```
# Importing Necessary Libraries
import pandas as pd
import numpy as np
import seaborn as sns
```

```
#Importing dataset
ipl_df = pd.read_csv('ipl_data.csv')
print(f"Dataset successfully Imported of Shape : {ipl_df.shape}")
```

```
Dataset successfully Imported of Shape : (76014, 15)
```

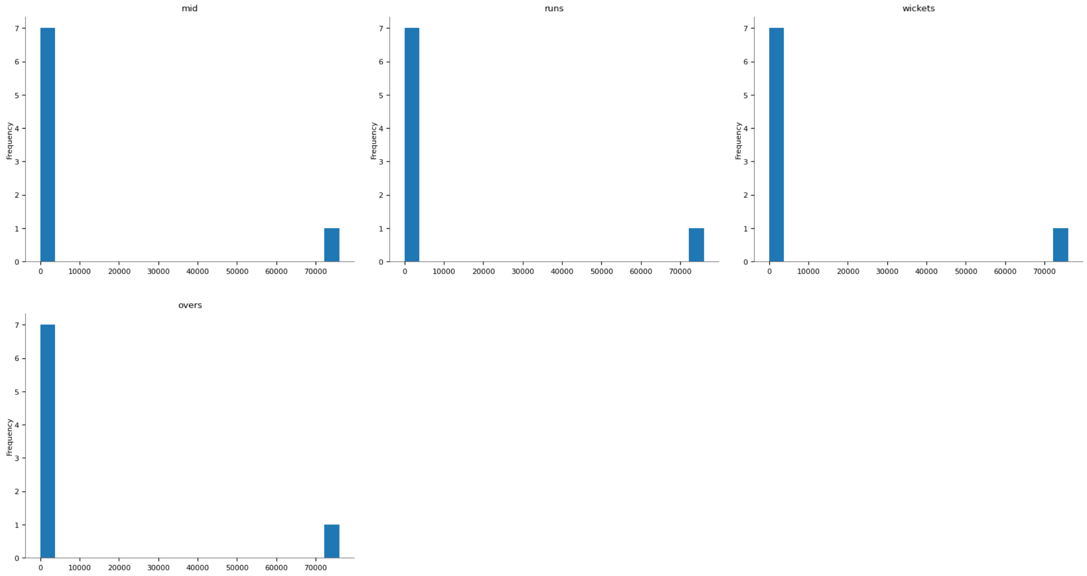
```
# First 5 Columns Data
ipl_df.head()
```

	mid	date	venue	bat_team	bowl_team	batsman	bowler	runs	wickets
0	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	SC Ganguly	P Kumar	1	0
1	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullum	P Kumar	1	0
2	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullum	P Kumar	2	0
3	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullum	P Kumar	2	0
4	1	2008-04-18	M Chinnaswamy Stadium	Kolkata Knight Riders	Royal Challengers Bangalore	BB McCullum	P Kumar	2	0

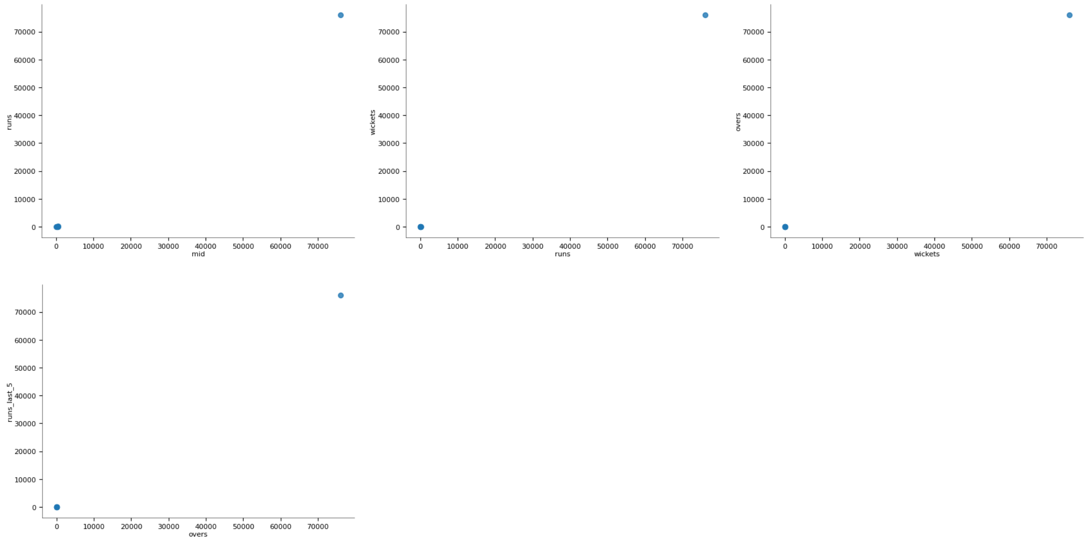
```
# Describing the ipl_dfset
ipl_df.describe()
```

