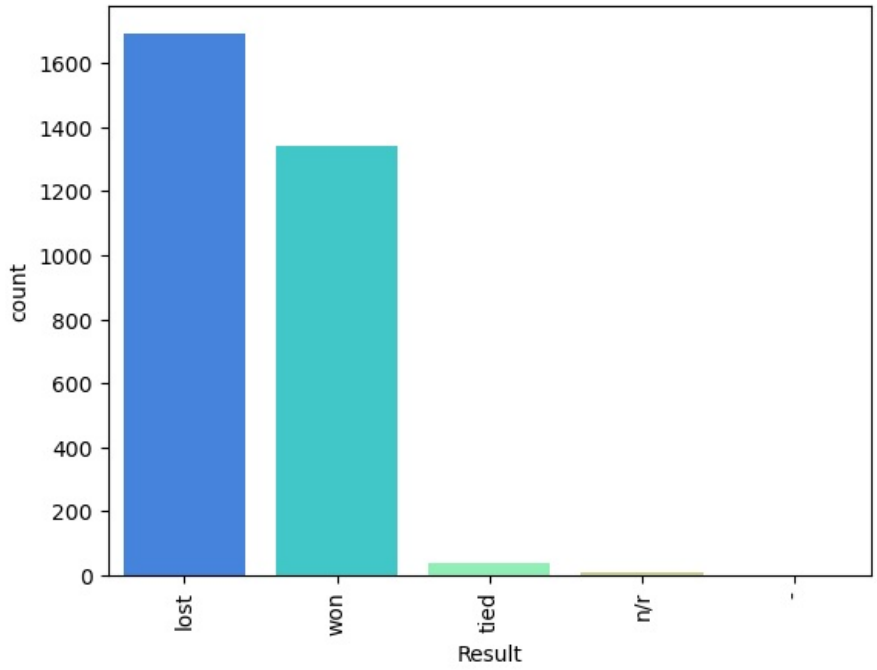
In [10]:
```
df['Country'].value_counts()
```

Out[10]:
```
England         480
India           390
NewZealand      350
SouthAfrica     323
Australia       306
Pakistan        265
Bangladesh      254
SriLanka        244
Afghanistan     244
WestIndies      230
Name: Country, dtype: int64
```

In [11]:
```
sns.countplot(x='Country', data=df, palette='rainbow')
plt.xticks(rotation=90)
plt.show()
```



In [12]:
```
sns.countplot(x='Result', data=df, palette='rainbow')
plt.xticks(rotation=90)
plt.show()
```

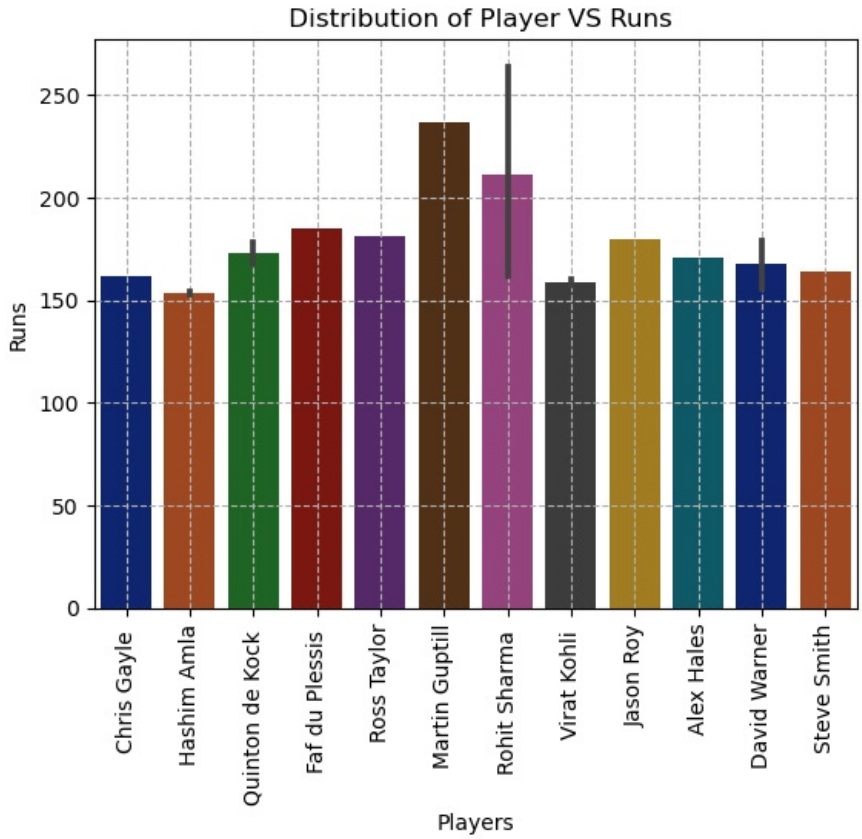


In [13]:
```
x = df.loc[df['Runs']>150]
arr = x['Player'].values
```

In [14]:
```
player  = x['Player'].values
runs = x['Runs'].values
print(player,runs)
```

```
['Chris Gayle ' 'Hashim Amla' 'Hashim Amla' 'Quinton de Kock '
 'Quinton de Kock ' 'Faf du Plessis ' 'Ross Taylor' 'Martin Guptill'
 'Rohit Sharma ' 'Rohit Sharma ' 'Rohit Sharma ' 'Virat Kohli '
 'Virat Kohli ' 'Jason Roy' 'Alex Hales' 'David Warner' 'David Warner'
 'Steve Smith'] [162 153 154 178 168 185 181 237 264 208 162 160 157 180 171 156 179 164]
```

In [15]:
```
sns.barplot(x=player,y=runs, palette='dark')
plt.xticks(rotation=90)
plt.title(' Distribution of Player VS Runs')
plt.xlabel('Players')
plt.ylabel('Runs')
plt.grid(linestyle='--')
plt.show()
```

