

# CAPSTONE PROJECT

## Cricket Win Prediction

By Vishesh Sharma

# Contents

1. Introduction
2. Data Cleaning and Pre- Processing
3. EDA and Business Implication
  - 3.1 Univariate Analysis
  - 3.2 Bivariate Analysis
  - 3.3 Impact on our Business Problem
4. Model Building and Validation
  - 4.1 Model performance, Model Selection
  - 4.2 Mode Fine tuning
  - 4.3 Feature Importance and Prediction
5. Final Interpretation and recommendations

# 1.Introduction

Our primary goal is to develop accurate Machine Learning models capable of predicting a victory for the Indian Cricket Team using historical match data. Upon achieving this, we aim to extract actionable insights and recommendations from the model's predictions to enhance India's chances of winning. Furthermore, we have upcoming matches for which we need to predict the outcomes. In the event that our model predicts a loss, we must provide unique and feasible strategies tailored to the variables within the dataset. These strategies should vary across the matches to prevent the opposition from anticipating our approach and devising counter-strategies effectively.

Upcoming 5 matches are: -

1. 1 Test match with England in England. All the match are day matches. In England, it will be rainy season at the time to match.
2. 2 T20 match with Australia in India. All the match are Day and Night matches. In India, it will be winter season at the time to match.
3. 2 ODI match with Sri Lanka in India. All the match are Day and Night matches. In India, it will be winter season at the time to match.

## 2. Data Cleaning and Pre-Processing

### Values ambiguity

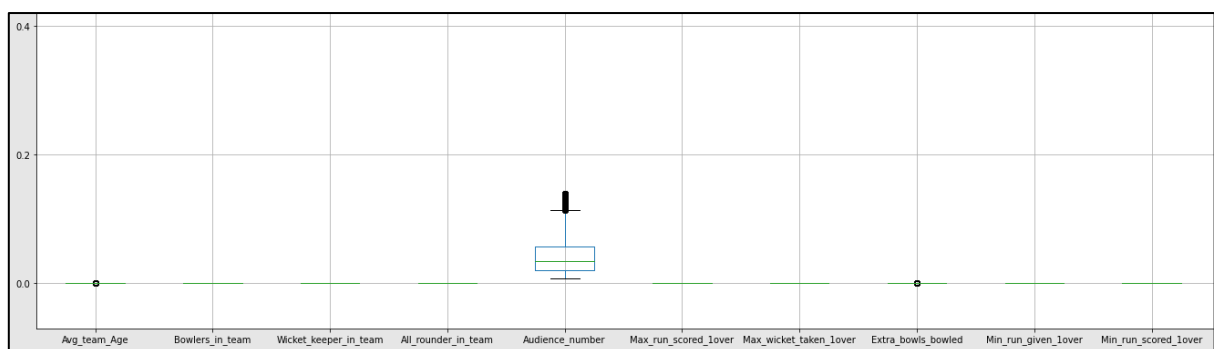
Dataset contains 2930 rows and 23 columns. Initially, data contained 617 rows with missing values. First we started with checking the values in categorical column and seeing how many unique entries are present and any of them is wrongly written or repeated in other form. For example 'Match format' columns contains entries like,

1. T20
2. Test
3. ODI

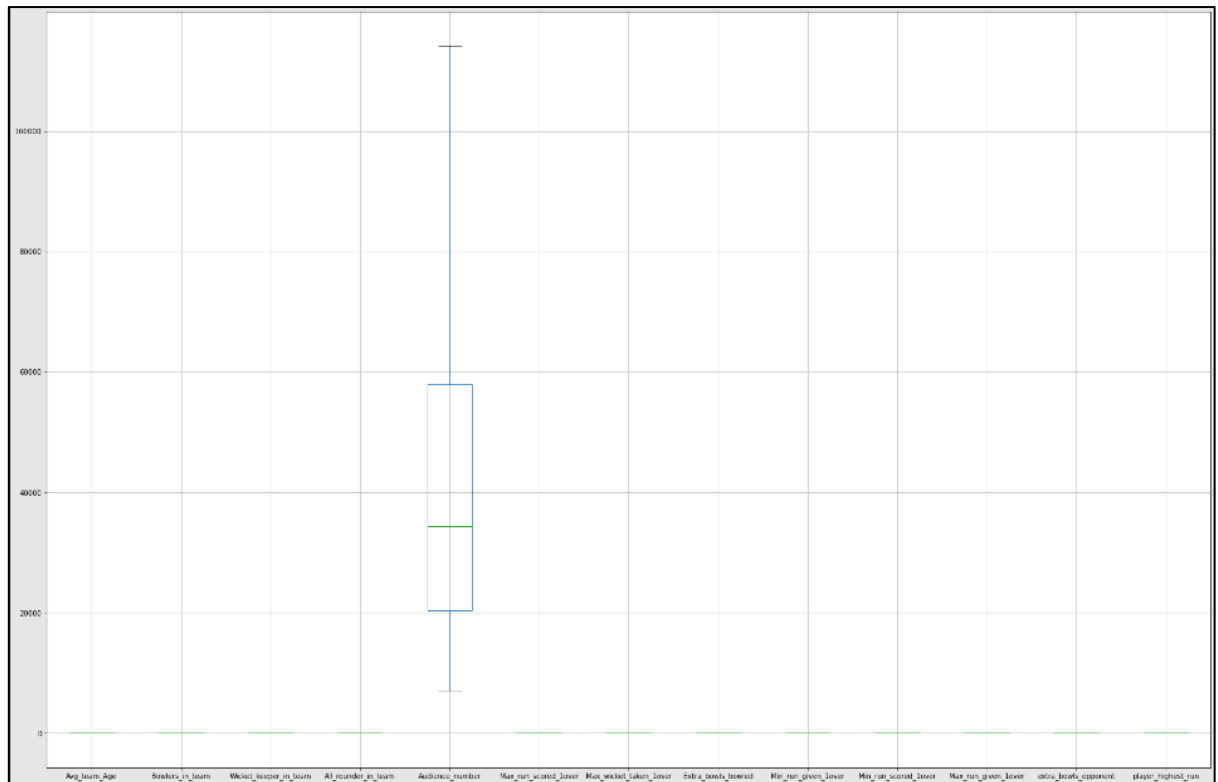
But we found that it has also one entry '20-20' which is same for T20 so first replaced '20-20' with 'T20'.