	mid	runs	wickets	overs	runs_last_5	wickets_
count	76014.000000	76014.000000	76014.000000	76014.000000	76014.000000	7601
mean	308.627740	74.889349	2.415844	9.783068	33.216434	1
std	178.156878	48.823327	2.015207	5.772587	14.914174	1
min	1.000000	0.000000	0.000000	0.000000	0.000000	(
25%	154.000000	34.000000	1.000000	4.600000	24.000000	(
50%	308.000000	70.000000	2.000000	9.600000	34.000000	1
75%	463.000000	111.000000	4.000000	14.600000	43.000000	2
max	617.000000	263.000000	10.000000	19.600000	113.000000	7

Distributions



2-d distributions



Values

```
# Information about Each Column
ipl_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 76014 entries, 0 to 76013
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   mid             76014 non-null  int64  
 1   date            76014 non-null  object 
 2   venue           76014 non-null  object 
 3   bat_team        76014 non-null  object 
 4   bowl_team       76014 non-null  object 
 5   batsman         76014 non-null  object 
 6   bowler          76014 non-null  object 
 7   runs            76014 non-null  int64  
 8   wickets         76014 non-null  int64  
 9   overs           76014 non-null  float64
 10  runs_last_5     76014 non-null  int64  
 11  wickets_last_5  76014 non-null  int64  
 12  striker         76014 non-null  int64  
 13  non-striker     76014 non-null  int64  
```

```
      14  total           76014 non-null  int64
     dtypes: float64(1), int64(8), object(6)
     memory usage: 8.7+ MB
```

```
# Number of Unique Values in each column
ipl_df.nunique()
```

```
     mid               617
     date              442
     venue              35
     bat_team           14
     bowl_team          14
     batsman           411
     bowler            329
     runs              252
     wickets            11
     overs             140
     runs_last_5       102
     wickets_last_5      8
     striker           155
     non-striker        88
     total             138
     dtype: int64
```

```
# ipl_df types of all Columns
ipl_df.dtypes
```

```
     mid                 int64
     date               object
     venue              object
     bat_team           object
     bowl_team          object
     batsman            object
     bowler             object
     runs                int64
     wickets             int64
     overs             float64
     runs_last_5         int64
     wickets_last_5      int64
     striker             int64
     non-striker         int64
     total               int64
     dtype: object
```

```
# Names of all columns
ipl_df.columns
```

```
     Index(['mid', 'date', 'venue', 'bat_team', 'bowl_team', 'batsman', 'bowler',
            'runs', 'wickets', 'overs', 'runs_last_5', 'wickets_last_5',
     'striker',
            'non-striker', 'total'],
           dtype='object')
```

```
irrelevant = ['mid', 'date', 'venue','batsman', 'bowler', 'striker', 'non-striker']
print(f'Before Removing Irrelevant Columns : {ipl_df.shape}')
ipl_df = ipl_df.drop(irrelevant, axis=1) # Drop Irrelevant Columns
```

```
print(f'After Removing Irrelevant Columns : {ipl_df.shape}')
ipl_df.head()
```

KEEPING ONLY CONSISTENT TEAMS

```
# Define Consistent Teams
const_teams = ['Kolkata Knight Riders', 'Chennai Super Kings', 'Rajasthan Royals',
              'Mumbai Indians', 'Kings XI Punjab', 'Royal Challengers Bangalore',
              'Delhi Daredevils', 'Sunrisers Hyderabad']
```