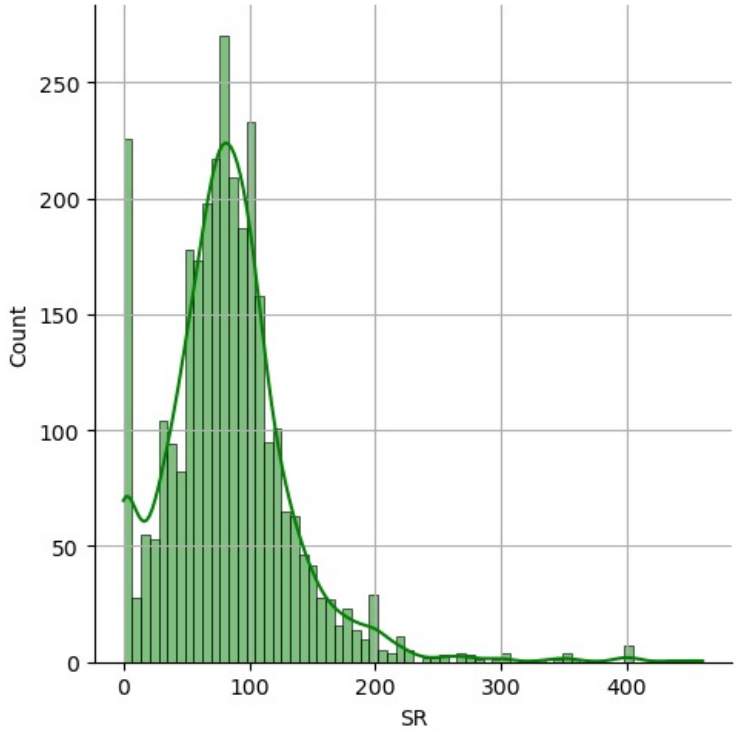


In [16]:
```
df.columns
```

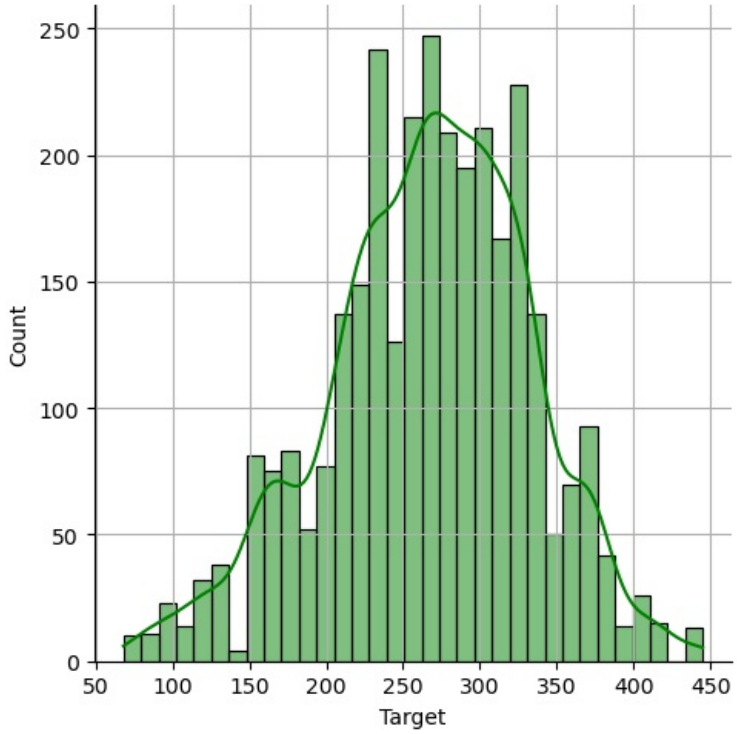
Out[16]:
```
Index(['ID', 'Player_ID', 'Player', 'Country', 'Opposition', 'Bat1', 'Runs',
       'BF', 'SR', '4s', '6s', 'Ground', 'Start Date', 'Match_ID', 'Team Runs',
       'Target', 'Overs', 'RPO', 'Inns', 'Result'],
      dtype='object')
```

https://github.com/varshithgitub/Cricket-Player-Performance-Prediction-Using-Machine-Learning/blob/main/Cricket%20Player%20Prediction.ipynb

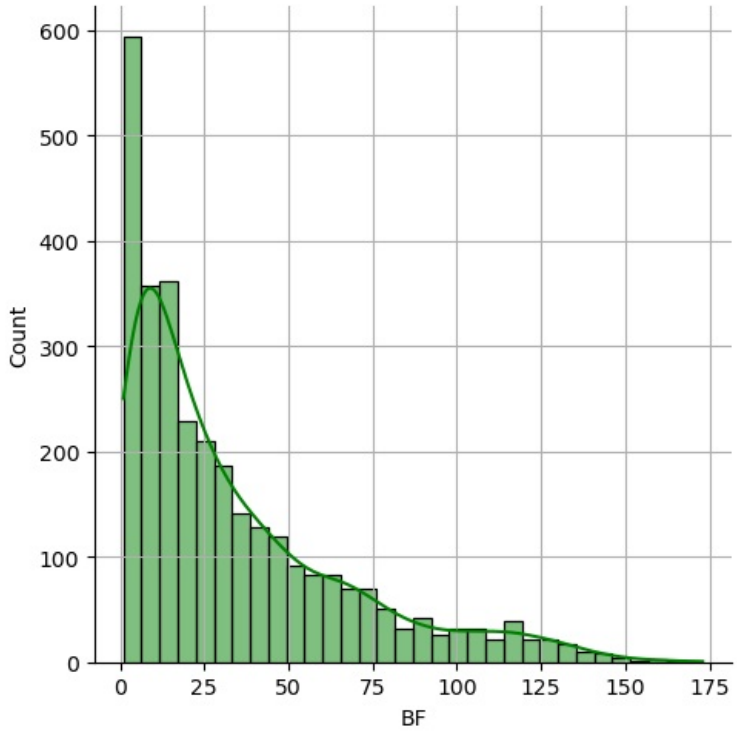
In [17]:
```
sns.displot(df['SR'], kde=True, color='green')
plt.grid()
plt.show()
```



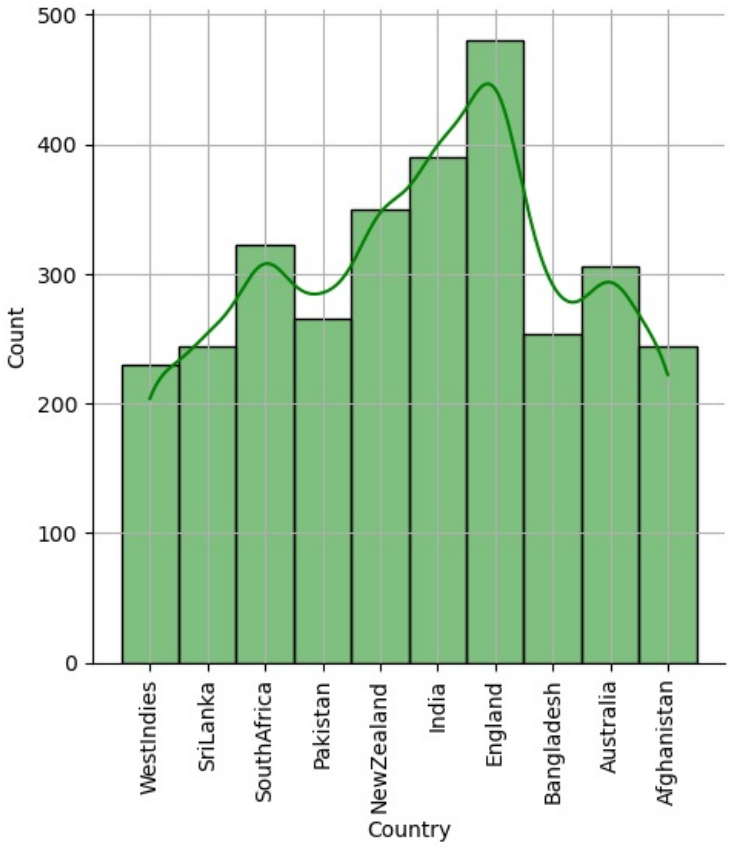
In [18]:
```
sns.displot(df['Target'],kde=True, color='green')
plt.grid()
plt.show()
```



In [19]:
```
sns.displot(df['BF'], kde=True, color='green')
plt.grid()
plt.show()
```



In [20]:
```
sns.displot(df['Country'], kde=True, color='green')
plt.grid()
plt.xticks(rotation =90)
plt.show()
```



In [21]:
```
df = pd.get_dummies(df)
```

In [22]:
```
df
```

Out[22]:

	ID	Player_ID	Runs	BF	SR	4s	6s	Target	Overs	Inns	...	RPO_8.11	RPO_8.19	RPO_8.32	RPO_8.67	RPO_9.45	Result_-
7	18	49619	1	1	100.00	0	0	323.0	49.4	2.0	...	0	0	0	0	0	0
9	27	49619	4	5	80.00	0	0	226.0	27.5	2.0	...	0	0	0	0	0	0
10	30	49619	0	3	0.00	0	0	361.0	43.4	2.0	...	0	0	0	0	0	0
11	31	49619	3	5	60.00	0	0	316.0	40.3	2.0	...	0	0	0	0	0	0
12	32	49619	7	17	41.17	0	0	332.0	46.1	2.0	...	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8810	11140	352048	4	5	80.00	0	0	250.0	41.2	2.0	...	0	0	0	0	0	0
8811	11141	352048	42	38	110.52	5	0	256.0	42.1	2.0	...	0	0	0	0	0	0
8812	11142	352048	10	7	142.85	1	0	258.0	49.3	2.0	...	0	0	0	0	0	0
8814	11144	352048	15	46	32.60	1	0	253.0	49.5	2.0	...	0	0	0	0	0	0
8815	11145	352048	46	61	75.40	6	1	162.0	41.5	2.0	...	0	0	0	0	0	0