Similarly, we checked for categorical columns and replaced the ambiguous values with the values relevant with data.

### Outlier Treatment



From the boxplot we can see that there were various outliers in continuous variables so we replaced these outliers using IQR method ie. to replace extreme values with the whiskers value of boxplot. After treatment below is the outlier free data.



## Datatype conversion, Columns dropped, Duplicate Values

There are some features in the data which earlier present as integer data. But as those columns didn't contain continuous data we converted to categorical data.

1. Batters\_in\_team
2. All\_rounder\_in\_team
3. Max\_wicket\_taken\_in\_1\_over

We also dropped 'Wicket\_Keeper\_in\_Team' column as it contained only one value throughout data ie. one. We also dropped 'Game\_Number' which was just sort of index column.

We checked the data for duplicate values. **The data didn't contain duplicate entries.**

## Missing Values Treatment

After this we checked for rows with missing values. If rows have missing values less than 3% then we tried to drop those rows.

But our result showed that there were not enough rows with 1,2,3 missing values to drop less than 3%.

So for imputing missing values we used different approach for categorical and continuous columns.

For Continuous columns we used **KNN Imputer with Grid Search with Cross Validation** in order to select best parameter and for KNN it would be best nearest neighbour.

And for Categorical columns we used **mode** to replace value.

After the treatment data looks like below: -

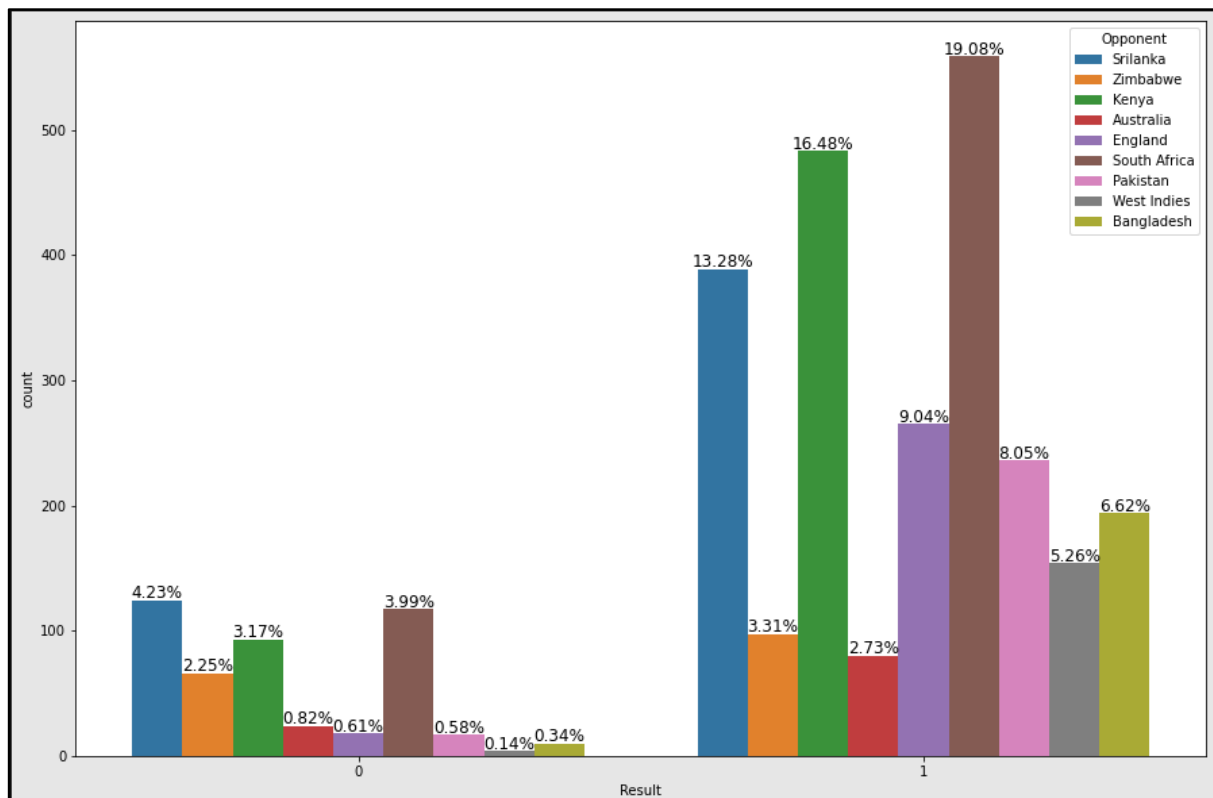
```
RangeIndex: 2930 entries, 0 to 2929
Data columns (total 21 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Avg_team_Age                          2930 non-null   float64
1   Audience_number                       2930 non-null   float64
2   Max_run_scored_lover                  2930 non-null   float64
3   Extra_bowls_bowled                    2930 non-null   float64
4   Min_run_given_lover                   2930 non-null   float64
5   Min_run_scored_lover                  2930 non-null   float64
6   Max_run_given_lover                   2930 non-null   float64
7   extra_bowls_opponent                  2930 non-null   float64
8   player_highest_run                    2930 non-null   float64
9   Result                                2930 non-null   object
10  Match_light_type                       2930 non-null   object
11  Match_format                           2930 non-null   object
12  Bowlers_in_team                        2930 non-null   object
13  All_rounder_in_team                   2930 non-null   object
14  First_selection                        2930 non-null   object
15  Opponent                               2930 non-null   object
16  Season                                 2930 non-null   object
17  Offshore                               2930 non-null   object
18  Max_wicket_taken_lover                 2930 non-null   object
19  Players_scored_zero                    2930 non-null   int64
20  player_highest_wicket                  2930 non-null   int64
dtypes: float64(9), int64(2), object(10)
```

Finally we replace 'Loss' with 0 and 'Win' with 1 in Result column

### 3. EDA and Business Implication

#### Univariate Analysis

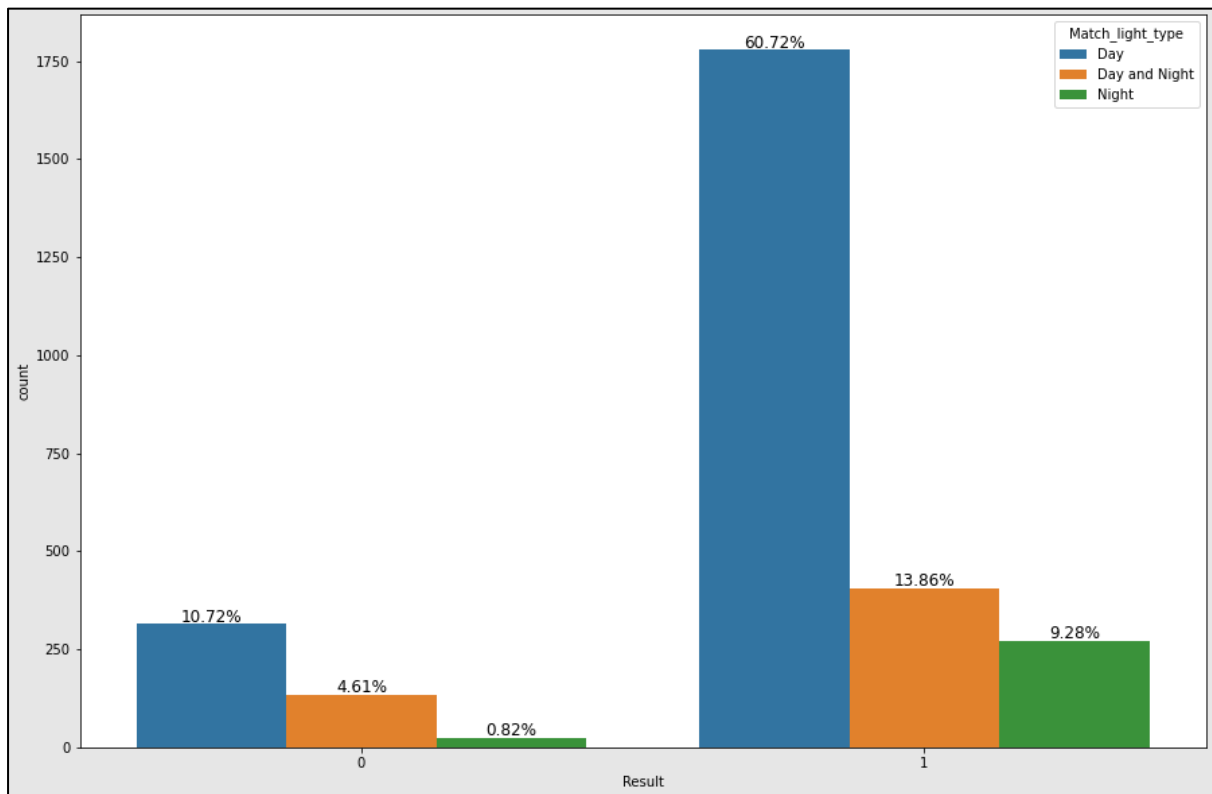
Win/Loss distribution against different opponents