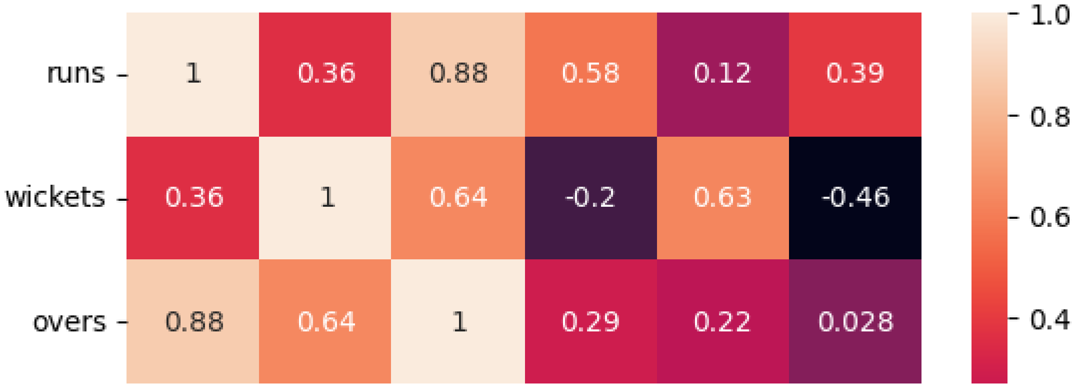
```
     Before Removing Irrelevant Columns : (76014, 15)
```

```
print(f'Before Removing Inconsistent Teams : {ipl_df.shape}')
ipl_df = ipl_df[(ipl_df['bat_team'].isin(const_teams)) & (ipl_df['bowl_team'].isin(cons
print(f'After Removing Irrelevant Columns : {ipl_df.shape}')
print(f"Consistent Teams : \n{ipl_df['bat_team'].unique()}")
ipl_df.head()
```

```
print(f'Before Removing Overs : {ipl_df.shape}')
ipl_df = ipl_df[ipl_df['overs'] >= 5.0]
print(f'After Removing Overs : {ipl_df.shape}')
ipl_df.head()
```

```
from seaborn import heatmap
heatmap(data=ipl_df.corr(), annot=True)
```

```
<ipython-input-16-8fde0e13fc28>:2: FutureWarning: The default value of numeri
  heatmap(data=ipl_df.corr(), annot=True)
<Axes: >
```



```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
le = LabelEncoder()
for col in ['bat_team', 'bowl_team']:
  ipl_df[col] = le.fit_transform(ipl_df[col])
ipl_df.head()
```

	bat_team	bowl_team	runs	wickets	overs	runs_last_5	wickets_last_5	to
32	3	6	61	0	5.1	59	0	1
33	3	6	61	1	5.2	59	1	1
34	3	6	61	1	5.3	59	1	1
35	3	6	61	1	5.4	59	1	1
36	3	6	61	1	5.5	58	1	1

```
from sklearn.compose import ColumnTransformer
columnTransformer = ColumnTransformer([('encoder',
                                        OneHotEncoder(),
                                        [0, 1])],
                                      remainder='passthrough')
```

```
ipl_df = np.array(columnTransformer.fit_transform(ipl_df))
```

```
cols = ['batting_team_Chennai Super Kings', 'batting_team_Delhi Daredevils', 'batting_t
              'batting_team_Kolkata Knight Riders', 'batting_team_Mumbai Indians', 'bat
              'batting_team_Royal Challengers Bangalore', 'batting_team_Sunrisers Hyder
              'bowling_team_Chennai Super Kings', 'bowling_team_Delhi Daredevils', 'bow
              'bowling_team_Kolkata Knight Riders', 'bowling_team_Mumbai Indians', 'bow
              'bowling_team_Royal Challengers Bangalore', 'bowling_team_Sunrisers Hyder
       'runs_last_5', 'wickets_last_5', 'total']
df = pd.DataFrame(ipl_df, columns=cols)
```

```
# Encoded Data
df.head()
```

	batting_team_Chennai Super Kings	batting_team_Delhi Daredevils	batting_team_Kings XI Punjab	batting_team Knigh
0	0.0	0.0	0.0	
1	0.0	0.0	0.0	
2	0.0	0.0	0.0	
3	0.0	0.0	0.0	
4	0.0	0.0	0.0	