3086 rows × 1750 columns

In [23]:
```
df.columns.tolist()
```

Out[23]:
```
['ID',
 'Player_ID',
 'Runs',
 'BF',
 'SR',
 '4s',
 '6s',
 'Target',
 'Overs',
 'Inns',
 'Player_Aaron Finch ',
 'Player_Abid Ali',
 'Player_Adam Zampa',
 'Player_Adil Rashid',
 'Player_Aftab Alam',
 'Player_Aiden Markram',
 'Player_Alex Carey ',
 'Player_Alex Hales',
 'Player_Andile Phehlukwayo',
 'Player_Andre Russell',
 'Player_Angelo Mathews',
 'Player_Anrich Nortje',
```

```
'Player_Asghar Afghan',
'Player_Ashley Nurse',
'Player_Avishka Fernando',
'Player_Babar Azam ',
'Player_Ben Stokes',
'Player_Bhuvneshwar Kumar',
'Player_Carlos Brathwaite',
'Player_Chris Gayle ',
'Player_Chris Morris',
'Player_Chris Woakes',
'Player_Colin Munro',
'Player_Colin de Grandhomme',
'Player_Dale Steyn',
'Player_Darren Bravo',
'Player_David Miller',
'Player_David Warner',
'Player_David Willey',
'Player_Dawlat Zadran',
'Player_Dhananjaya de Silva ',
'Player_Dimuth Karunaratne ',
'Player_Dinesh Karthik',
'Player_Dwaine Pretorius',
'Player_Eoin Morgan ',
'Player_Evin Lewis',
'Player_Fabian Allen',
'Player_Faf du Plessis ',
'Player_Faheem Ashraf',
'Player_Fakhar Zaman',
'Player_Glenn Maxwell',
'Player_Gulbadin Naib ',
'Player_Hamid Hassan',
'Player_Hardik Pandya',
'Player_Haris Sohail',
'Player_Hasan Ali',
'Player_Hashim Amla',
'Player_Hashmatullah Shahidi',
'Player_Hazratullah Zazai',
'Player_Henry Nicholls',
'Player_Imad Wasim',
'Player_Imam-ul-Haq',
'Player_Imran Tahir',
'Player_Ish Sodhi',
'Player_Isuru Udana',
'Player_JP Duminy',
'Player_James Neesham',
'Player_Jason Behrendorff',
'Player_Jason Holder ',
'Player_Jason Roy',
'Player_Jasprit Bumrah',
'Player_Jeevan Mendis',
'Player_Jeffrey Vandersay',
'Player_Jhye Richardson',
'Player_Joe Denly',
'Player_Joe Root',
'Player_Jonny Bairstow',
'Player_Jos Buttler ',
'Player_Junaid Khan',
'Player_K. L. Rahul',
'Player_Kagiso Rabada',
'Player_Kane Richardson',
'Player_Kane Williamson ',
'Player_Kedar Jadhav',
'Player_Kemar Roach',
'Player_Kuldeep Yadav',
'Player_Kusal Mendis',
'Player_Kusal Perera ',
'Player_Lahiru Thirimanne',
'Player_Lasith Malinga',
'Player_Liam Plunkett',
'Player_Liton Das',
'Player_Lockie Ferguson',
'Player_Lungi Ngidi',
'Player_MS Dhoni ',
'Player_Mahmudullah',
'Player_Marcus Stoinis',
'Player_Mark Wood',
'Player_Martin Guptill',
'Player_Mashrafe Mortaza ',
'Player_Matt Henry',
'Player_Milinda Siriwardana',
'Player_Mitchell Santner',
'Player_Mitchell Starc',
'Player_Moeen Ali',
'Player_Mohammad Hafeez',
'Player_Mohammad Hasnain',
'Player_Mohammad Mithun',
'Player_Mohammad Nabi',
'Player_Mohammad Saifuddin',
'Player_Mohammad Shahzad ',
```

```
'Player_Mohammed Shami',
'Player_Mosaddek Hossain',
'Player_Mujeeb Ur Rahman',
'Player_Mushfiqur Rahim ',
'Player_Mustafizur Rahman',
'Player_Najibullah Zadran',
'Player_Nathan Coulter-Nile',
'Player_Nathan Lyon',
'Player_Noor Ali Zadran',
'Player_Nuwan Pradeep',
'Player_Oshane Thomas',
'Player_Pat Cummins ',
'Player_Quinton de Kock ',
'Player_Rahmat Shah',
'Player_Rashid Khan ',
'Player_Rassie van der Dussen',
'Player_Ravindra Jadeja',
'Player_Rohit Sharma ',
'Player_Ross Taylor',
'Player_Rubel Hossain',
'Player_Sabbir Rahman',
'Player_Samiullah Shinwari',
'Player_Sarfaraz Ahmed ',
'Player_Shadab Khan',
'Player_Shaheen Afridi',
'Player_Shai Hope',
'Player_Shakib Al Hasan ',
'Player_Shannon Gabriel',
'Player_Shaun Marsh',
'Player_Sheldon Cottrell',
'Player_Shikhar Dhawan',
'Player_Shimron Hetmyer',
'Player_Shoaib Malik',
'Player_Soumya Sarkar',
'Player_Steve Smith',
'Player_Tabraiz Shamsi',
'Player_Tamim Iqbal',
'Player_Thisara Perera',
'Player_Tim Southee',
'Player_Tom Curran',
'Player_Tom Latham ',
'Player_Trent Boult',
'Player_Usman Khawaja',
'Player_Vijay Shankar',
'Player_Virat Kohli ',
'Player_Yuzvendra Chahal',
'Country_Afghanistan',
'Country_Australia',
'Country_Bangladesh',
'Country_England',
'Country_India',
'Country_NewZealand',
'Country_Pakistan',
'Country_SouthAfrica',
'Country_SriLanka',
'Country_WestIndies',
'Opposition_v Afghanistan',
'Opposition_v Australia',
'Opposition_v Bangladesh',
'Opposition_v England',
'Opposition_v Hong Kong',
'Opposition_v India',
'Opposition_v Ireland',
'Opposition_v Kenya',
'Opposition_v New Zealand',
'Opposition_v P.N.G.',
'Opposition_v Pakistan',
'Opposition_v Scotland',
'Opposition_v South Africa',
'Opposition_v Sri Lanka',
'Opposition_v U.A.E.',
'Opposition_v West Indies',
'Opposition_v Zimbabwe',
'Bat1_0',
'Bat1_0*',
'Bat1_1',
'Bat1_1*',
'Bat1_10',
'Bat1_10*',
'Bat1_100',
'Bat1_100*',
'Bat1_101',
'Bat1_101*',
'Bat1_102',
'Bat1_102*',
'Bat1_103',
'Bat1_103*',
'Bat1_104',
'Bat1_105',
```

```
'Bat1_105*',
'Bat1_106',
'Bat1_107',
'Bat1_108',
'Bat1_108*',
'Bat1_109',
'Bat1_11',
'Bat1_11*',
'Bat1_110',
'Bat1_110*',
'Bat1_111*',
'Bat1_112',
'Bat1_112*',
'Bat1_113',
'Bat1_113*',
'Bat1_114',
'Bat1_115',
'Bat1_115*',
'Bat1_116*',
'Bat1_117',
'Bat1_117*',
'Bat1_118',
'Bat1_118*',
'Bat1_119',
'Bat1_12',
'Bat1_12*',
'Bat1_120',
'Bat1_121',
'Bat1_122',
'Bat1_123',
'Bat1_123*',
'Bat1_124',
'Bat1_124*',
'Bat1_125',
'Bat1_125*',
'Bat1_126',
'Bat1_127',
'Bat1_128',
'Bat1_128*',
'Bat1_129',
'Bat1_13',
'Bat1_13*',
'Bat1_130',
'Bat1_130*',
'Bat1_131',
'Bat1_131*',
'Bat1_132',
'Bat1_133',
'Bat1_133*',
'Bat1_135',
'Bat1_136',
'Bat1_137',
'Bat1_137*',
'Bat1_138',
'Bat1_138*',
'Bat1_139*',
'Bat1_14',
'Bat1_14*',
'Bat1_140',
'Bat1_140*',
'Bat1_141*',
'Bat1_144',
'Bat1_145*',
'Bat1_146*',
'Bat1_147',
'Bat1_148',
'Bat1_15',
'Bat1_15*',
'Bat1_150',
'Bat1_153*',
'Bat1_154',
'Bat1_156',
'Bat1_157*',
'Bat1_16',
'Bat1_16*',
'Bat1_160*',
'Bat1_162',
'Bat1_164',
'Bat1_168*',
'Bat1_17',
'Bat1_17*',
'Bat1_171',
'Bat1_178',
'Bat1_179',
'Bat1_18',
'Bat1_18*',
'Bat1_180',
'Bat1_181*',
'Bat1_185',
```

```
'Bat1_19',
'Bat1_19*',
'Bat1_2',
'Bat1_2*',
'Bat1_20',
'Bat1_20*',
'Bat1_208*',
'Bat1_21',
'Bat1_21*',
'Bat1_22',
'Bat1_22*',
'Bat1_23',
'Bat1_23*',
'Bat1_237*',
'Bat1_24',
'Bat1_24*',
'Bat1_25',
'Bat1_25*',
'Bat1_26',
'Bat1_26*',
'Bat1_264',
'Bat1_27',
'Bat1_27*',
'Bat1_28',
'Bat1_28*',