This distribution shows that the data contains major proportion of wins over loose. But if we just consider the data

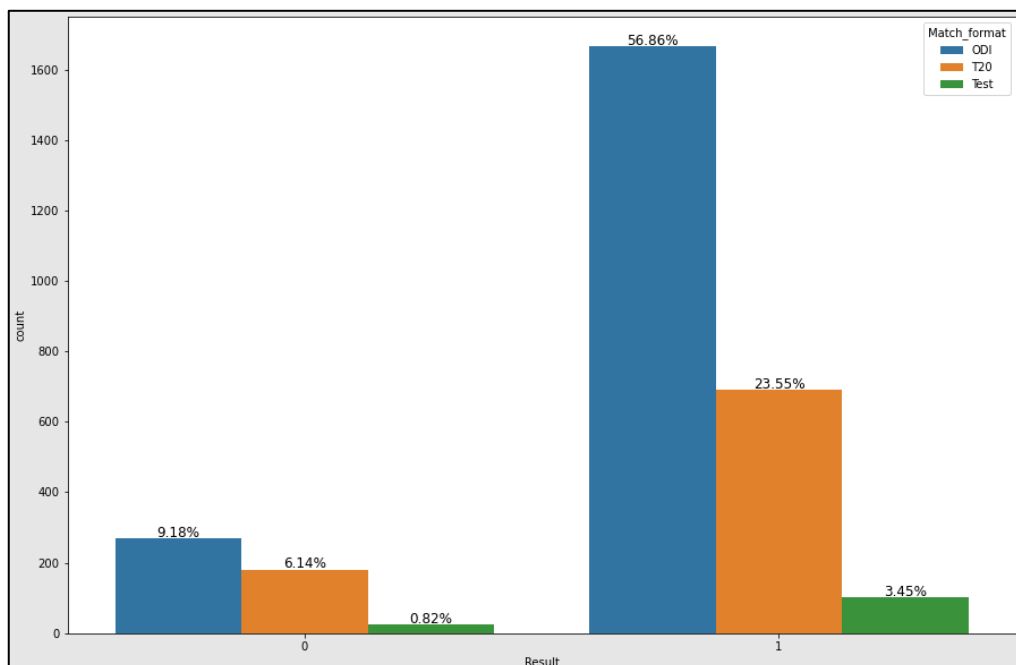
- Top 3 teams against India has won is South Africa- 19% of times, Kenya- 16.5% of times and Sri Lanka- 13.3% of times.

## Win/Loss Distribution against on basis of Match Light Type



- India has won maximum matches in Day time- 60% of time, Day and Night matches- 14% and Night matches 9.3% of times.

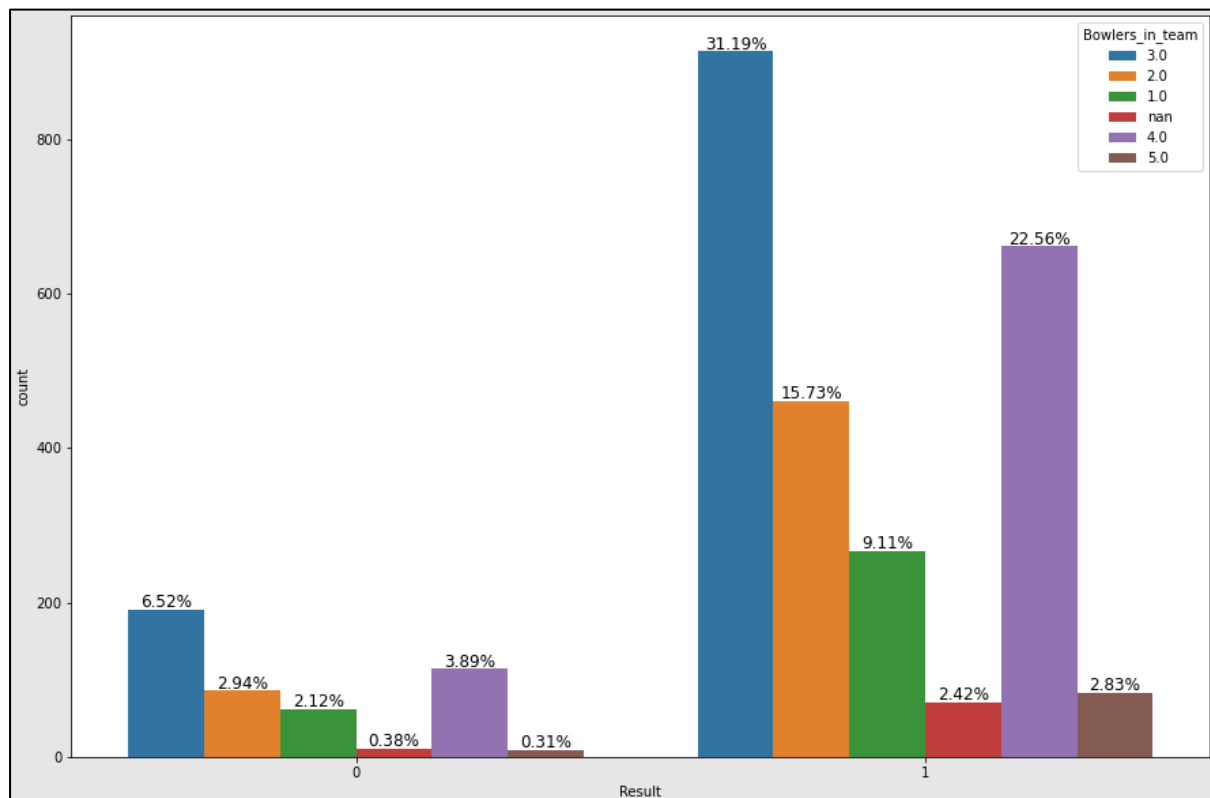
## Win/Loss distribution against Match Type