5 rows × 22 columns

```
features = df.drop(['total'], axis=1)
labels = df['total']
```

```
from sklearn.model_selection import train_test_split
train_features, test_features, train_labels, test_labels = train_test_split(features, l
print(f"Training Set : {train_features.shape}\nTesting Set : {test_features.shape}")
```

```
Training Set : (32086, 21)
Testing Set : (8022, 21)
```

```
models = dict()
```

```
from sklearn.ensemble import RandomForestRegressor
forest = RandomForestRegressor()
# Train Model
forest.fit(train_features, train_labels)
```

```
▾ RandomForestRegressor
RandomForestRegressor()
```

```
def score_predict(batting_team, bowling_team, runs, wickets, overs, runs_last_5, wicket
  prediction_array = []
  # Batting Team
  if batting_team == 'Chennai Super Kings':
    prediction_array = prediction_array + [1,0,0,0,0,0,0,0]
  elif batting_team == 'Delhi Daredevils':
    prediction_array = prediction_array + [0,1,0,0,0,0,0,0]
  elif batting_team == 'Kings XI Punjab':
    prediction_array = prediction_array + [0,0,1,0,0,0,0,0]
  elif batting_team == 'Kolkata Knight Riders':
    prediction_array = prediction_array + [0,0,0,1,0,0,0,0]
  elif batting_team == 'Mumbai Indians':
    prediction_array = prediction_array + [0,0,0,0,1,0,0,0]
  elif batting_team == 'Rajasthan Royals':
    prediction_array = prediction_array + [0,0,0,0,0,1,0,0]
  elif batting_team == 'Royal Challengers Bangalore':
    prediction_array = prediction_array + [0,0,0,0,0,0,1,0]
  elif batting_team == 'Sunrisers Hyderabad':
    prediction_array = prediction_array + [0,0,0,0,0,0,0,1]
  # Bowling Team
  if bowling_team == 'Chennai Super Kings':
    prediction_array = prediction_array + [1,0,0,0,0,0,0,0]
  elif bowling_team == 'Delhi Daredevils':
    prediction_array = prediction_array + [0,1,0,0,0,0,0,0]
  elif bowling_team == 'Kings XI Punjab':
    prediction_array = prediction_array + [0,0,1,0,0,0,0,0]
  elif bowling_team == 'Kolkata Knight Riders':
    prediction_array = prediction_array + [0,0,0,1,0,0,0,0]
  elif bowling_team == 'Mumbai Indians':
    prediction_array = prediction_array + [0,0,0,0,1,0,0,0]
  elif bowling_team == 'Rajasthan Royals':
    prediction_array = prediction_array + [0,0,0,0,0,1,0,0]
  elif bowling_team == 'Royal Challengers Bangalore':
    prediction_array = prediction_array + [0,0,0,0,0,0,1,0]
  elif bowling_team == 'Sunrisers Hyderabad':
    prediction_array = prediction_array + [0,0,0,0,0,0,0,1]
  prediction_array = prediction_array + [runs, wickets, overs, runs_last_5, wickets_las
  prediction_array = np.array([prediction_array])
  pred = model.predict(prediction_array)
  return int(round(pred[0]))
```

TEST 1

```
batting_team='Mumbai Indians'
bowling_team='Kings XI Punjab'
score = score_predict(batting_team, bowling_team, overs=12.3, runs=113, wickets=2, runs_
print(f'Predicted Score : {score} || Actual Score : 176')
```

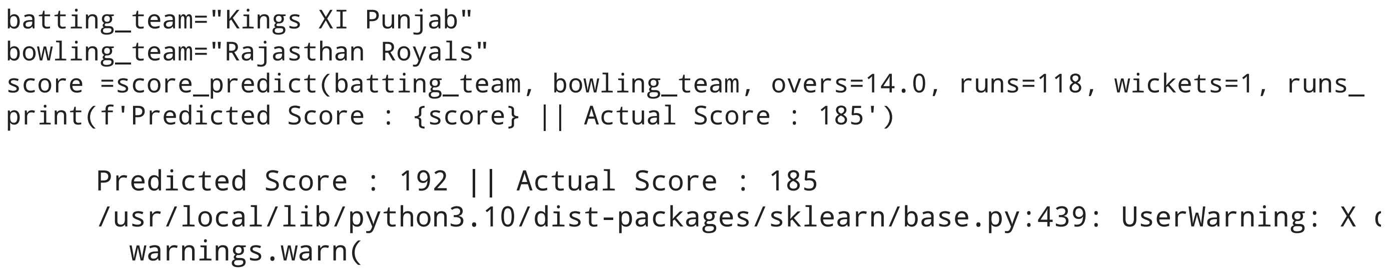
```
Predicted Score : 187 || Actual Score : 176
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X
  warnings.warn(
```

