Cricket Prediction

```
Prediction  of One day Cricket Match

Data Used from website Cricsheet
```

Reading required Libraries

In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [7]:
```
data=pd.read_csv('E:\Sumayya-Donot Delete\Internship\internship2\odi.csv')
```

In [8]:
```
data.head()
```

Out[8]:

	mid	date	venue	bat_team	bowl_team	batsman	bowler	runs	wickets	overs	ru
0	1	2006-06-13	Civil Service Cricket Club, Stormont	England	Ireland	ME Trescothick	DT Johnston	0	0	0.1	
1	1	2006-06-13	Civil Service Cricket Club, Stormont	England	Ireland	ME Trescothick	DT Johnston	0	0	0.2	
2	1	2006-06-13	Civil Service Cricket Club, Stormont	England	Ireland	ME Trescothick	DT Johnston	4	0	0.3	
3	1	2006-06-13	Civil Service Cricket Club, Stormont	England	Ireland	ME Trescothick	DT Johnston	6	0	0.4	
4	1	2006-06-13	Civil Service Cricket Club, Stormont	England	Ireland	ME Trescothick	DT Johnston	6	0	0.5	

Some Eda on data

```
1. Any null values ?
2. which team has taken highest score?
```

In [63]:

```python
data.isnull().sum().sum()
data['total'].max()
data[data['total']==444]
```

Out[63]:

	mid	date	venue	bat_team	bowl_team	batsman	bowler	runs	wickets	overs
305335	1034	2016-08-30	Trent Bridge	England	Pakistan	JJ Roy	Mohammad Amir	0	0	0.1
305336	1034	2016-08-30	Trent Bridge	England	Pakistan	JJ Roy	Mohammad Amir	0	0	0.2
305337	1034	2016-08-30	Trent Bridge	England	Pakistan	JJ Roy	Mohammad Amir	0	0	0.3
305338	1034	2016-08-30	Trent Bridge	England	Pakistan	JJ Roy	Mohammad Amir	0	0	0.4
305339	1034	2016-08-30	Trent Bridge	England	Pakistan	JJ Roy	Mohammad Amir	4	0	0.5
...	...	...	...	...	...	...	...	...	...	..
305638	1034	2016-08-30	Trent Bridge	England	Pakistan	EJG Morgan	Hasan Ali	439	3	49.2
305639	1034	2016-08-30	Trent Bridge	England	Pakistan	EJG Morgan	Hasan Ali	440	3	49.3
305640	1034	2016-08-30	Trent Bridge	England	Pakistan	JC Buttler	Hasan Ali	440	3	49.4
305641	1034	2016-08-30	Trent Bridge	England	Pakistan	JC Buttler	Hasan Ali	440	3	49.5
305642	1034	2016-08-30	Trent Bridge	England	Pakistan	JC Buttler	Hasan Ali	444	3	49.6

308 rows × 15 columns

In [20]:
```
data[data['mid']==888]
```

Out[20]:

	mid	date	venue	bat_team	bowl_team	batsman	bowler	runs	wickets	overs
261537	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	RG Sharma	SL Malinga	0	0	0.1
261538	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	RG Sharma	SL Malinga	0	0	0.2
261539	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	RG Sharma	SL Malinga	0	0	0.3
261540	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	RG Sharma	SL Malinga	1	0	0.4
261541	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	S Dhawan	SL Malinga	1	0	0.5
...	...	...	...	...	...	...	...	...	...	...
261838	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	Mohammed Shami	SL Malinga	261	9	49.2
261839	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	RA Jadeja	SL Malinga	262	9	49.3
261840	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	Mohammed Shami	SL Malinga	262	9	49.4
261841	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	Mohammed Shami	SL Malinga	263	9	49.5
261842	888	2014-02-28	Khan Shaheb Osman Ali Stadium	India	Sri Lanka	RA Jadeja	SL Malinga	264	9	49.6

306 rows × 15 columns

Features used

```
Runs
Wicket
Over
Striker
Non Striker
```

Label

```
Total
```

In [24]:
```
X=data.iloc[:,[7,8,9,12,13]].values #runs,wicket,over,striker,non striker
y=data.iloc[:,14].values #runs
```

Splitting data into train and test Model

In [35]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_stat
```

In [36]:
```
X_test
```

Out[36]:
```
array([[58. ,  6. , 20.2, 22. ,  0. ],
       [81. ,  5. , 23.5, 29. , 14. ],
       [19. ,  0. ,  5.5, 13. ,  2. ],
       ...,
       [80. ,  0. , 10.3, 39. , 38. ],
       [ 5. ,  0. ,  0.3,  0. ,  0. ],
       [13. ,  0. ,  4.2,  7. ,  6. ]])
```

Scaling data before applying ML algorithms

In [37]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Prediction using Linear Regreesion

```
In [38]:  from sklearn.linear_model import LinearRegression
          lin=LinearRegression()
          lin.fit(X_train,y_train)
```

```
Out[38]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=Fa
         lse)
```

Checking the goodness of model using R2 value

```
In [42]:  y_pred=lin.predict(X_test)
          score=lin.score(X_test,y_test)*100
          print("R-squared value:" , score)
          y_pred
```

```
         R-squared value: 52.737657811129445
```

```
Out[42]: array([164.08674918, 192.56966018, 255.74830646, ..., 308.90763905,
                258.37939597, 253.98483871])
```

Prediction using Random Forest

```
In [43]:  from sklearn.ensemble import RandomForestRegressor
          lin = RandomForestRegressor(n_estimators=100,max_features=None)
          lin.fit(X_train,y_train)
```

```
Out[43]: RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                               max_depth=None, max_features=None, max_leaf_nodes=Non
         e,
                               max_samples=None, min_impurity_decrease=0.0,
                               min_impurity_split=None, min_samples_leaf=1,
                               min_samples_split=2, min_weight_fraction_leaf=0.0,
                               n_estimators=100, n_jobs=None, oob_score=False,
                               random_state=None, verbose=0, warm_start=False)
```

checking goodness of model using R2 and we got a better accuracy with random Forest

```
In [44]:  y_pred=lin.predict(X_test)
          score=lin.score(X_test,y_test)*100
          print("R-squared value:" , score)
          y_pred
```

```
         R-squared value: 79.5323792837045
```

```
Out[44]: array([124.03      , 215.92      , 254.06      , ..., 306.52      ,
                226.57192166, 293.97105952])
```

checking with random inputs to check fitness

In [49]:
```
import numpy as np
new_prediction = lin.predict(sc.transform(np.array([[80,0,13,50,50]])))
print("Prediction score:" , new_prediction)
```

```
Prediction score: [329.53666667]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: