CAPSTONE PROJECT - WP_SP_01

Aniket Ganguly

PGPDSBA – JULY 2023

SOLUTION

1. **Problem Understanding**
   a) Defining problem statement b) Need of the study/project c) Understanding business/social opportunity

a) Problem Statement

The aim of this project is to predict the performance of the Indian cricket team based on historical match data. Specifically, we seek to predict the outcome of matches (win/loss) using a variety of features related to the match conditions, player statistics, and other relevant variables.

b) Need of the study/project

Performance Prediction: Accurately predicting the outcomes of cricket matches can be highly valuable for various stakeholders:

- **Team Management:** Helps in strategizing and making informed decisions regarding team selection, game plans, and training focus.
- **Betting Agencies:** Assists in setting odds and understanding the likely outcomes of games.
- **Fans and Analysts:** Provides deeper insights and engagement with the sport by understanding the factors that influence match outcomes.

Improving Team Performance: By identifying key factors that contribute to winning or losing, the team can focus on improving specific areas, leading to better overall performance.

Resource Allocation: Helps in better allocation of resources and efforts towards aspects that have a significant impact on the match outcomes.

c) Understanding business/social opportunity

Commercial Opportunities:

- **Sports Analytics Services:** Developing a predictive analytics service that can be sold to cricket boards, franchises, and other organizations.
- **Betting Markets:** Enhancing the accuracy of betting odds and markets.

Enhancing Fan Engagement:

- **Fantasy Leagues:** Providing more accurate predictions can make fantasy leagues more competitive and engaging.

- **Broadcasting Enhancements:** Use predictions to enhance live commentary and analysis during the matches.

Data-Driven Decision Making:

- **Cricket Boards and Teams:** Use data to make strategic decisions regarding player selections, training regimes, and match tactics.

2. **Data Report**
a) Understanding how data was collected in terms of time, frequency and methodology b) Visual inspection of data (rows, columns, descriptive details) c) Understanding of attributes (variable info, renaming if required)

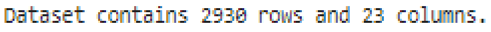
a) Understanding how data was collected in terms of time, frequency and methodology.

To effectively analyse and predict the performance of the Indian cricket team, understanding the context of the data collection is crucial. Here are key points to consider:

- **Time Period:** The data has been collected for cricket matches for all the seasons – Summer, Winter, Rainy.
- **Frequency:** The data has been collected for every match, every series and for specific tournaments as the matches has been played on different locations against different countries.

b) Visual inspection of data (rows, columns, descriptive details)

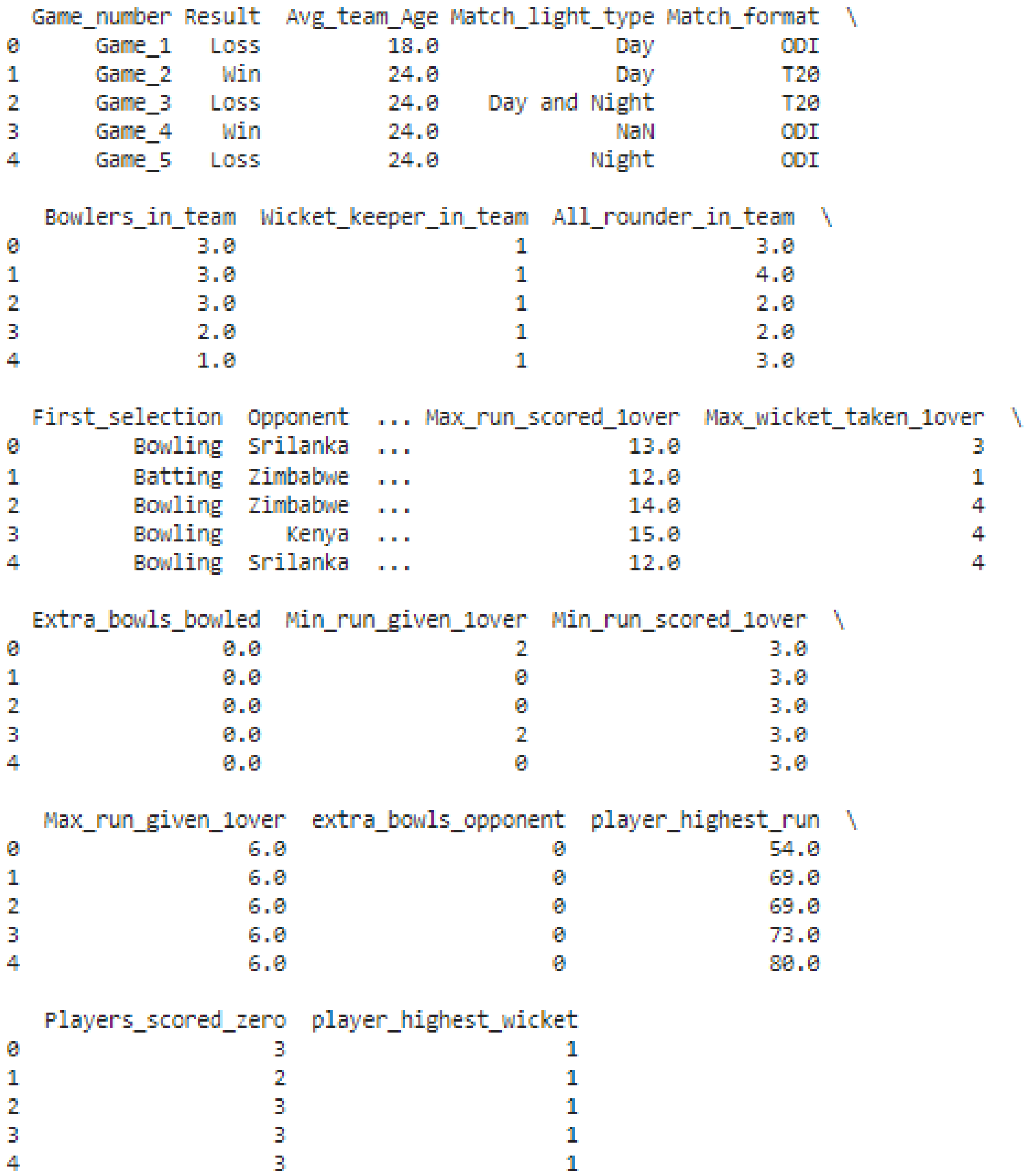
- **Total Rows and Columns:**



```
Dataset contains 2930 rows and 23 columns.
```

Fig 1

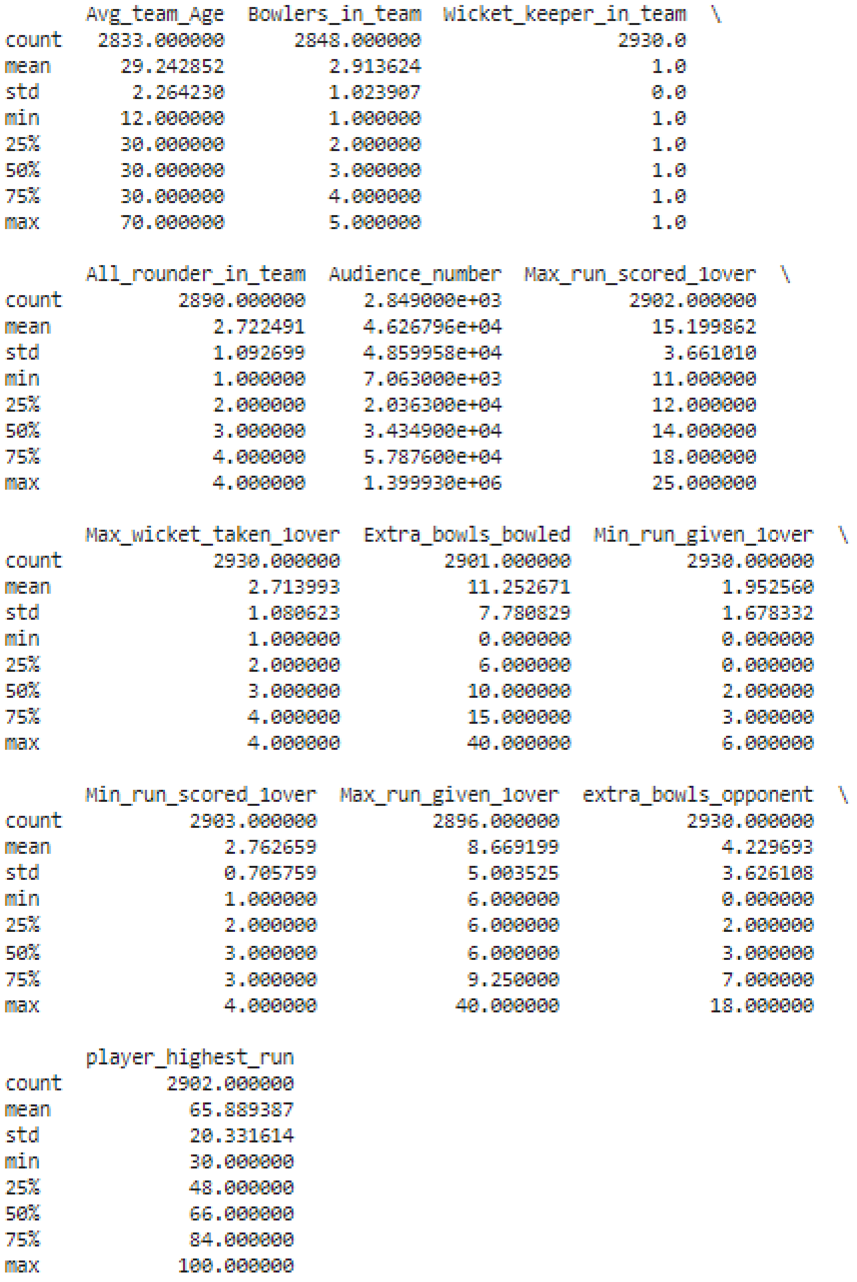
- **Top 5 rows:**



```
  Game_number Result  Avg_team_Age Match_light_type Match_format  \
0      Game_1   Loss          18.0              Day          ODI   
1      Game_2    Win          24.0              Day          T20   
2      Game_3   Loss          24.0    Day and Night          T20   
3      Game_4    Win          24.0              NaN          ODI   
4      Game_5   Loss          24.0            Night          ODI   

   Bowlers_in_team  Wicket_keeper_in_team  All_rounder_in_team  \
0              3.0                      1                  3.0   
1              3.0                      1                  4.0   
2              3.0                      1                  2.0   
3              2.0                      1                  2.0   
4              1.0                      1                  3.0   

  First_selection  Opponent  ... Max_run_scored_1over Max_wicket_taken_1over  \
0         Bowling  Srilanka  ...                 13.0                      3   
1         Batting  Zimbabwe  ...                 12.0                      1   
2         Bowling  Zimbabwe  ...                 14.0                      4   
3         Bowling     Kenya  ...                 15.0                      4   
4         Bowling  Srilanka  ...                 12.0                      4   

  Extra_bowls_bowled  Min_run_given_1over  Min_run_scored_1over  \
0                0.0                    2                   3.0   
1                0.0                    0                   3.0   
2                0.0                    0                   3.0   
3                0.0                    2                   3.0   
4                0.0                    0                   3.0   

   Max_run_given_1over  extra_bowls_opponent  player_highest_run  \
0                  6.0                     0                54.0   
1                  6.0                     0                69.0   
2                  6.0                     0                69.0   
3                  6.0                     0                73.0   
4                  6.0                     0                80.0   

   Players_scored_zero  player_highest_wicket
0                    3                      1
1                    2                      1
2                    3                      1
3                    3                      1
4                    3                      1
```

Fig 2

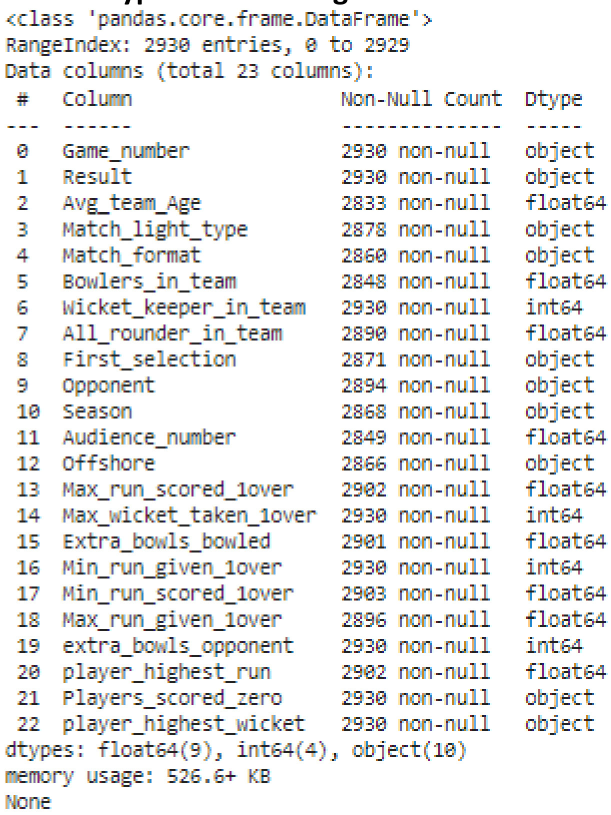
- **Statistical Summary of the dataset**



```
      Avg_team_Age  Bowlers_in_team  Wicket_keeper_in_team  \
count  2833.000000      2848.000000                 2930.0   
mean     29.242852         2.913624                    1.0   
std       2.264230         1.023907                    0.0   
min      12.000000         1.000000                    1.0   
25%      30.000000         2.000000                    1.0   
50%      30.000000         3.000000                    1.0   
75%      30.000000         4.000000                    1.0   
max      70.000000         5.000000                    1.0   

      All_rounder_in_team  Audience_number  Max_run_scored_1over  \
count         2890.000000     2.849000e+03           2902.000000   
mean             2.722491     4.626796e+04             15.199862   
std              1.092699     4.859958e+04              3.661010   
min              1.000000     7.063000e+03             11.000000   
25%              2.000000     2.036300e+04             12.000000   
50%              3.000000     3.434900e+04             14.000000   
75%              4.000000     5.787600e+04             18.000000   
max              4.000000     1.399930e+06             25.000000   

      Max_wicket_taken_1over  Extra_bowls_bowled  Min_run_given_1over  \
count            2930.000000         2901.000000          2930.000000   
mean                2.713993           11.252671             1.952560   
std                 1.080623            7.780829             1.678332   
min                 1.000000            0.000000             0.000000   
25%                 2.000000            6.000000             0.000000   
50%                 3.000000           10.000000             2.000000   
75%                 4.000000           15.000000             3.000000   
max                 4.000000           40.000000             6.000000   

      Min_run_scored_1over  Max_run_given_1over  extra_bowls_opponent  \
count          2903.000000          2896.000000           2930.000000   
mean              2.762659             8.669199              4.229693   
std               0.705759             5.003525              3.626108   
min               1.000000             6.000000              0.000000   
25%               2.000000             6.000000              2.000000   
50%               3.000000             6.000000              3.000000   
75%               3.000000             9.250000              7.000000   
max               4.000000            40.000000             18.000000   

      player_highest_run  
count         2902.000000  
mean            65.889387  
std             20.331614  
min             30.000000  
25%             48.000000  
50%             66.000000  
75%             84.000000  
max            100.000000  
```

Fig 3

- **Datatypes and Missing values in the columns**

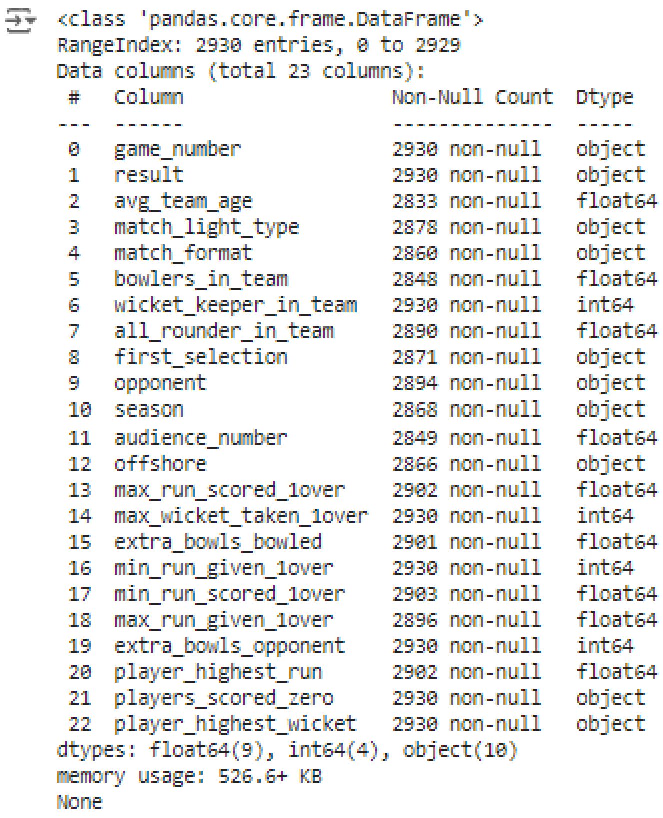


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2930 entries, 0 to 2929
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Game_number            2930 non-null   object 
 1   Result                 2930 non-null   object 
 2   Avg_team_Age           2833 non-null   float64
 3   Match_light_type       2878 non-null   object 
 4   Match_format           2860 non-null   object 
 5   Bowlers_in_team        2848 non-null   float64
 6   Wicket_keeper_in_team  2930 non-null   int64  
 7   All_rounder_in_team    2890 non-null   float64
 8   First_selection        2871 non-null   object 
 9   Opponent               2894 non-null   object 
 10  Season                 2868 non-null   object 
 11  Audience_number        2849 non-null   float64
 12  Offshore               2866 non-null   object 
 13  Max_run_scored_1over   2902 non-null   float64
 14  Max_wicket_taken_1over 2930 non-null   int64  
 15  Extra_bowls_bowled     2901 non-null   float64
 16  Min_run_given_1over    2930 non-null   int64  
 17  Min_run_scored_1over   2903 non-null   float64
 18  Max_run_given_1over    2896 non-null   float64
 19  extra_bowls_opponent   2930 non-null   int64  
 20  player_highest_run     2902 non-null   float64
 21  Players_scored_zero    2930 non-null   object 
 22  player_highest_wicket  2930 non-null   object 
dtypes: float64(9), int64(4), object(10)
memory usage: 526.6+ KB
None
```

Fig 4

c) Understanding of attributes (variable info, renaming if required)

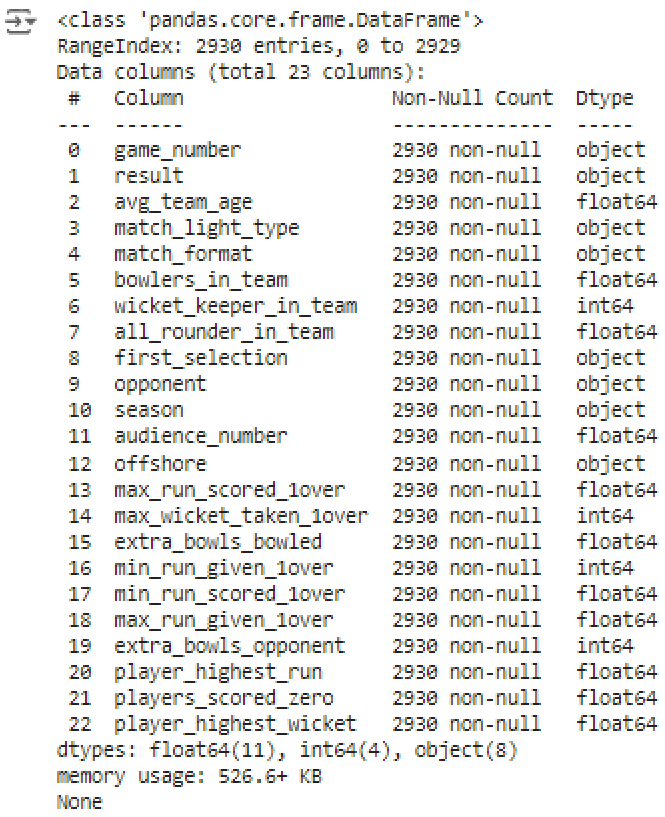
- **Renaming Variables:** Converted all the column names to lower case alphabets.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2930 entries, 0 to 2929
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   game_number            2930 non-null   object 
 1   result                 2930 non-null   object 
 2   avg_team_age           2833 non-null   float64
 3   match_light_type       2878 non-null   object 
 4   match_format           2860 non-null   object 
 5   bowlers_in_team        2848 non-null   float64
 6   wicket_keeper_in_team  2930 non-null   int64  
 7   all_rounder_in_team    2890 non-null   float64
 8   first_selection        2871 non-null   object 
 9   opponent               2894 non-null   object 
 10  season                 2868 non-null   object 
 11  audience_number        2849 non-null   float64
 12  offshore               2866 non-null   object 
 13  max_run_scored_1over   2902 non-null   float64
 14  max_wicket_taken_1over 2930 non-null   int64  
 15  extra_bowls_bowled     2901 non-null   float64
 16  min_run_given_1over    2930 non-null   int64  
 17  min_run_scored_1over   2903 non-null   float64
 18  max_run_given_1over    2896 non-null   float64
 19  extra_bowls_opponent   2930 non-null   int64  
 20  player_highest_run     2902 non-null   float64
 21  players_scored_zero    2930 non-null   object 
 22  player_highest_wicket  2930 non-null   object 
dtypes: float64(9), int64(4), object(10)
memory usage: 526.6+ KB
None
```

Fig 5

- **Missing Values:** Missing values has been treated, the missing values in numerical columns has been filled with mean and the missing values in categorical columns have been filled with mode.



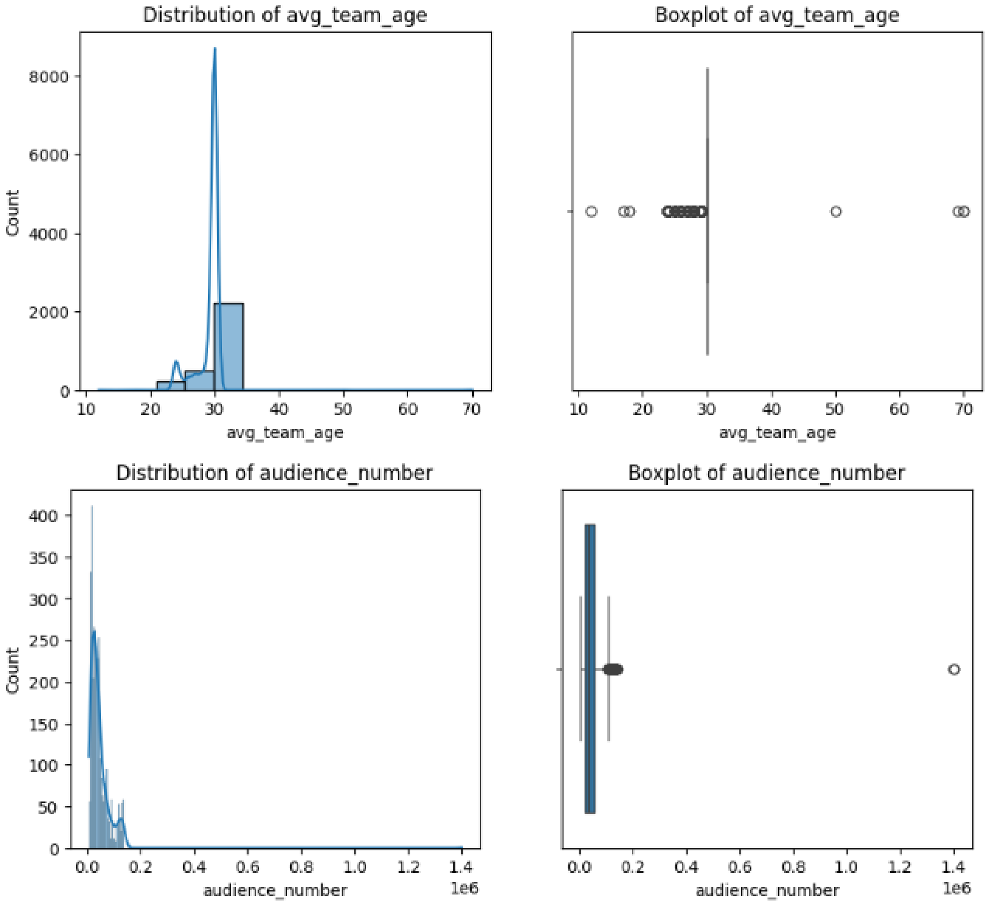
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2930 entries, 0 to 2929
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   game_number            2930 non-null   object 
 1   result                 2930 non-null   object 
 2   avg_team_age           2930 non-null   float64
 3   match_light_type       2930 non-null   object 
 4   match_format           2930 non-null   object 
 5   bowlers_in_team        2930 non-null   float64
 6   wicket_keeper_in_team  2930 non-null   int64  
 7   all_rounder_in_team    2930 non-null   float64
 8   first_selection        2930 non-null   object 
 9   opponent               2930 non-null   object 
 10  season                 2930 non-null   object 
 11  audience_number        2930 non-null   float64
 12  offshore               2930 non-null   object 
 13  max_run_scored_1over   2930 non-null   float64
 14  max_wicket_taken_1over 2930 non-null   int64  
 15  extra_bowls_bowled     2930 non-null   float64
 16  min_run_given_1over    2930 non-null   int64  
 17  min_run_scored_1over   2930 non-null   float64
 18  max_run_given_1over    2930 non-null   float64
 19  extra_bowls_opponent   2930 non-null   int64  
 20  player_highest_run     2930 non-null   float64
 21  players_scored_zero    2930 non-null   float64
 22  player_highest_wicket  2930 non-null   float64
dtypes: float64(11), int64(4), object(8)
memory usage: 526.6+ KB
None
```

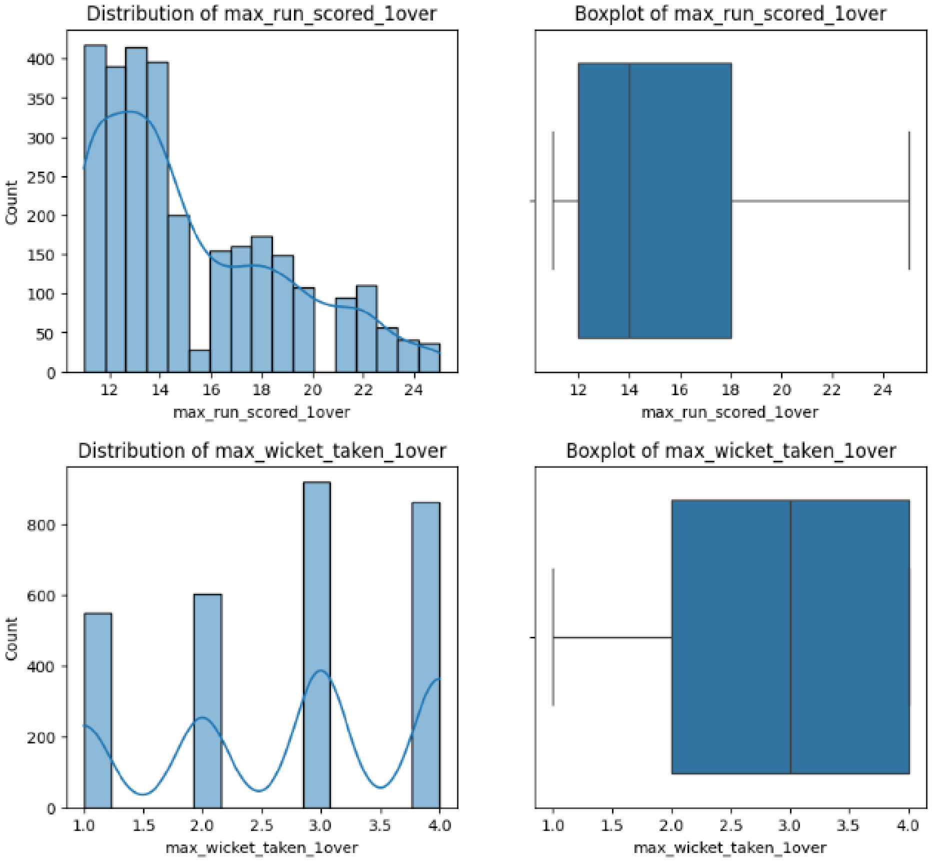
Fig 6

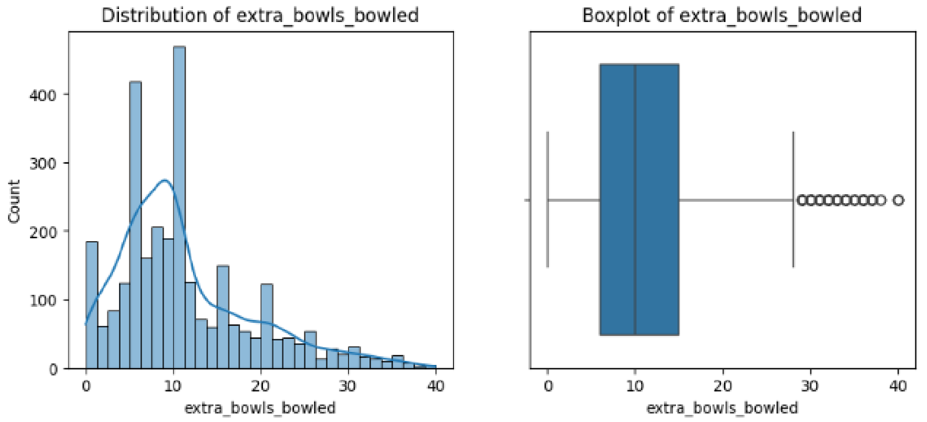
3. **Exploratory Data Analysis**
   a) **Univariate analysis (distribution and spread for every continuous attribute, distribution of data in categories for categorical ones) b) Bivariate analysis (relationship between different variables , correlations) c) Removal of unwanted variables (if applicable) d) Missing Value treatment (if applicable) e) Outlier treatment (if required) f) Variable transformation (if applicable) g) Addition of new**

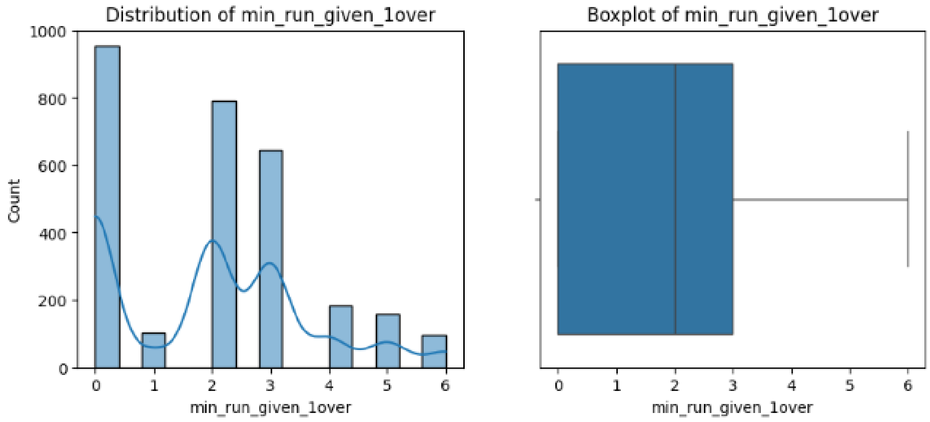
a) Univariate analysis (distribution and spread for every continuous attribute, distribution of data in categories for categorical ones)

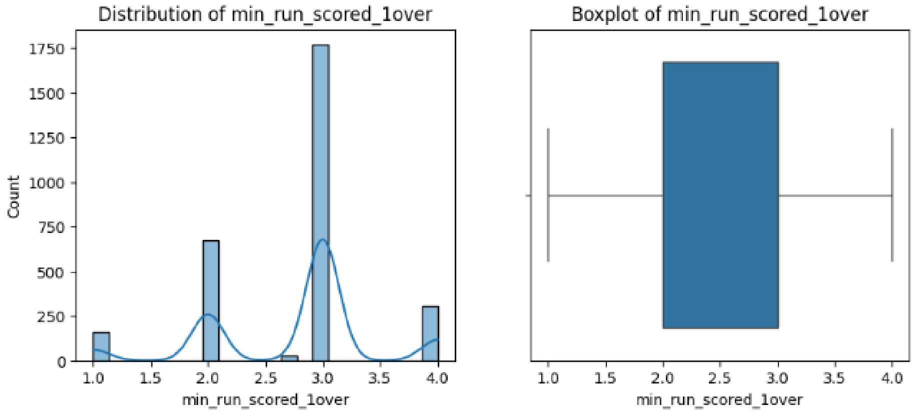
- **Continuous Attributes**

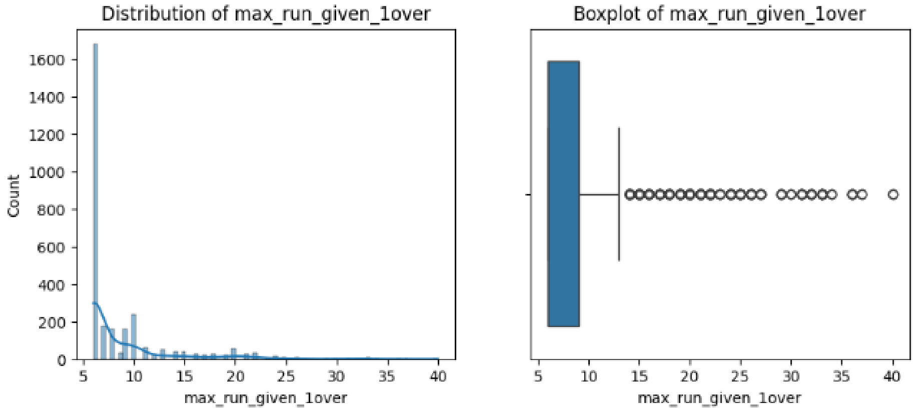

Distribution of avg_team_age
Count
avg_team_age
Boxplot of avg_team_age
avg_team_age
Distribution of audience_number
Count
audience_number
1e6
Boxplot of audience_number
audience_number
1e6

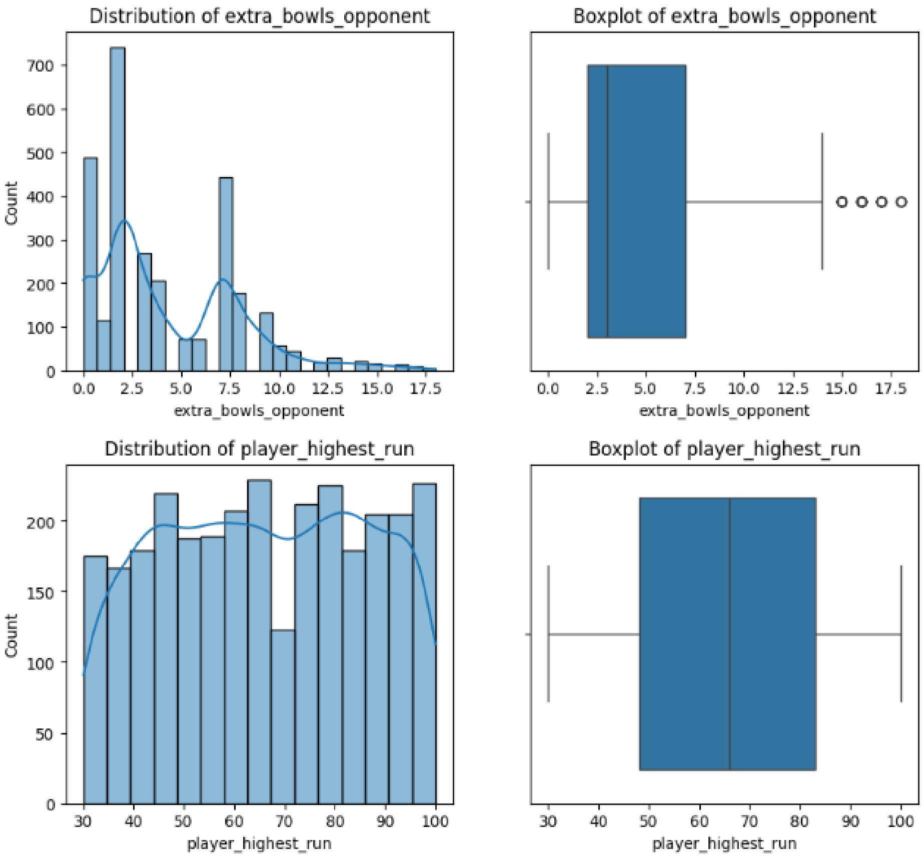

Distribution of max_run_scored_1over
Count
max_run_scored_1over
Boxplot of max_run_scored_1over
max_run_scored_1over
Distribution of max_wicket_taken_1over
Count
max_wicket_taken_1over
Boxplot of max_wicket_taken_1over
max_wicket_taken_1over

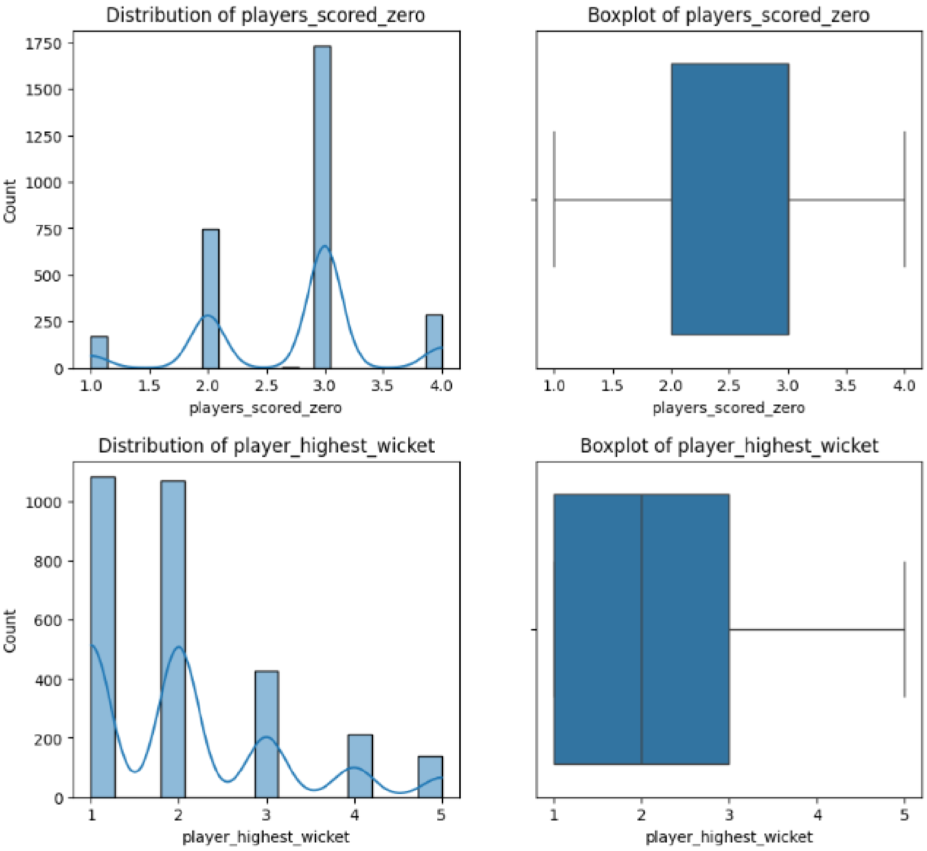

Distribution of extra_bowls_bowled
Count
400
300
200
100
0
0
10
20
30
40
extra_bowls_bowled
Boxplot of extra_bowls_bowled
0
10
20
30
40
extra_bowls_bowled


Distribution of min_run_given_1over
1000
800
600
400
200
0
Count
0
1
2
3
4
5
6
min_run_given_1over
Boxplot of min_run_given_1over
0
1
2
3
4
5
6
min_run_given_1over


Distribution of min_run_scored_1over
Count
1750
1500
1250
1000
750
500
250
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
min_run_scored_1over
Boxplot of min_run_scored_1over
1.0
1.5
2.0
2.5
3.0
3.5
4.0
min_run_scored_1over


Distribution of max_run_given_1over
Count
1600
1400
1200
1000
800
600
400
200
0
5
10
15
20
25
30
35
40
max_run_given_1over
Boxplot of max_run_given_1over
5
10
15
20
25
30
35
40
max_run_given_1over


Distribution of extra_bowls_opponent
Count
0
100
200
300
400
500
600
700
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
extra_bowls_opponent
Boxplot of extra_bowls_opponent
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
extra_bowls_opponent
Distribution of player_highest_run
Count
0
50
100
150
200
30
40
50
60
70
80
90
100
player_highest_run
Boxplot of player_highest_run
30
40
50
60
70
80
90
100
player_highest_run


Distribution of players_scored_zero
Count
0
250
500
750
1000
1250
1500
1750
1.0
1.5
2.0
2.5
3.0
3.5
4.0
players_scored_zero
Boxplot of players_scored_zero
1.0
1.5
2.0
2.5
3.0
3.5
4.0
players_scored_zero
Distribution of player_highest_wicket
Count
0
200
400
600
800
1000
1
2
3
4
5
player_highest_wicket
Boxplot of player_highest_wicket
1
2
3
4
5
player_highest_wicket

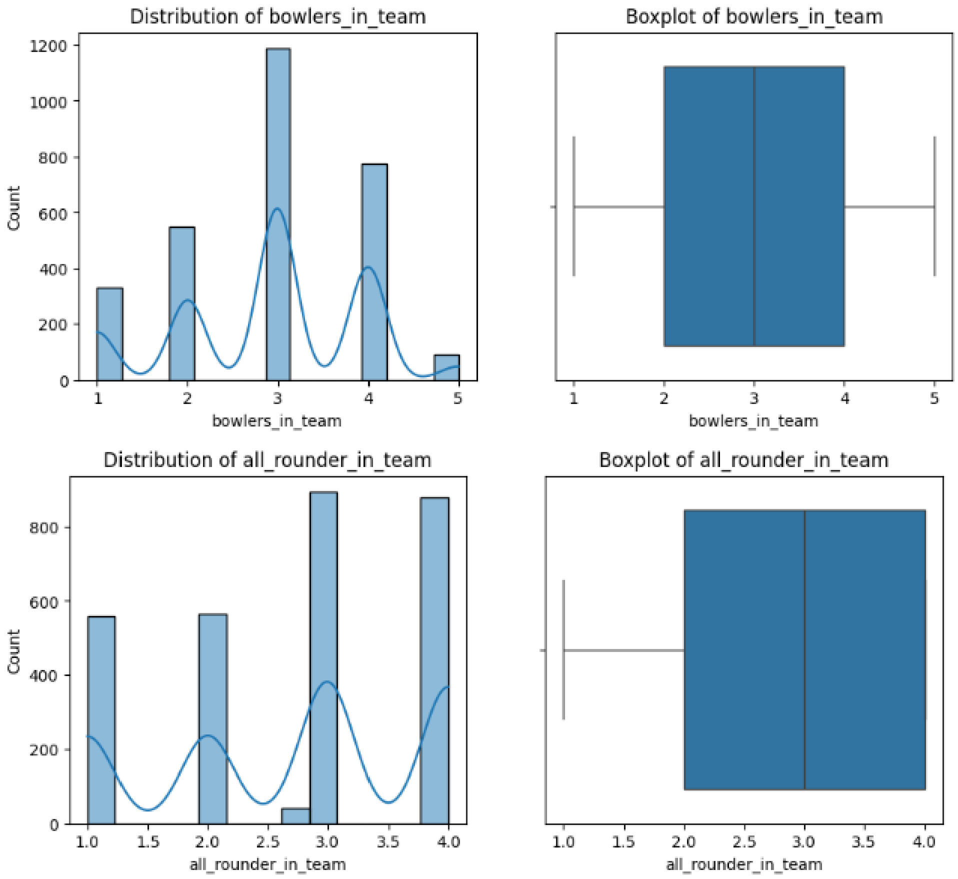
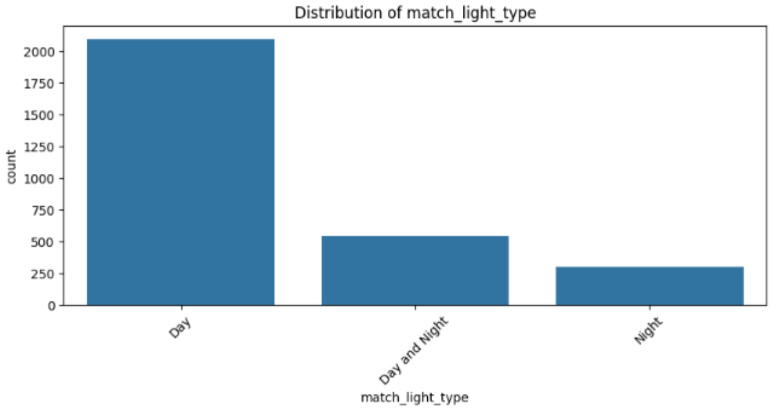
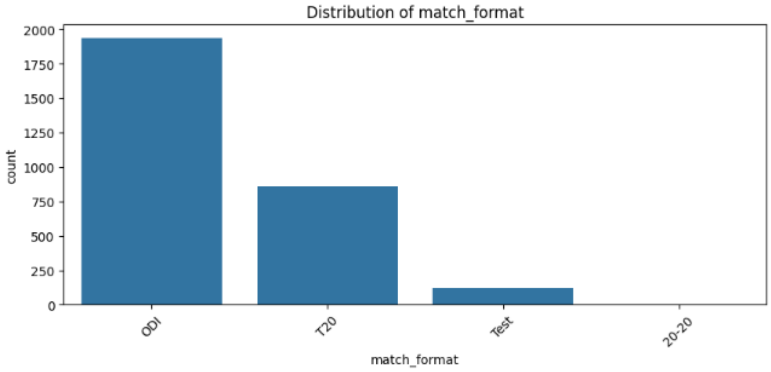
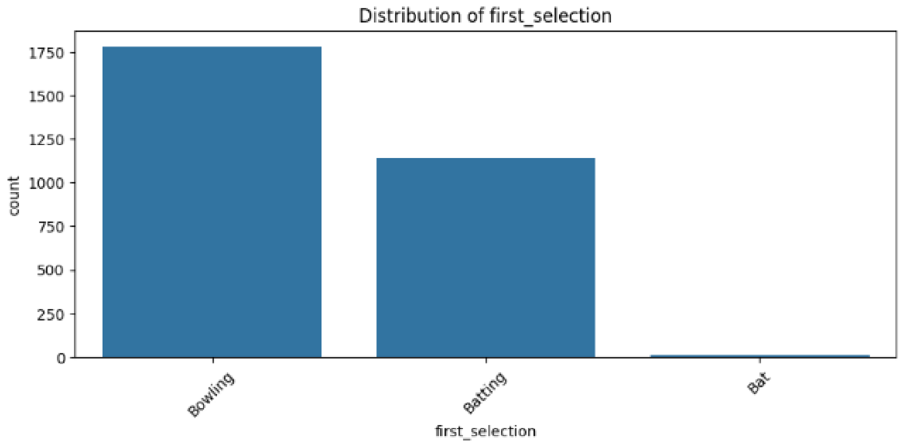

Distribution of bowlers_in_team
Count
1200
1000
800
600
400
200
0
1
2
3
4
5
bowlers_in_team
Boxplot of bowlers_in_team
1
2
3
4
5
bowlers_in_team
Distribution of all_rounder_in_team
Count
800
600
400
200
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
all_rounder_in_team
Boxplot of all_rounder_in_team
1.0
1.5
2.0
2.5
3.0
3.5
4.0
all_rounder_in_team

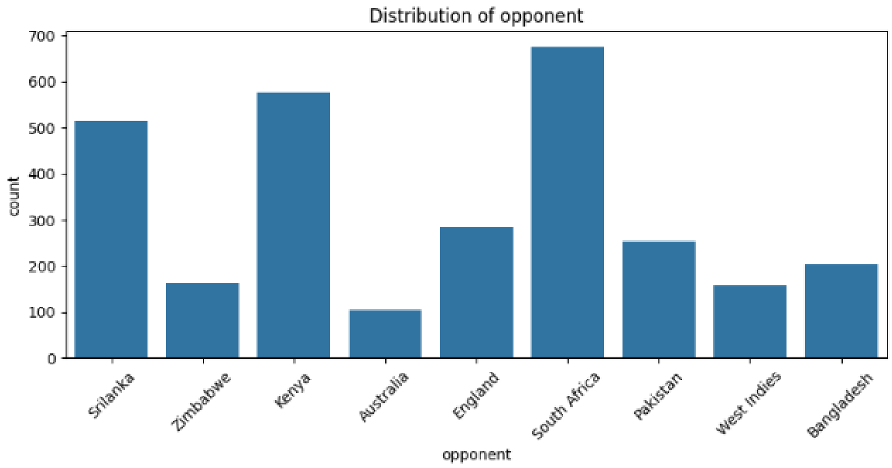
Fig 7

- **Categorical Attributes**


Distribution of match_light_type


Distribution of match_format


Distribution of first_selection


Distribution of opponent

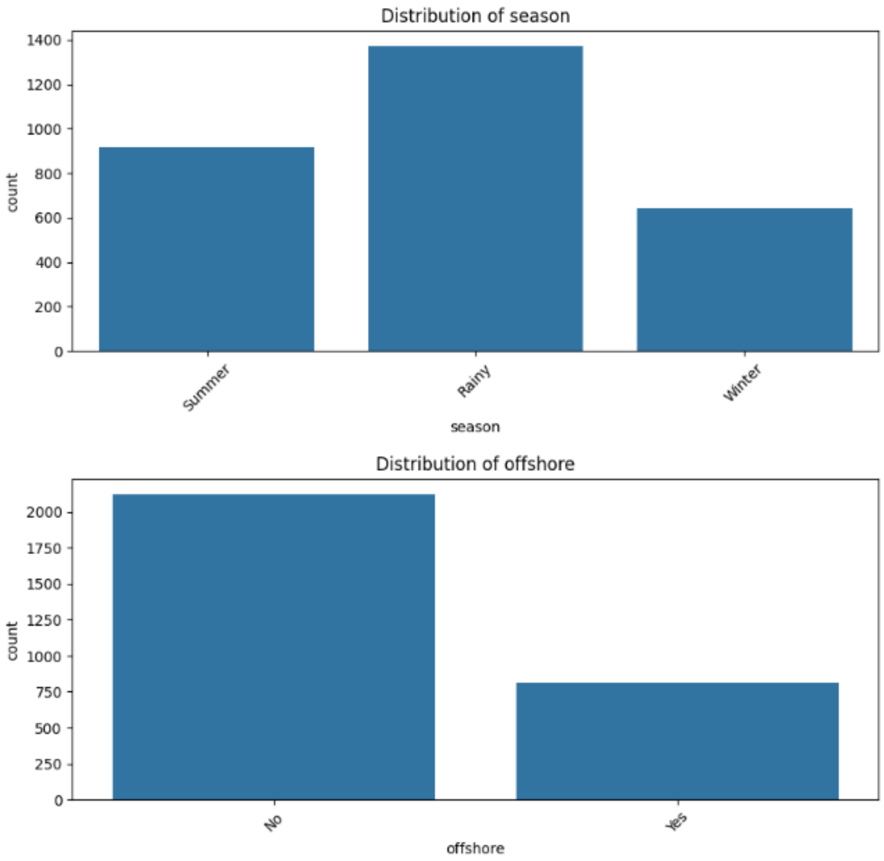

Distribution of season
count
1400
1200
1000
800
600
400
200
0
Summer
Rainy
Winter
season
Distribution of offshore
count
2000
1750
1500
1250
1000
750
500
250
0
No
Yes
offshore

Fig 8

b) Bivariate analysis (relationship between different variables, correlations)

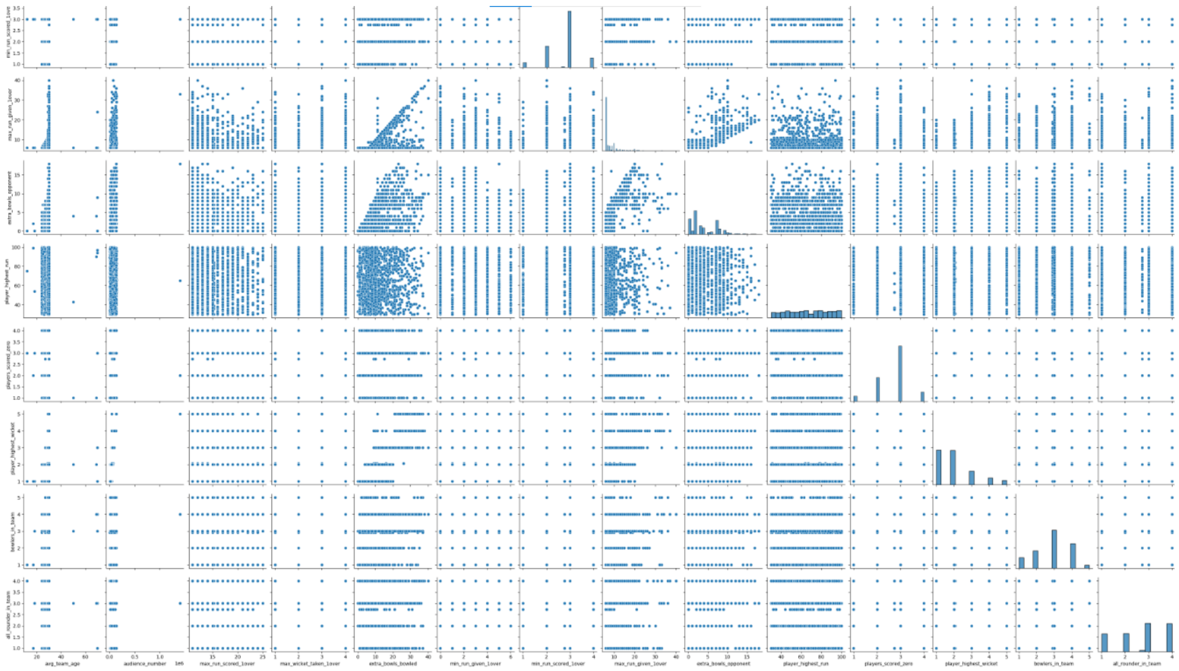
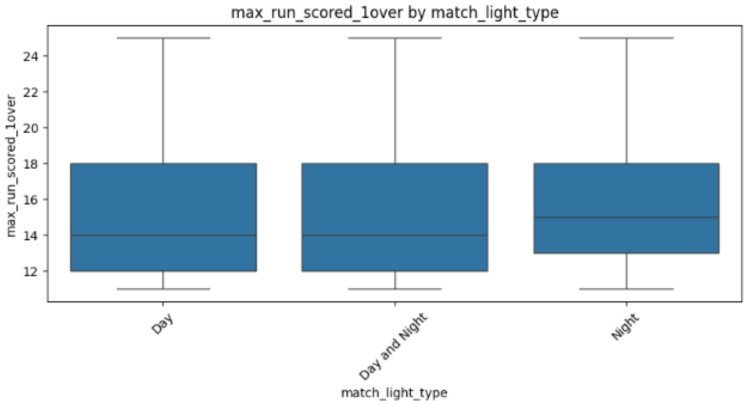
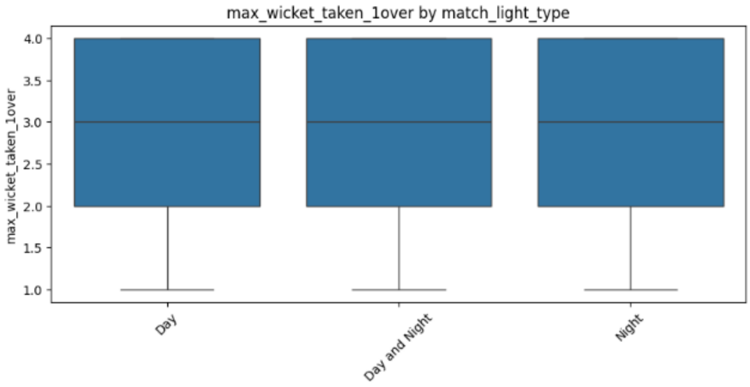
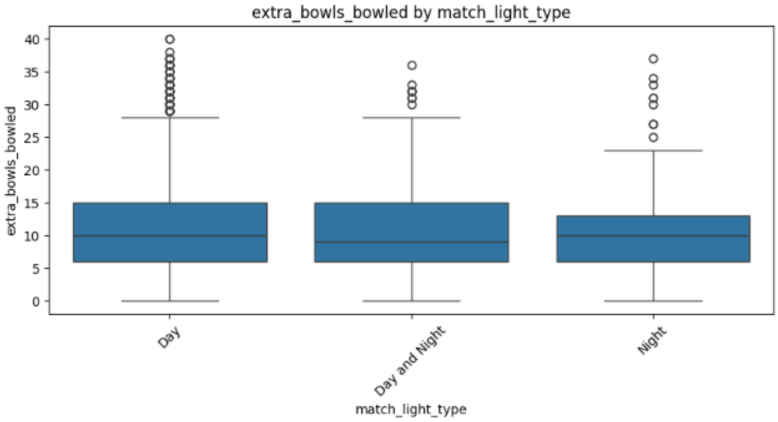
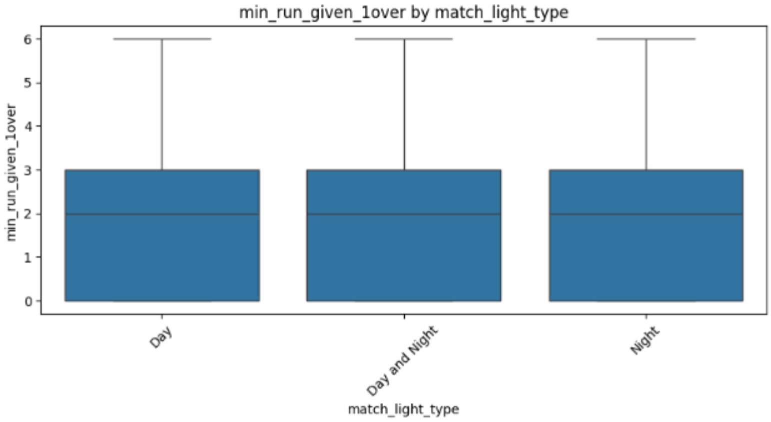


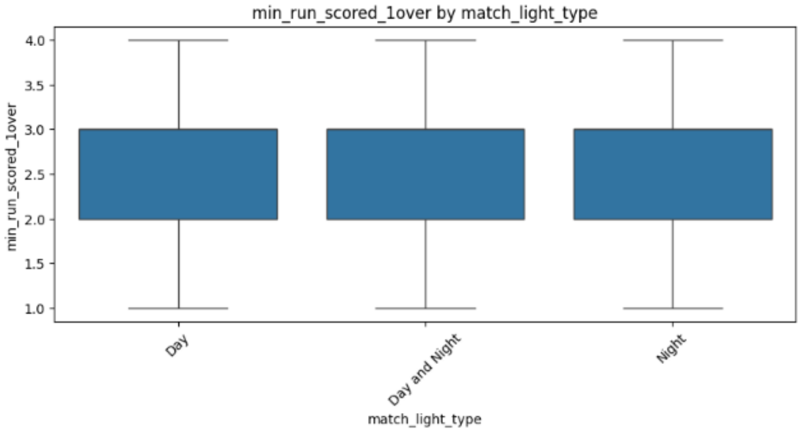
Fig 9

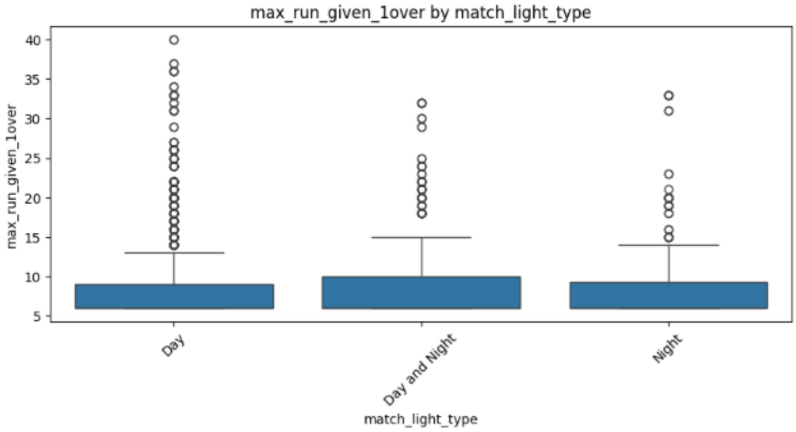


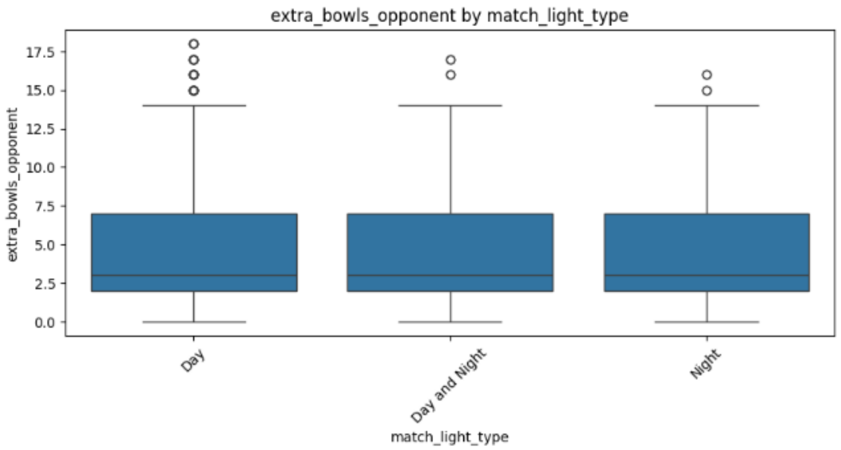


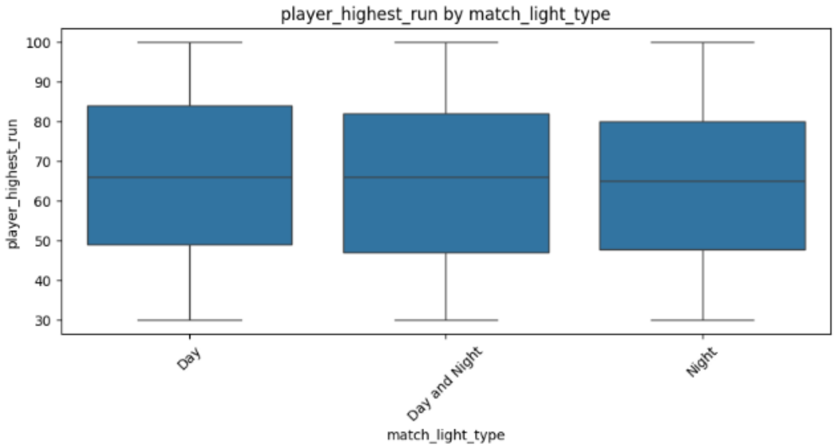

extra_bowls_bowled by match_light_type
extra_bowls_bowled
0
5
10
15
20
25
30
35
40
Day
Day and Night
Night
match_light_type

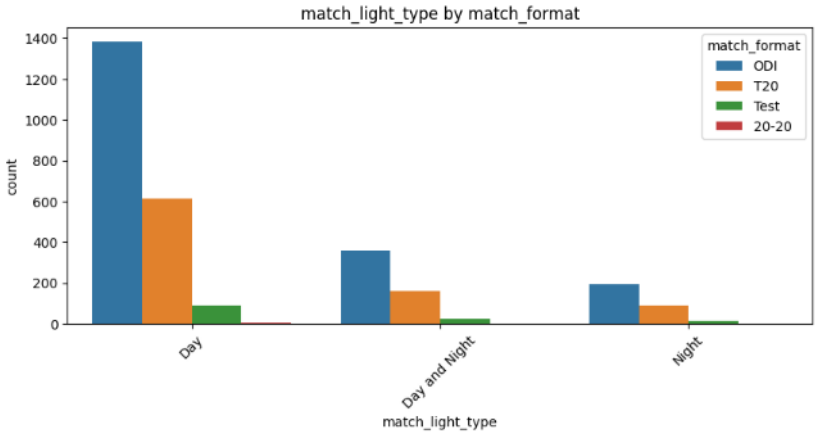

min_run_given_1over by match_light_type
min_run_given_1over
0
1
2
3
4
5
6
Day
Day and Night
Night
match_light_type

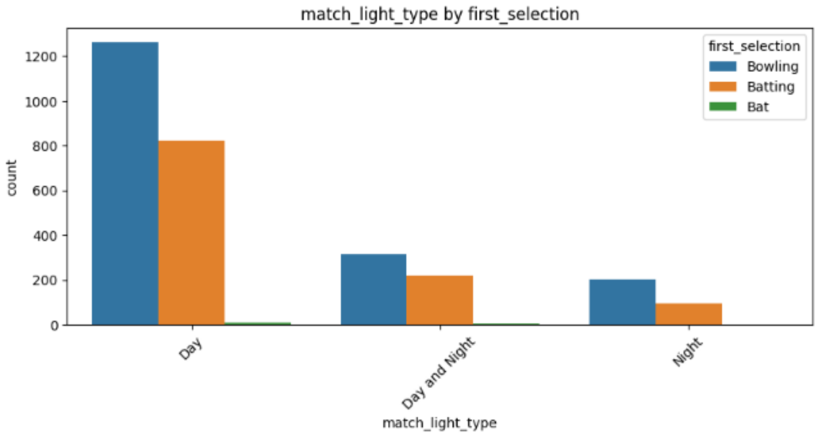

min_run_scored_1over by match_light_type
min_run_scored_1over
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Day
Day and Night
Night
match_light_type

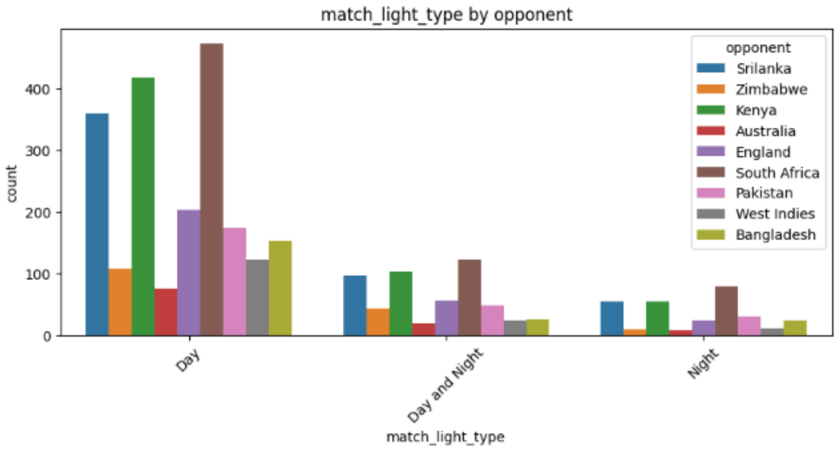

max_run_given_1over by match_light_type
max_run_given_1over
5
10
15
20
25
30
35
40
Day
Day and Night
Night
match_light_type

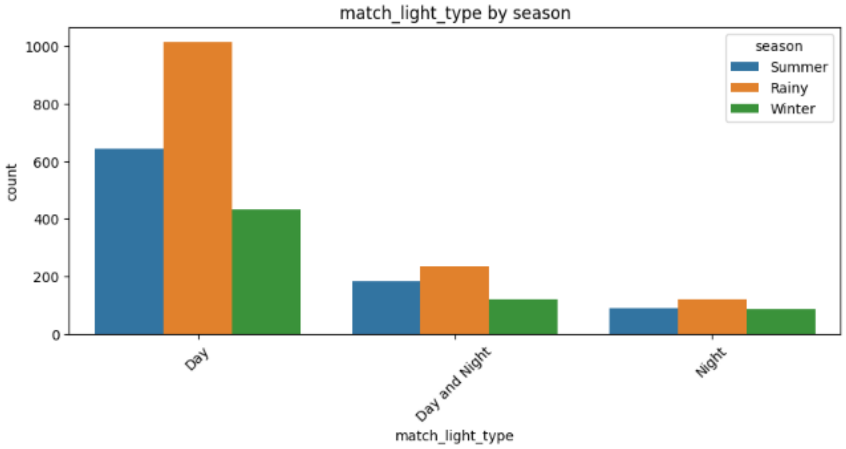

extra_bowls_opponent by match_light_type

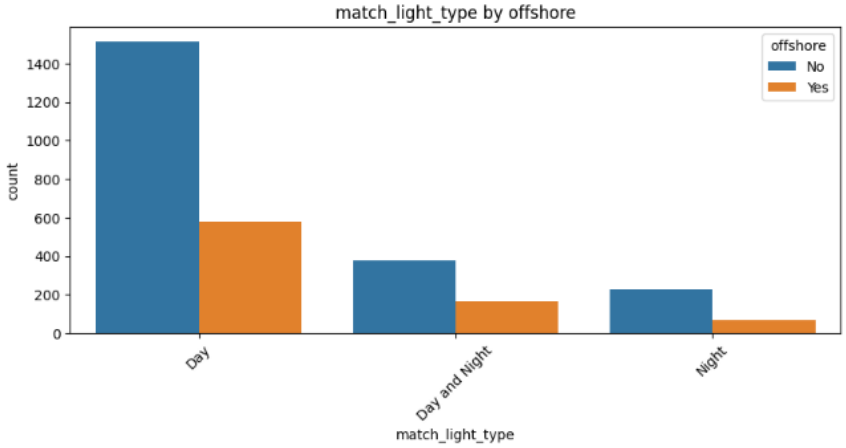

player_highest_run by match_light_type

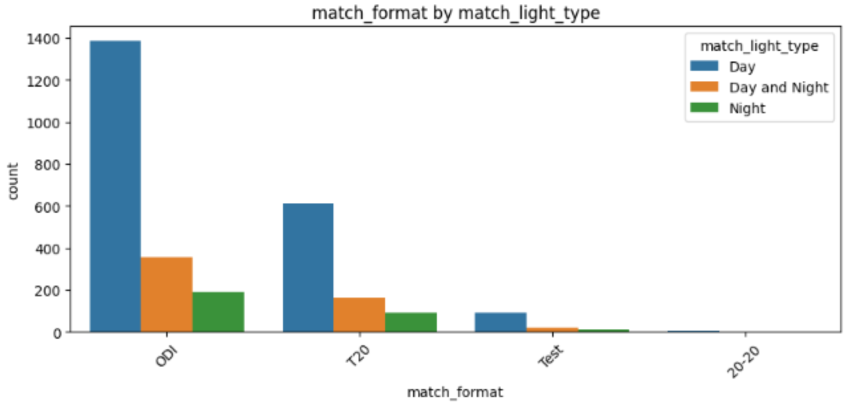

match_light_type by match_format


match_light_type by first_selection


match_light_type by opponent
count
0
100
200
300
400
Day
Day and Night
Night
match_light_type
opponent
Srilanka
Zimbabwe
Kenya
Australia
England
South Africa
Pakistan
West Indies
Bangladesh


match_light_type by season
count
0
200
400
600
800
1000
Day
Day and Night
Night
match_light_type
season
Summer
Rainy
Winter


match_light_type by offshore
count
0
200
400
600
800
1000
1200
1400
Day
Day and Night
Night
match_light_type
offshore
No
Yes


match_format by match_light_type
count
0
200
400
600
800
1000
1200
1400
ODI
T20
Test
20-20
match_format
match_light_type
Day
Day and Night
Night

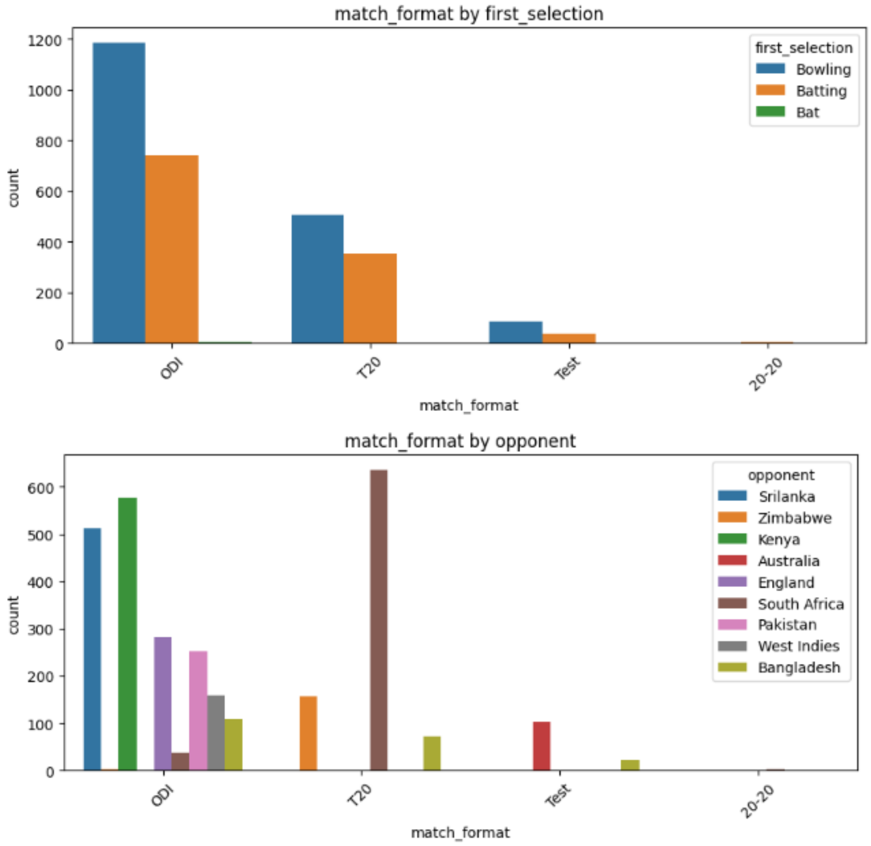


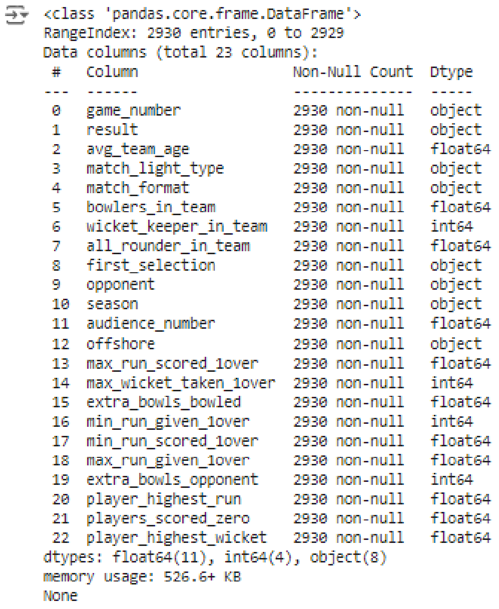
Fig 10

c) Removal of unwanted variables (if applicable)

Not Applicable

d) Missing Value treatment (if applicable)

Missing values treated.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2930 entries, 0 to 2929
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   game_number            2930 non-null   object
 1   result                 2930 non-null   object
 2   avg_team_age           2930 non-null   float64
 3   match_light_type       2930 non-null   object
 4   match_format           2930 non-null   object
 5   bowlers_in_team        2930 non-null   float64
 6   wicket_keeper_in_team  2930 non-null   int64
 7   all_rounder_in_team    2930 non-null   float64
 8   first_selection        2930 non-null   object
 9   opponent               2930 non-null   object
 10  season                 2930 non-null   object
 11  audience_number        2930 non-null   float64
 12  offshore               2930 non-null   object
 13  max_run_scored_1over   2930 non-null   float64
 14  max_wicket_taken_1over 2930 non-null   int64
 15  extra_bowls_bowled     2930 non-null   float64
 16  min_run_given_1over    2930 non-null   int64
 17  min_run_scored_1over   2930 non-null   float64
 18  max_run_given_1over    2930 non-null   float64
 19  extra_bowls_opponent   2930 non-null   int64
 20  player_highest_run     2930 non-null   float64
 21  players_scored_zero    2930 non-null   float64
 22  player_highest_wicket  2930 non-null   float64
dtypes: float64(11), int64(4), object(8)
memory usage: 526.6+ KB
None
```