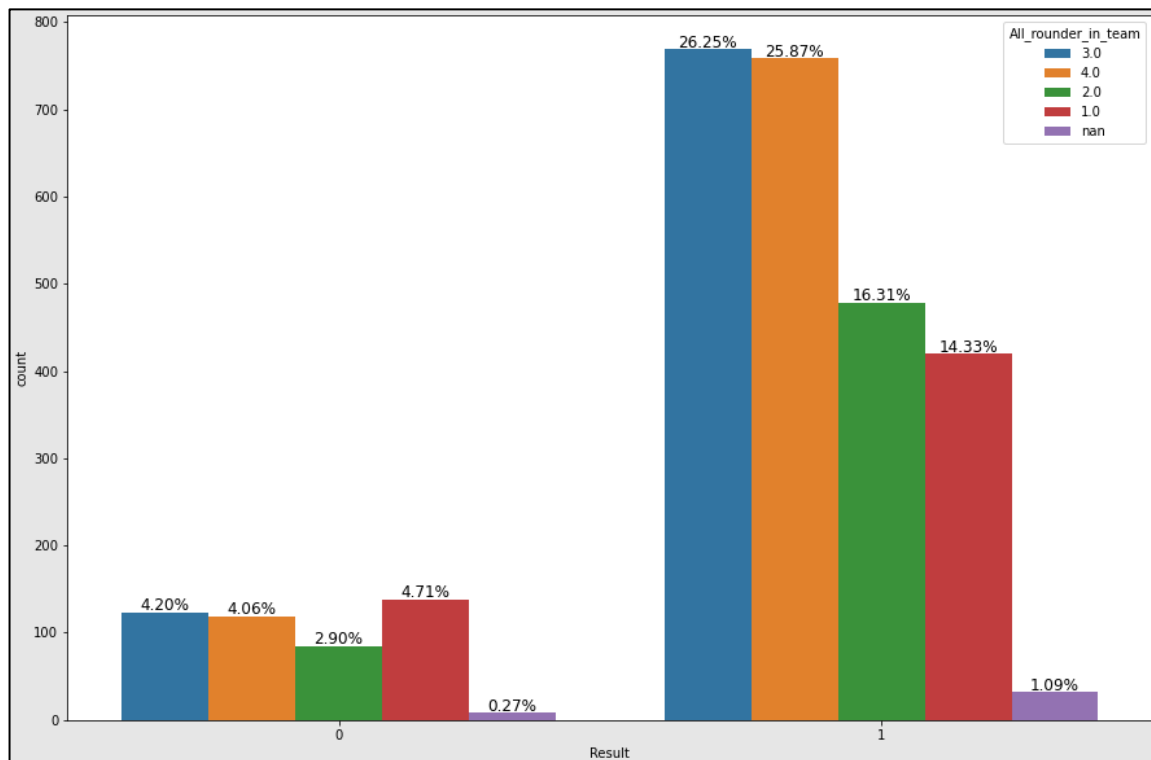
- India has maximum proportion of wins in ODI that is 57%, then T20 13.5% and Test ie. 3.5%.

### Win/Loss distribution against Bowlers in Team



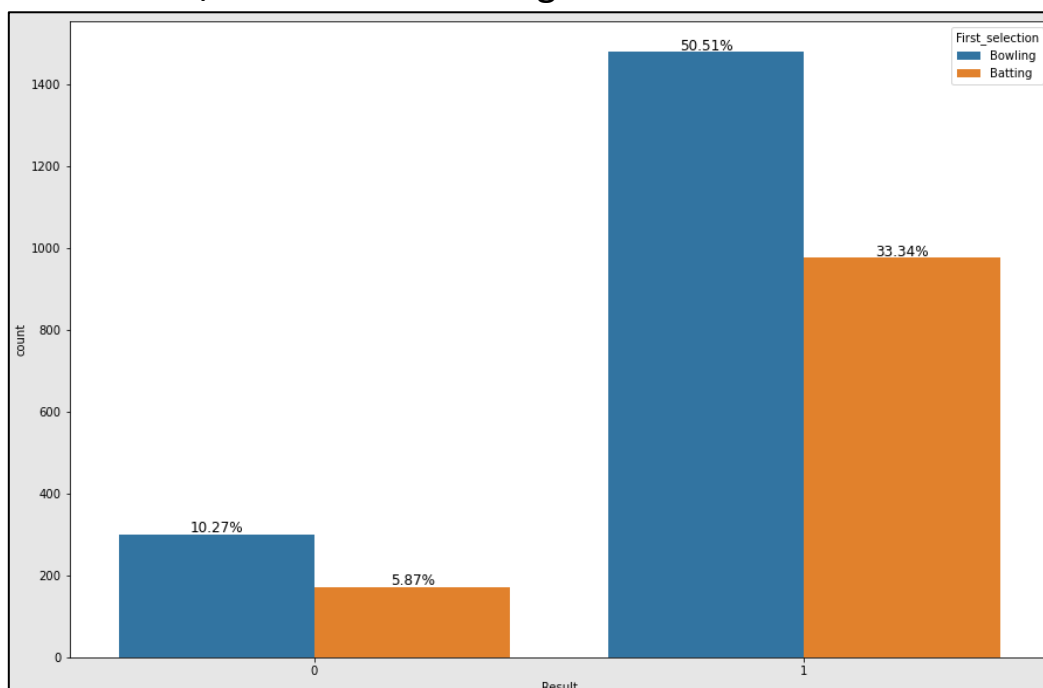
- India has won 31.2% of times with 3 number of bowlers in team, then 4 number of bowlers ie. 23% of times.