'Bat1_29',
'Bat1_29*',
'Bat1_3',
'Bat1_3*',
'Bat1_30',
'Bat1_30*',
'Bat1_31',
'Bat1_31*',
'Bat1_32',
'Bat1_33',
'Bat1_33*',
'Bat1_34',
'Bat1_34*',
'Bat1_35',
'Bat1_35*',
'Bat1_36',
'Bat1_36*',
'Bat1_37',
'Bat1_37*',
'Bat1_38',
'Bat1_38*',
'Bat1_39',
'Bat1_39*',
'Bat1_4',
'Bat1_4*',
'Bat1_40',
'Bat1_40*',
'Bat1_41',
'Bat1_41*',
'Bat1_42',
'Bat1_42*',
'Bat1_43',
'Bat1_43*',
'Bat1_44',
'Bat1_44*',
'Bat1_45',
'Bat1_45*',
'Bat1_46',
'Bat1_46*',
'Bat1_47',
'Bat1_47*',
'Bat1_48',
'Bat1_48*',
'Bat1_49',
'Bat1_49*',
'Bat1_5',
'Bat1_5*',
'Bat1_50',
'Bat1_50*',
'Bat1_51',
'Bat1_51*',
'Bat1_52',
'Bat1_52*',
'Bat1_53',
'Bat1_54',
'Bat1_54*',
'Bat1_55',
'Bat1_55*',
'Bat1_56',
'Bat1_57',
'Bat1_57*',
'Bat1_58',
'Bat1_58*',
'Bat1_59',
```

```
'Bat1_59*',
'Bat1_6',
'Bat1_6*',
'Bat1_60',
'Bat1_60*',
'Bat1_61',
'Bat1_61*',
'Bat1_62',
'Bat1_62*',
'Bat1_63',
'Bat1_63*',
'Bat1_64',
'Bat1_64*',
'Bat1_65',
'Bat1_65*',
'Bat1_66',
'Bat1_67',
'Bat1_67*',
'Bat1_68',
'Bat1_68*',
'Bat1_69',
'Bat1_69*',
'Bat1_7',
'Bat1_7*',
'Bat1_70',
'Bat1_71',
'Bat1_71*',
'Bat1_72',
'Bat1_72*',
'Bat1_73',
'Bat1_73*',
'Bat1_74',
'Bat1_74*',
'Bat1_75',
'Bat1_75*',
'Bat1_76',
'Bat1_77',
'Bat1_77*',
'Bat1_78',
'Bat1_78*',
'Bat1_79',
'Bat1_79*',
'Bat1_8',
'Bat1_8*',
'Bat1_80',
'Bat1_80*',
'Bat1_81',
'Bat1_81*',
'Bat1_82',
'Bat1_82*',
'Bat1_83',
'Bat1_83*',
'Bat1_84',
'Bat1_84*',
'Bat1_85',
'Bat1_85*',
'Bat1_86',
'Bat1_86*',
'Bat1_87',
'Bat1_87*',
'Bat1_88',
'Bat1_88*',
'Bat1_89',
'Bat1_9',
'Bat1_9*',
'Bat1_90',
'Bat1_90*',
'Bat1_91',
'Bat1_91*',
'Bat1_92',
'Bat1_92*',
'Bat1_93',
'Bat1_94',
'Bat1_95',
'Bat1_95*',
'Bat1_96',
'Bat1_96*',
'Bat1_97',
'Bat1_97*',
'Bat1_98',
'Bat1_99',
'Bat1_99*',
'Ground_Abu Dhabi',
'Ground_Adelaide',
'Ground_Auckland',
'Ground_Basseterre',
'Ground_Belfast',
'Ground_Bengaluru',
'Ground_Benoni',
```

```
'Ground_Birmingham',
'Ground_Bloemfontein',
'Ground_Bridgetown',
'Ground_Brisbane',
'Ground_Bristol',
'Ground_Bulawayo',
'Ground_Canberra',
'Ground_Cape Town',
'Ground_Cardiff',
'Ground_Centurion',
'Ground_Chattogram',
'Ground_Chennai',
'Ground_Chester-le-Street',
'Ground_Christchurch',
'Ground_Colombo (RPS)',
'Ground_Cuttack',
'Ground_Dambulla',
'Ground_Dehradun',
'Ground_Delhi',
'Ground_Dhaka',
'Ground_Dharamsala',
'Ground_Dubai (DSC)',
'Ground_Dublin',
'Ground_Dublin (Malahide)',
'Ground_Dunedin',
'Ground_Durban',
'Ground_Edinburgh',
'Ground_Fatullah',
'Ground_Galle',
'Ground_Greater Noida',
'Ground_Gros Islet',
'Ground_Hambantota',
'Ground_Hamilton',
'Ground_Harare',
'Ground_Hobart',
'Ground_Hyderabad (Deccan)',
'Ground_ICCA Dubai',
'Ground_Indore',
'Ground_Jaipur',
'Ground_Johannesburg',
'Ground_Kanpur',
'Ground_Kimberley',
'Ground_Kingston',
'Ground_Kolkata',
'Ground_Kuala Lumpur',
'Ground_Lahore',
'Ground_Leeds',
"Ground_Lord's",
'Ground_Manchester',
'Ground_Melbourne',
'Ground_Mohali',
'Ground_Mount Maunganui',
'Ground_Mumbai',
'Ground_Mumbai (BS)',
'Ground_Nagpur',
'Ground_Napier',
'Ground_Nelson',
'Ground_North Sound',
'Ground_Nottingham',
'Ground_Paarl',
'Ground_Pallekele',
'Ground_Perth',
'Ground_Port Elizabeth',
'Ground_Port of Spain',
'Ground_Potchefstroom',
'Ground_Providence',
'Ground_Pune',
'Ground_Queenstown',
'Ground_Rajkot',
'Ground_Ranchi',
'Ground_Sharjah',
'Ground_Southampton',
"Ground_St George's",
'Ground_Sydney',
'Ground_The Oval',
'Ground_Visakhapatnam',
'Ground_Wellington',
'Start Date_1 Aug 2013',
'Start Date_1 Dec 2014',
'Start Date_1 Feb 2015',
'Start Date_1 Feb 2017',
'Start Date_1 Jan 2014',
'Start Date_1 Jun 2017',
'Start Date_1 Mar 2014',
'Start Date_1 Mar 2015',
'Start Date_1 May 2014',
'Start Date_1 Nov 2013',
'Start Date_1 Oct 2017',
'Start Date_10 Dec 2017',
```

```
'Start Date_10 Feb 2013',
'Start Date_10 Feb 2017',
'Start Date_10 Feb 2018',
'Start Date_10 Jul 2015',
'Start Date_10 Jun 2017',
'Start Date_10 Jun 2018',
'Start Date_10 Mar 2015',
'Start Date_10 Mar 2019',
'Start Date_10 May 2019',
'Start Date_11 Dec 2018',
'Start Date_11 Feb 2018',
'Start Date_11 Jan 2013',
'Start Date_11 Jun 2016',
'Start Date_11 Jun 2017',
'Start Date_11 May 2019',
'Start Date_11 Nov 2015',
'Start Date_11 Oct 2014',
'Start Date_11 Sep 2015',
'Start Date_12 Feb 2016',
'Start Date_12 Jan 2019',
'Start Date_12 Jul 2014',
'Start Date_12 Jul 2018',
'Start Date_12 Jun 2013',
'Start Date_12 Oct 2014',
'Start Date_12 Oct 2016',
'Start Date_13 Dec 2017',
'Start Date_13 Feb 2018',
'Start Date_13 Feb 2019',
'Start Date_13 Jan 2013',
'Start Date_13 Jan 2017',
'Start Date_13 Jan 2018',
'Start Date_13 Jun 2013',
'Start Date_13 Jun 2016',
'Start Date_13 Jun 2018',
'Start Date_13 Mar 2015',
'Start Date_13 Mar 2019',
'Start Date_13 Nov 2014',
'Start Date_13 Nov 2015',
'Start Date_13 Oct 2013',
'Start Date_13 Oct 2017',
'Start Date_13 Oct 2018',
'Start Date_14 Feb 2015',
'Start Date_14 Feb 2016',
'Start Date_14 Jan 2015',
'Start Date_14 Jan 2018',
'Start Date_14 Jul 2013',
'Start Date_14 Jul 2018',
'Start Date_14 Jun 2013',
'Start Date_14 Jun 2015',
'Start Date_14 Jun 2017',
'Start Date_14 Mar 2015',
'Start Date_14 Nov 2014',
'Start Date_14 Nov 2016',
'Start Date_14 Oct 2015',
'Start Date_14 Sep 2013',
'Start Date_15 Feb 2015',
'Start Date_15 Jan 2015',
'Start Date_15 Jan 2017',
'Start Date_15 Jan 2018',
'Start Date_15 Jan 2019',
'Start Date_15 Jul 2015',
'Start Date_15 Jun 2014',
'Start Date_15 Jun 2016',
'Start Date_15 Jun 2017',
'Start Date_15 Mar 2015',
'Start Date_15 May 2019',
'Start Date_15 Oct 2017',
'Start Date_15 Sep 2018',
'Start Date_16 Feb 2018',
'Start Date_16 Feb 2019',
'Start Date_16 Jan 2015',
'Start Date_16 Jul 2018',
'Start Date_16 Jun 2013',
'Start Date_16 Nov 2014',
'Start Date_16 Oct 2013',
'Start Date_16 Oct 2017',
'Start Date_16 Sep 2013',
'Start Date_16 Sep 2018',
'Start Date_17 Apr 2015',
'Start Date_17 Dec 2014',
'Start Date_17 Feb 2013',
'Start Date_17 Feb 2015',
'Start Date_17 Jan 2018',
'Start Date_17 May 2017',
'Start Date_17 Oct 2014',
'Start Date_17 Sep 2018',
'Start Date_18 Dec 2013',
'Start Date_18 Jan 2013',
'Start Date_18 Jan 2015',
```