Fig 11

e) Outlier treatment (if required)

Outlier treated for Continuous attributes.

f) Variable transformation (if applicable)

Transformed variables to meet assumptions of modelling techniques and to improve interpretability.

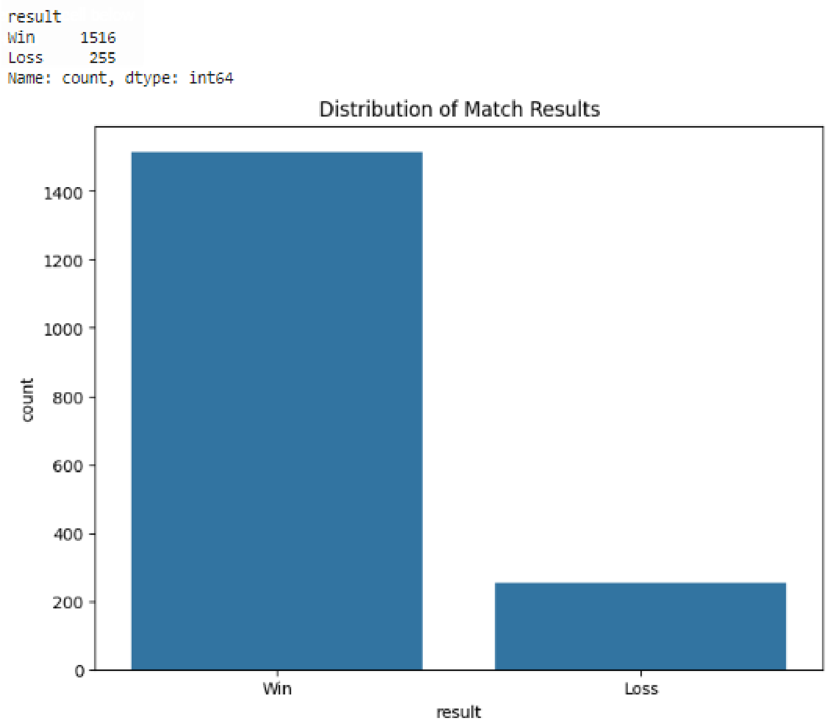
g) Addition of new

Not Required.

4. **Business insights from EDA**
   a) Is the data unbalanced? If so, what can be done? Please explain in the context of the business b) Any business insights using clustering (if applicable) c) Any other business insights

a) Is the data unbalanced? If so, what can be done? Please explain in the context of the business

Yes.



```
result
Win     1516
Loss     255
Name: count, dtype: int64
```

Fig 12

Resampling:

- **Oversampling:** Increase the number of instances in the minority class.
- **Undersampling:** Decrease the number of instances in the majority class.
- **SMOTE (Synthetic Minority Over-sampling Technique):** Generate synthetic examples for the minority class.

Algorithmic Approaches:

- **Adjusting Class Weights:** Modify the algorithm to give more importance to the minority class.
- **Anomaly Detection:** Treat the minority class as anomalies and use anomaly detection techniques.

Ensemble Methods:

- Use ensemble techniques like bagging and boosting that can handle unbalanced datasets better.

b) Any business insights using clustering (if applicable)

Performance Clusters: Identifing clusters where the team performs exceptionally well or poorly and analysing the conditions (e.g., match format, opponent, season) contributing to these performances.

Strategy Development: Tailor strategies for matches based on the identified clusters. For instance, if a cluster shows poor performance in T20s against a particular opponent, specific strategies can be developed for those scenarios.

c) Any other business insights

Player Performance Analysis:

Age and Experience: Analysed the impact of team age and experience on match outcomes. As the team age went high the winning percentage went more and more. Age seems to bring experience to the team for better and accurate performance of the team.

Match Conditions:

Home vs. Away Performance: Win rate is high on home turf whereas the losing rate is same on home turf and away turf.

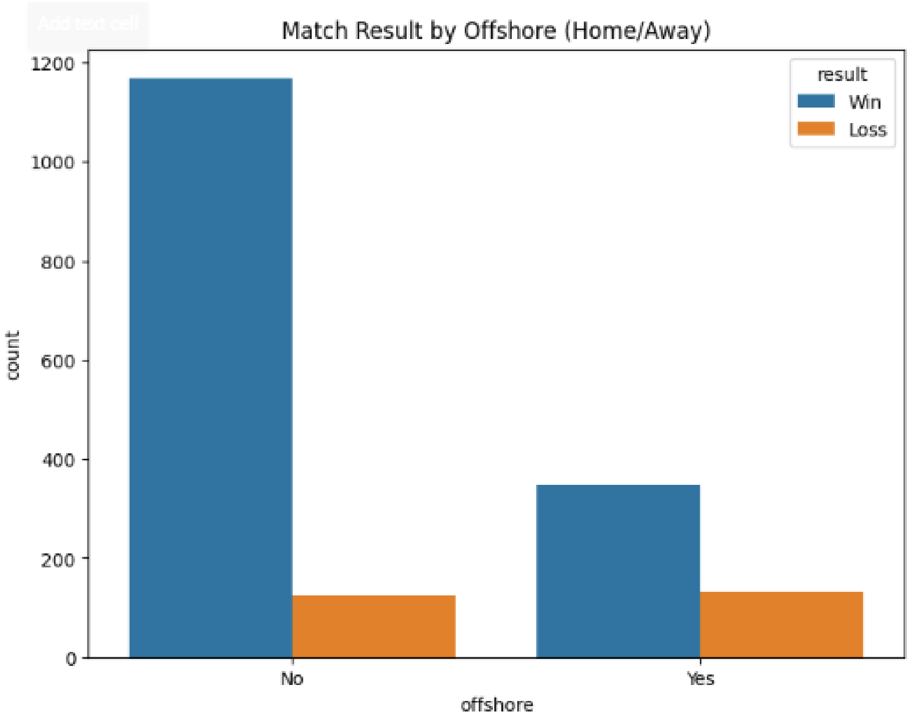


Fig 13

Seasonal Impact: Performance of batsmen are better in Day and Day and Night conditions as compared to Night matches.

Audience Influence:

Audience Size: Analysed that larger audience numbers correlate with better performance, indicating a possible boost from crowd support.

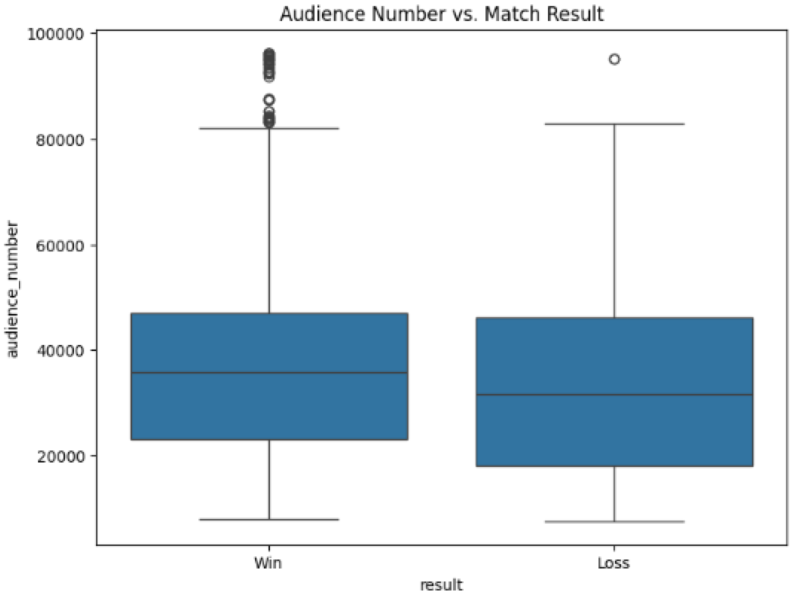


Fig 14