## Win/Loss Distribution against All rounders



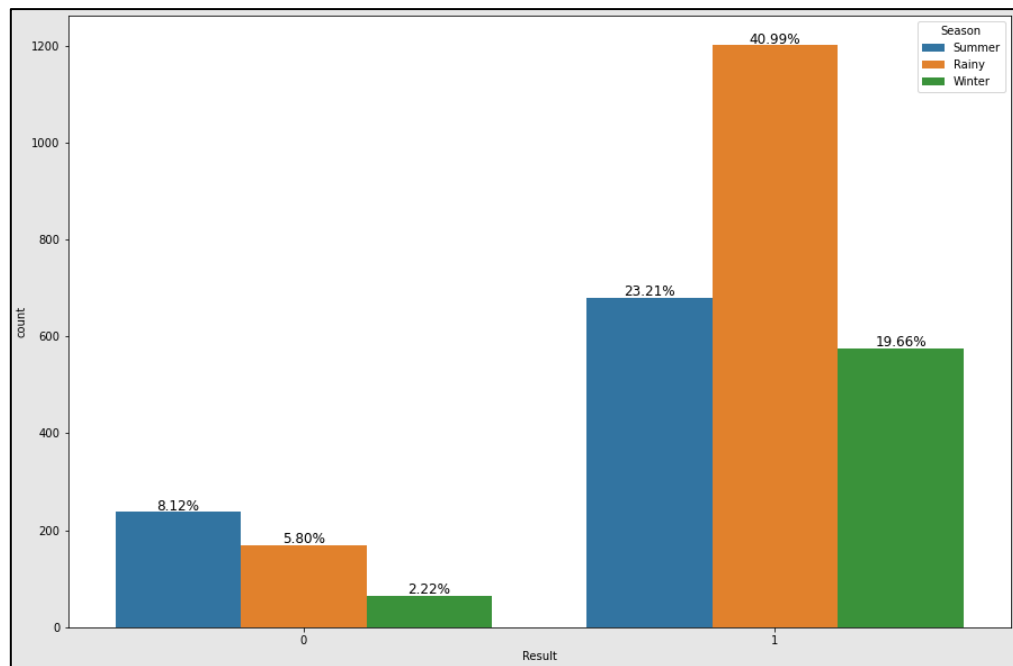
- India has won major matches with 3 number of all rounders in team 26.3% of times, 25.9% with 4 all rounders.

## Win/Loss Distribution against First Selection



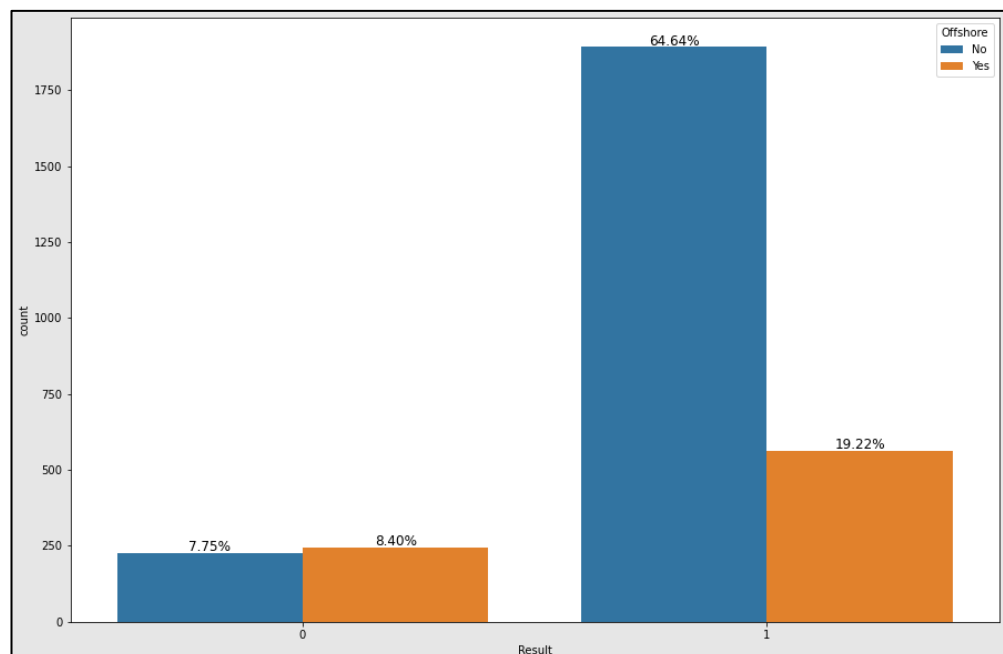
- India has won matches by selecting bowling first 50.15% of the time and by taking batting in beginning 33% of times.