'Start Date_18 Jan 2019',
'Start Date_18 Jul 2018',
'Start Date_18 Jun 2015',
'Start Date_19 Dec 2014',
'Start Date_19 Feb 2017',
'Start Date_19 Jan 2013',
'Start Date_19 Jan 2018',
'Start Date_19 Jul 2013',
'Start Date_19 Jul 2015',
'Start Date_19 Jul 2016',
'Start Date_19 Mar 2018',
'Start Date_19 May 2017',
'Start Date_19 May 2019',
'Start Date_19 Nov 2014',
'Start Date_19 Nov 2016',
'Start Date_2 Dec 2014',
'Start Date_2 Jan 2016',
'Start Date_2 Jul 2016',
'Start Date_2 Jul 2017',
'Start Date_2 Mar 2014',
'Start Date_2 Mar 2019',
'Start Date_2 Nov 2014',
'Start Date_2 Oct 2013',
'Start Date_2 Oct 2016',
'Start Date_2 Sep 2014',
'Start Date_20 Aug 2014',
'Start Date_20 Dec 2013',
'Start Date_20 Feb 2015',
'Start Date_20 Jan 2013',
'Start Date_20 Jan 2016',
'Start Date_20 Mar 2018',
'Start Date_20 Nov 2015',
'Start Date_20 Sep 2018',
'Start Date_21 Aug 2014',
'Start Date_21 Feb 2017',
'Start Date_21 Jan 2018',
'Start Date_21 Jun 2018',
'Start Date_21 Mar 2015',
'Start Date_21 Oct 2014',
'Start Date_21 Sep 2018',
'Start Date_22 Dec 2013',
'Start Date_22 Feb 2014',
'Start Date_22 Feb 2015',
'Start Date_22 Feb 2017',
'Start Date_22 Jan 2013',
'Start Date_22 Jan 2017',
'Start Date_22 Jan 2019',
'Start Date_22 Jul 2014',
'Start Date_22 Jul 2018',
'Start Date_22 May 2014',
'Start Date_22 Oct 2015',
'Start Date_22 Oct 2017',
'Start Date_23 Dec 2017',
'Start Date_23 Feb 2014',
'Start Date_23 Jan 2013',
'Start Date_23 Jan 2015',
'Start Date_23 Jan 2016',
'Start Date_23 Mar 2013',
'Start Date_23 Mar 2018',
'Start Date_23 Nov 2016',
'Start Date_23 Oct 2013',
'Start Date_24 Feb 2013',
'Start Date_24 Jul 2013',
'Start Date_24 Jun 2018',
'Start Date_24 Mar 2013',
'Start Date_24 Mar 2017',
'Start Date_24 May 2017',
'Start Date_24 Nov 2013',
'Start Date_24 Oct 2018',
'Start Date_24 Sep 2017',
'Start Date_25 Feb 2018',
'Start Date_25 Jan 2013',
'Start Date_25 Jan 2015',
'Start Date_25 Jan 2016',
'Start Date_25 Jan 2018',
'Start Date_25 Jan 2019',
'Start Date_25 Jun 2017',
'Start Date_25 Mar 2017',
'Start Date_25 Nov 2016',
'Start Date_25 Sep 2018',
'Start Date_26 Dec 2013',
'Start Date_26 Dec 2017',
'Start Date_26 Feb 2013',
'Start Date_26 Feb 2014',
'Start Date_26 Feb 2015',
'Start Date_26 Jan 2014',
'Start Date_26 Jan 2017',
'Start Date_26 Jan 2019',
'Start Date_26 Jul 2013',

```
'Start Date_26 Jun 2016',
'Start Date_26 Mar 2015',
'Start Date_26 May 2013',
'Start Date_26 Oct 2018',
'Start Date_26 Sep 2018',
'Start Date_27 Aug 2014',
'Start Date_27 Dec 2013',
'Start Date_27 Feb 2015',
'Start Date_27 Feb 2019',
'Start Date_27 Mar 2019',
'Start Date_27 May 2017',
'Start Date_27 Nov 2013',
'Start Date_27 Nov 2016',
'Start Date_27 Sep 2016',
'Start Date_28 Feb 2014',
'Start Date_28 Feb 2015',
'Start Date_28 Feb 2019',
'Start Date_28 Jan 2014',
'Start Date_28 Jan 2015',
'Start Date_28 Jul 2013',
'Start Date_28 Jul 2018',
'Start Date_28 Jun 2013',
'Start Date_28 Sep 2017',
'Start Date_28 Sep 2018',
'Start Date_29 Dec 2015',
'Start Date_29 Mar 2019',
'Start Date_29 May 2015',
'Start Date_29 May 2017',
'Start Date_29 Nov 2014',
'Start Date_29 Oct 2013',
'Start Date_29 Oct 2016',
'Start Date_29 Oct 2017',
'Start Date_29 Oct 2018',
'Start Date_3 Aug 2013',
'Start Date_3 Feb 2013',
'Start Date_3 Feb 2015',
'Start Date_3 Feb 2016',
'Start Date_3 Feb 2019',
'Start Date_3 Jan 2019',
'Start Date_3 Jun 2014',
'Start Date_3 Jun 2016',
'Start Date_3 Jun 2017',
'Start Date_3 Mar 2017',
'Start Date_3 Mar 2018',
'Start Date_3 May 2019',
'Start Date_3 Oct 2015',
'Start Date_3 Sep 2013',
'Start Date_3 Sep 2015',
'Start Date_30 Aug 2014',
'Start Date_30 Aug 2016',
'Start Date_30 Jan 2019',
'Start Date_30 Jun 2013',
'Start Date_30 Jun 2017',
'Start Date_30 Nov 2013',
'Start Date_30 Oct 2013',
'Start Date_30 Sep 2016',
'Start Date_30 Sep 2018',
'Start Date_31 Aug 2017',
'Start Date_31 Aug 2018',
'Start Date_31 Dec 2015',
'Start Date_31 Dec 2016',
'Start Date_31 Jan 2019',
'Start Date_31 Mar 2019',
'Start Date_31 May 2013',
'Start Date_31 Oct 2013',
'Start Date_4 Aug 2015',
'Start Date_4 Dec 2014',
'Start Date_4 Dec 2016',
'Start Date_4 Jan 2014',
'Start Date_4 Jan 2016',
'Start Date_4 Jun 2017',
'Start Date_4 Mar 2014',
'Start Date_4 Mar 2017',
'Start Date_4 Nov 2018',
'Start Date_4 Oct 2013',
'Start Date_4 Sep 2016',
'Start Date_5 Aug 2018',
'Start Date_5 Jan 2016',
'Start Date_5 Jan 2019',
'Start Date_5 Jul 2013',
'Start Date_5 Jun 2013',
'Start Date_5 Mar 2014',
'Start Date_5 Mar 2015',
'Start Date_5 May 2017',
'Start Date_5 Oct 2015',
'Start Date_5 Oct 2016',
'Start Date_5 Sep 2014',
'Start Date_5 Sep 2015',
'Start Date_6 Dec 2016',
```