TEST 2

```
batting_team='Delhi Daredevils'
bowling_team='Chennai Super Kings'
score = score_predict(batting_team, bowling_team, overs=10.2, runs=68, wickets=3, runs_
print(f'Predicted Score : {score} || Actual Score : 147')
```

```
Predicted Score : 146 || Actual Score : 147
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X
  warnings.warn(
```

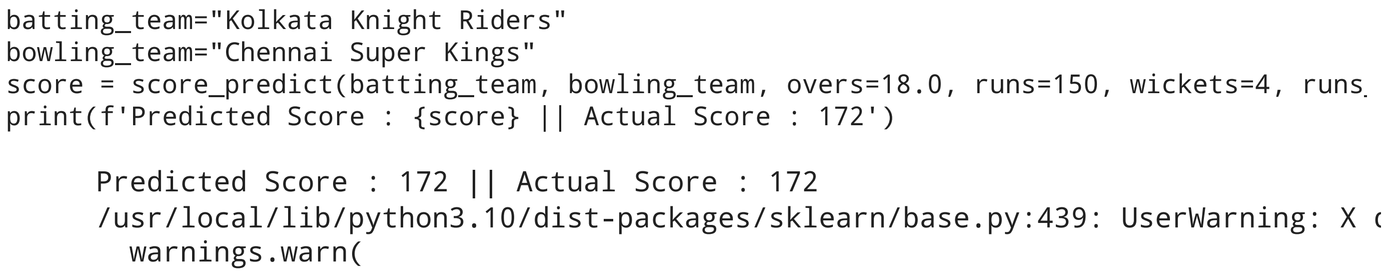
TEST **3**



```
batting_team="Kings XI Punjab"
bowling_team="Rajasthan Royals"
score =score_predict(batting_team, bowling_team, overs=14.0, runs=118, wickets=1, runs_
print(f'Predicted Score : {score} || Actual Score : 185')
```

```
Predicted Score : 192 || Actual Score : 185
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X c
  warnings.warn(
```

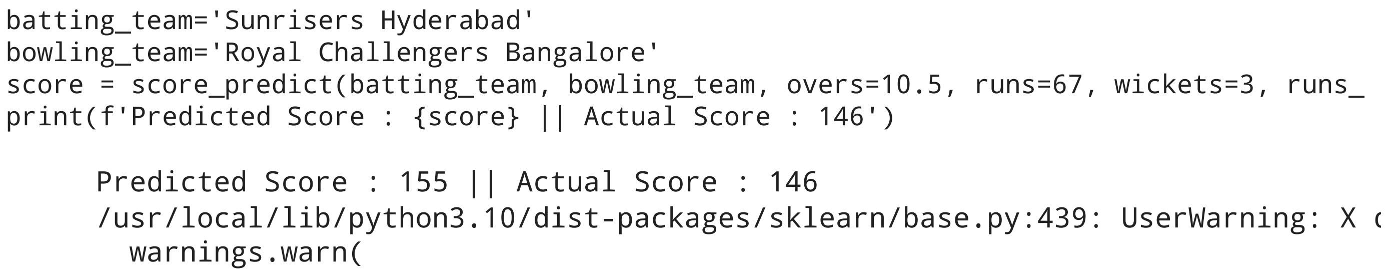
TEST 4



```
batting_team="Kolkata Knight Riders"
bowling_team="Chennai Super Kings"
score = score_predict(batting_team, bowling_team, overs=18.0, runs=150, wickets=4, runs_
print(f'Predicted Score : {score} || Actual Score : 172')
```

```
Predicted Score : 172 || Actual Score : 172
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X c
  warnings.warn(
```

TEST 5



```
batting_team='Sunrisers Hyderabad'
bowling_team='Royal Challengers Bangalore'
score = score_predict(batting_team, bowling_team, overs=10.5, runs=67, wickets=3, runs_
print(f'Predicted Score : {score} || Actual Score : 146')
```

```
Predicted Score : 155 || Actual Score : 146
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X c
  warnings.warn(
```