Win/Loss distribution against Season Type



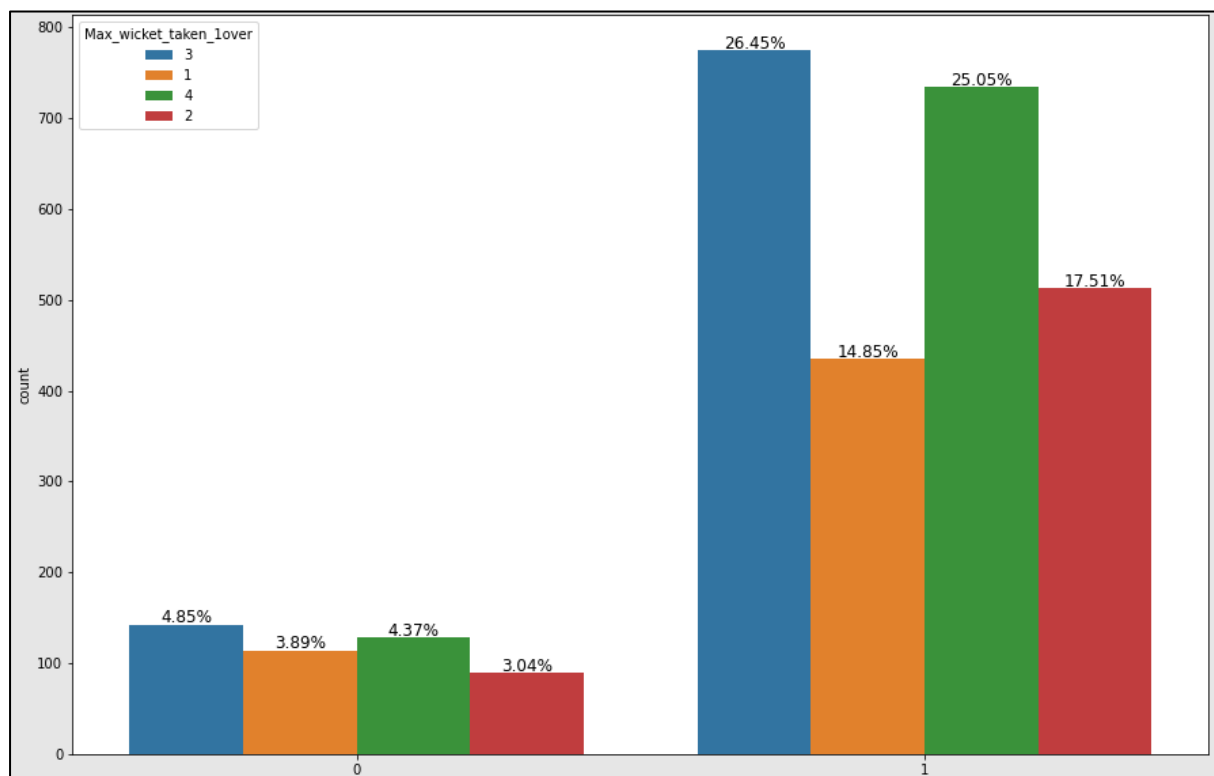
- India has won 41% of times in Rainy season then in Summer season 23.2% and Winter 19.66%