'Start Date_6 Feb 2013',
'Start Date_6 Jan 2013',
'Start Date_6 Jan 2016',
'Start Date_6 Jan 2018',
'Start Date_6 Jul 2014',
'Start Date_6 Jul 2017',
'Start Date_6 Jun 2013',
'Start Date_6 Jun 2017',
'Start Date_6 Mar 2013',
'Start Date_6 Mar 2014',
'Start Date_6 Mar 2018',
'Start Date_6 Mar 2019',
'Start Date_6 Oct 2018',
'Start Date_6 Sep 2014',
'Start Date_7 Apr 2017',
'Start Date_7 Dec 2014',
'Start Date_7 Dec 2017',
'Start Date_7 Feb 2017',
'Start Date_7 Feb 2018',
'Start Date_7 Jul 2013',
'Start Date_7 Jun 2013',
'Start Date_7 Mar 2015',
'Start Date_7 Mar 2018',
'Start Date_7 Nov 2018',
'Start Date_7 Oct 2014',
'Start Date_8 Dec 2014',
'Start Date_8 Jan 2019',
'Start Date_8 Jun 2017',
'Start Date_8 Mar 2013',
'Start Date_8 Mar 2014',
'Start Date_8 Mar 2015',
'Start Date_8 Mar 2018',
'Start Date_8 Mar 2019',
'Start Date_8 Sep 2013',
'Start Date_9 Apr 2017',
'Start Date_9 Dec 2016',
'Start Date_9 Feb 2016',
'Start Date_9 Jul 2013',
'Start Date_9 Jun 2015',
'Start Date_9 Jun 2017',
'Start Date_9 Mar 2015',
'Start Date_9 Mar 2017',
'Start Date_9 Nov 2018',
'Start Date_9 Oct 2016',
'Match_ID_ODI # 3316',
'Match_ID_ODI # 3317',
'Match_ID_ODI # 3318',
'Match_ID_ODI # 3319',
'Match_ID_ODI # 3321',
'Match_ID_ODI # 3322',
'Match_ID_ODI # 3324',
'Match_ID_ODI # 3325',
'Match_ID_ODI # 3326',
'Match_ID_ODI # 3327',
'Match_ID_ODI # 3328',
'Match_ID_ODI # 3331',
'Match_ID_ODI # 3332',
'Match_ID_ODI # 3334',
'Match_ID_ODI # 3335',
'Match_ID_ODI # 3339',
'Match_ID_ODI # 3340',
'Match_ID_ODI # 3341',
'Match_ID_ODI # 3342',
'Match_ID_ODI # 3349',
'Match_ID_ODI # 3350',
'Match_ID_ODI # 3358',
'Match_ID_ODI # 3360',
'Match_ID_ODI # 3362',
'Match_ID_ODI # 3363',
'Match_ID_ODI # 3364',
'Match_ID_ODI # 3369',
'Match_ID_ODI # 3370',
'Match_ID_ODI # 3371',
'Match_ID_ODI # 3373',
'Match_ID_ODI # 3378',
'Match_ID_ODI # 3380',
'Match_ID_ODI # 3383',
'Match_ID_ODI # 3385',
'Match_ID_ODI # 3387',
'Match_ID_ODI # 3389',
'Match_ID_ODI # 3391',
'Match_ID_ODI # 3395',
'Match_ID_ODI # 3398',
'Match_ID_ODI # 3399',
'Match_ID_ODI # 3400',
'Match_ID_ODI # 3402',
'Match_ID_ODI # 3403',
'Match_ID_ODI # 3409',
'Match_ID_ODI # 3412',

```
'Match_ID_ODI # 3415',
'Match_ID_ODI # 3416',
'Match_ID_ODI # 3417',
'Match_ID_ODI # 3418',
'Match_ID_ODI # 3419',
'Match_ID_ODI # 3420',
'Match_ID_ODI # 3422',
'Match_ID_ODI # 3423',
'Match_ID_ODI # 3424',
'Match_ID_ODI # 3426',
'Match_ID_ODI # 3427',
'Match_ID_ODI # 3437',
'Match_ID_ODI # 3438',
'Match_ID_ODI # 3440',
'Match_ID_ODI # 3441',
'Match_ID_ODI # 3445',
'Match_ID_ODI # 3446',
'Match_ID_ODI # 3447',
'Match_ID_ODI # 3449',
'Match_ID_ODI # 3450',
'Match_ID_ODI # 3451',
'Match_ID_ODI # 3452',
'Match_ID_ODI # 3463',
'Match_ID_ODI # 3465',
'Match_ID_ODI # 3471',
'Match_ID_ODI # 3472',
'Match_ID_ODI # 3474',
'Match_ID_ODI # 3476',
'Match_ID_ODI # 3478',
'Match_ID_ODI # 3479',
'Match_ID_ODI # 3482',
'Match_ID_ODI # 3483',
'Match_ID_ODI # 3484',
'Match_ID_ODI # 3485',
'Match_ID_ODI # 3486',
'Match_ID_ODI # 3487',
'Match_ID_ODI # 3492',
'Match_ID_ODI # 3496',
'Match_ID_ODI # 3497',
'Match_ID_ODI # 3500',
'Match_ID_ODI # 3502',
'Match_ID_ODI # 3505',
'Match_ID_ODI # 3509',
'Match_ID_ODI # 3510',
'Match_ID_ODI # 3516',
'Match_ID_ODI # 3519',
'Match_ID_ODI # 3520',
'Match_ID_ODI # 3522',
'Match_ID_ODI # 3523',
'Match_ID_ODI # 3525',
'Match_ID_ODI # 3526',
'Match_ID_ODI # 3530',
'Match_ID_ODI # 3533',
'Match_ID_ODI # 3534',
'Match_ID_ODI # 3535',
'Match_ID_ODI # 3536',
'Match_ID_ODI # 3539',
'Match_ID_ODI # 3544',
'Match_ID_ODI # 3545',
'Match_ID_ODI # 3546',
'Match_ID_ODI # 3548',
'Match_ID_ODI # 3557',
'Match_ID_ODI # 3559',
'Match_ID_ODI # 3560',
'Match_ID_ODI # 3562',
'Match_ID_ODI # 3563',
'Match_ID_ODI # 3564',
'Match_ID_ODI # 3570',
'Match_ID_ODI # 3571',
'Match_ID_ODI # 3576',
'Match_ID_ODI # 3577',
'Match_ID_ODI # 3579',
'Match_ID_ODI # 3583',
'Match_ID_ODI # 3588',
'Match_ID_ODI # 3590',
'Match_ID_ODI # 3591',
'Match_ID_ODI # 3593',
'Match_ID_ODI # 3597',
'Match_ID_ODI # 3598',
'Match_ID_ODI # 3599',
'Match_ID_ODI # 3600',
'Match_ID_ODI # 3602',
'Match_ID_ODI # 3604',
'Match_ID_ODI # 3607',
'Match_ID_ODI # 3609',
'Match_ID_ODI # 3610',
'Match_ID_ODI # 3614',
'Match_ID_ODI # 3616',
...]
```

In [24]:
```
x = df.drop('Runs', axis=1)
y = df['Runs']
```

In [25]:
```
x
```

Out[25]:

	ID	Player_ID	BF	SR	4s	6s	Target	Overs	Inns	Player_Aaron Finch	...	RPO_8.11	RPO_8.19	RPO_8.32	RPO_8.67	RPO_9.45	R
7	18	49619	1	100.00	0	0	323.0	49.4	2.0	0	...	0	0	0	0	0	
9	27	49619	5	80.00	0	0	226.0	27.5	2.0	0	...	0	0	0	0	0	
10	30	49619	3	0.00	0	0	361.0	43.4	2.0	0	...	0	0	0	0	0	
11	31	49619	5	60.00	0	0	316.0	40.3	2.0	0	...	0	0	0	0	0	
12	32	49619	17	41.17	0	0	332.0	46.1	2.0	0	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
8810	11140	352048	5	80.00	0	0	250.0	41.2	2.0	0	...	0	0	0	0	0	
8811	11141	352048	38	110.52	5	0	256.0	42.1	2.0	0	...	0	0	0	0	0	
8812	11142	352048	7	142.85	1	0	258.0	49.3	2.0	0	...	0	0	0	0	0	
8814	11144	352048	46	32.60	1	0	253.0	49.5	2.0	0	...	0	0	0	0	0	
8815	11145	352048	61	75.40	6	1	162.0	41.5	2.0	0	...	0	0	0	0	0	

3086 rows × 1749 columns

In [26]:
```
y
```

Out[26]:
```
7        1
9        4
10       0
11       3
12       7
        ..
8810     4
8811    42
8812    10
8814    15
8815    46
Name: Runs, Length: 3086, dtype: int64
```

In [27]:
```
from sklearn.model_selection import train_test_split
xtrain , xtest, ytrain, ytest = train_test_split(x,y, test_size=0.2, random_state=101)
```

In [28]:
```
print('xtrain',xtrain.shape)
print('xtest',xtest.shape)
print('ytrain',ytrain.shape)
print('ytest',ytest.shape)
```

```
xtrain (2468, 1749)
xtest (618, 1749)
ytrain (2468,)
ytest (618,)
```

In [29]:
```
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
from sklearn.tree import plot_tree

tree = DecisionTreeRegressor()
tree
```

Out[29]:
```
DecisionTreeRegressor()
```

In [30]:
```
tree.fit(xtrain,ytrain)
```

Out[30]:
```
DecisionTreeRegressor()
```

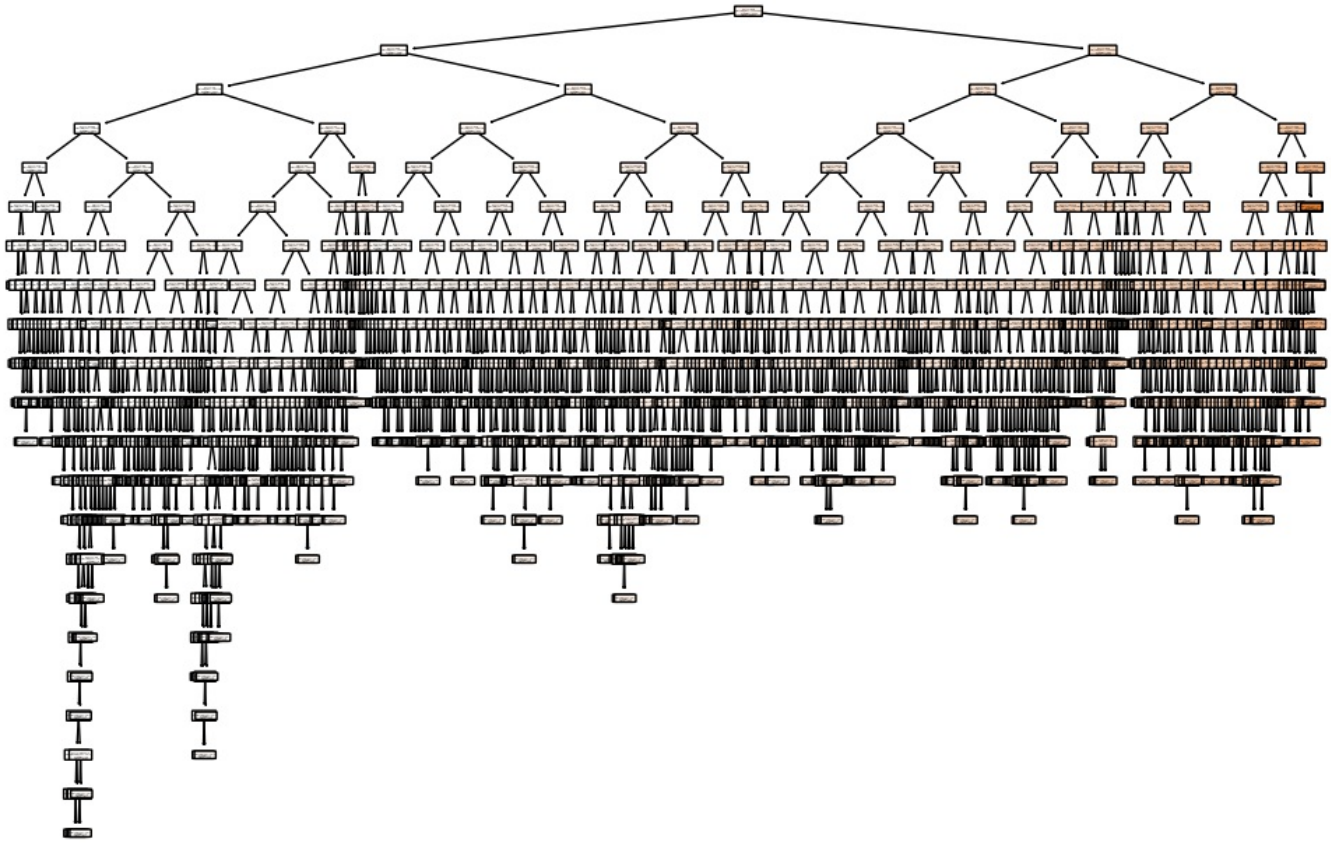
In [31]:
```
pred1 = tree.predict(xtest)
pred1
```

Out[31]:
```
array([ 56.,  77.,  72.,  29.,  20.,  30., 130.,  11., 100.,  80.,  22.,
        44.,  14.,   0.,  46.,   1.,   1.,  19.,  69.,  36.,   5.,  15.,
        25.,   5.,  47.,  17.,   2., 100.,  10.,  10.,  63.,  21.,  54.,
         1.,  39.,  14.,  28.,  10.,   0.,   6.,  33.,  35.,  49.,  19.,
         2.,   1.,  10.,  50.,   0.,   0.,   8.,   6.,  43.,  56.,  14.,
        25.,  21.,   9.,  13.,  10.,  28.,   5.,  10., 178.,  27.,  39.,
         4.,   4.,  41.,  50.,  15.,  40.,  20.,  13.,  12.,  49.,   7.,
         5.,   7.,   2.,   9.,   4.,   0.,   0.,   3.,  28.,  20., 100.,
        25.,  12.,  23.,  11.,   0.,   6.,  38.,   0.,   0.,  48.,   2.,
         5.,   2.,   2.,   8.,   1.,   6., 107.,  19.,  18.,   1.,  18.,
         2.,  23.,  61.,  62.,  44.,   1.,   0., 138.,  42.,  22.,   0.,
        12.,   1.,  60.,   9., 122.,  65., 110.,  90.,   0.,  64.,   9.,
        38.,   0.,  29.,   0.,   1.,   0.,  22.,  23.,  19.,  79.,  16.,
         7.,  51., 109.,  19.,  59.,   0.,  12.,  16.,  95.,  33.,  16.,
        22.,  19.,  32.,  72.,  31.,   9.,   1.,  39.,   1.,  20.,  15.,
        16.,  98.,  65.,  17.,  14.,  33.,  22.,   0.,  30.,   8.,   4.,
         0.,   0.,  37.,  13.,   1.,   0.,   8.,   5.,  47.,   0.,  57.,
        22.,  13.,  99., 121.,  42.,  27., 108.,  14., 101.,  41.,   0.,
        22.,   5., 135.,  17., 102.,  47., 264.,  83.,   1.,  33.,  10.,
         3.,  15.,   3.,  16.,  26.,  13.,  10.,  73.,  19.,  11.,   1.,
        22.,   0.,   0.,   6.,   4.,  90.,   7.,   1.,  24.,  12.,  16.,
         0.,   2.,  49.,  21.,  33.,   1., 105.,   0.,   5.,  12.,  13.,
        72.,   4.,   8.,  22.,  15.,   3.,   4.,  13.,   0.,  39.,  88.,
         1.,  15.,  13., 109.,  31.,   0.,  12.,  34.,  11.,   4.,  11.,
       100.,  32.,  51.,  53.,   0.,   7.,  63.,  36.,  28.,  21.,  36.,
         5.,  44.,  33.,  16.,  11.,   0.,   4.,  17.,  40.,   3.,  22.,
         8.,  46.,  63., 100.,   6.,  11.,  12.,  51.,  44.,   0.,   1.,
        13.,  95.,   6.,  12.,   7.,   4.,   4.,   2.,  71.,  81.,  11.,
        62.,  20., 113., 135.,   0.,  33.,  98.,  33.,   0.,  18.,  74.,
        34.,  44.,  33.,  22.,  87.,  22.,   1.,   0.,  35.,  24.,  59.,
         1.,  16.,   1.,  32.,  30.,  55.,  26.,   9.,  65.,  36.,   2.,
        28.,   1.,  27.,   2.,   6.,   9.,  10.,  90.,  44., 102.,  13.,
        71.,  68.,  32.,   0.,  47.,   5.,   1.,  14.,  59.,  22.,   4.,
         3.,  14.,  17., 100., 179.,   1.,  58.,  16.,  60.,   9.,   0.,
         0.,  82.,  11.,  21., 116.,  38.,  10.,  10.,   3.,  59.,  47.,
        35.,   8.,   0.,  18.,   5.,  19.,  53.,   7.,   5.,   1.,  18.,
        11.,   3.,  64.,  22.,  16.,  11.,   8.,   6.,   1.,   1.,   1.,
        10.,  10.,   0.,  19., 100.,  22.,   3.,  51.,  71.,  27.,  12.,
         1.,  24.,  36.,  54.,  45.,  12.,  22.,   6.,  50.,   6.,  28.,
        25.,   3.,  32.,   1.,  34.,  48.,  49.,  53.,  60.,  43.,   8.,
        38.,  35.,   8.,   0.,  35.,   0.,  56.,  53.,  31.,   0.,  43.,
        87.,   8.,  15.,   3.,   2.,  58.,  33.,   1.,  25.,   2.,  22.,
       100.,  20.,  18.,  10.,  25.,   4.,  17.,  19.,  26.,  30.,  63.,
        13.,  30.,   1.,  10.,   8.,  67.,   7.,  32.,   0.,  55.,  32.,
         7., 125.,  12., 105.,   6.,  54.,   1.,  26.,   2.,  15., 118.,
        22.,   1.,  11.,   1.,   2.,  51.,   2.,   2.,   9.,   0.,  40.,
        13.,   5.,  20.,  32.,  19.,   8.,  14.,   7.,  13.,  41.,  23.,
        12.,   4.,   4.,   7.,   0.,  85.,   9.,  14.,  14.,  17., 125.,
        53.,  20.,  18.,  44.,  12.,   2.,  39.,  46.,   0.,  16.,  12.,
        29.,  87.,  11.,  12.,  22.,  81.,   7.,  40.,   1.,  81.,   4.,
         9.,  39.,   2.,  99.,  21.,  87.,  20.,  10.,   8.,  23.,  15.,
         9.,  44.,  14.,  12.,   2.,  56., 156.,  14.,   7.,  77.,  36.,
        90.,   2.,  97.,  14.,  13.,  14.,  57.,   5.,  39.,  34.,   9.,
         3.,   2.,  83.,   0.,  42.,  10.,  15.,  14.,  51.,  31.,  71.,
        14.,   9.,  75.,  13.,  25.,   2.,   2.,   3.,  87.,  14.,  84.,
         0., 118.,   9.,   0.,  56.,  36.,   6.,  41.,  14.,  15.,  85.,
         4.,  32.])
```

In [32]:
```
mse = mean_squared_error(ytest, pred1)
print('MEAN_SQUARE_ERROR of DECISIONTREEREGRESSOR:', mse)
```

```
MEAN_SQUARE_ERROR of DECISIONTREEREGRESSOR: 27.432038834951456
```

In [33]:
```
# Visualize the decision tree
plt.figure(figsize=(12, 8))
plot_tree(tree, filled=True, rounded=True)
plt.show()
```



In [34]:
```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from xgboost import XGBRegressor
from sklearn import metrics
```

In [35]:
```
knn = KNeighborsRegressor()
knn.fit(xtrain,ytrain)

rf = RandomForestClassifier()
rf.fit(xtrain,ytrain)

lr = LinearRegression()
lr.fit(xtrain,ytrain)

svm = SVR()
svm.fit(xtrain,ytrain)

xgb = XGBRegressor()
xgb.fit(xtrain,ytrain)
print( knn,'\n',rf,'\n',lr,'\n',svm,'\n',xgb)
```

```
KNeighborsRegressor()
 RandomForestClassifier()
 LinearRegression()
 SVR()
 XGBRegressor(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, random_state=None, ...)
```

In [36]:
```
pred1 = knn.predict(xtest)
pred2 = rf.predict(xtest)
pred3 = lr.predict(xtest)
pred4 = svm.predict(xtest)
pred5 = xgb.predict(xtest)
```

In [37]:
```
knn_score = metrics.r2_score(ytest,pred1)
rf_score = metrics.r2_score(ytest,pred2)
lr_score = metrics.r2_score(ytest,pred3)
xgb_score = metrics.r2_score(ytest,pred5)
```

In [38]:
```
print('KNeighborsRegressor: ',knn score)
```

```
print('RandomForestClassifier: ',rf_score)
print('LinearRegression: ',lr_score)
print('SVR: ',svm_score)
print('XGBRegressor: ',xgb_score)
```

```
KNeighborsRegressor:  0.6511594431761305
RandomForestClassifier:  0.9290952668673655
LinearRegression:  0.8985349976007593
SVR:  -0.07099628912959544
XGBRegressor:  0.993137777721142
```

In [39]:
```
final_data = pd.DataFrame({'Models':['KNN','RF','LR','SVR','XGB'],
                           'R2_SCORE':[knn_score,rf_score,lr_score,svm_score,xgb_score]})
```

In [40]:
```
final_data
```

Out[40]:

	Models	R2_SCORE
0	KNN	0.651159
1	RF	0.929095
2	LR	0.898535
3	SVR	-0.070996
4	XGB	0.993138

In [41]:
```
sns.barplot(x=final_data['Models'], y = final_data['R2_SCORE'])
plt.xlabel('Models')
plt.ylabel('R2 Score')
plt.title('R2 Score for Different Models')

plt.show()
```