Win/Loss distribution against Offshore



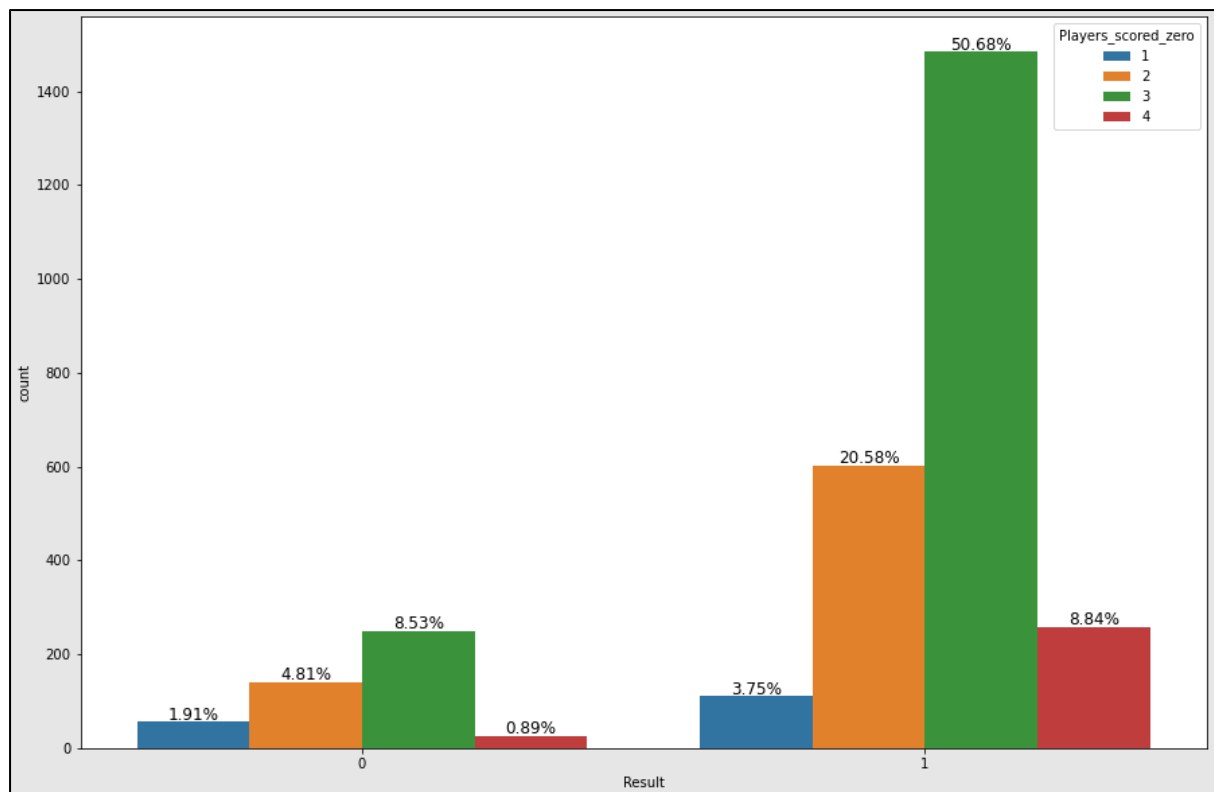
- India has won 64.7% of times when playing in India and only 19% when playing offshore.

### Win/Loss distribution against Max wicket taken in one over



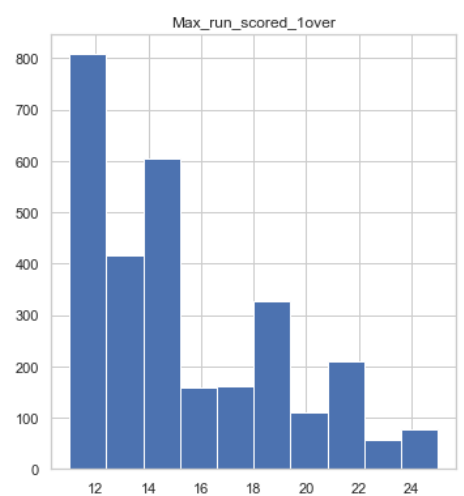
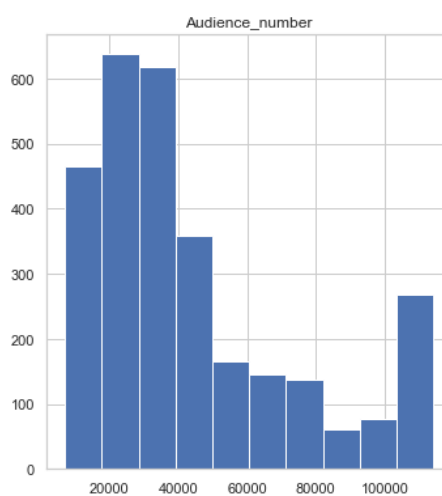
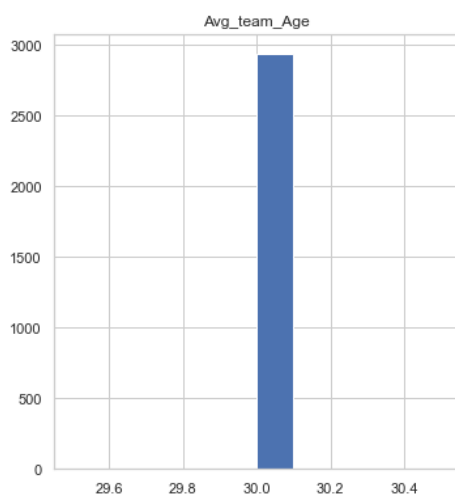
- India has won 26.5% of times with 3 wickets in an over, then 25% of times with 4 wickets in an over.

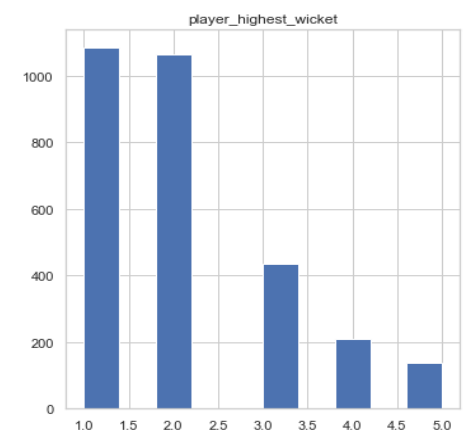
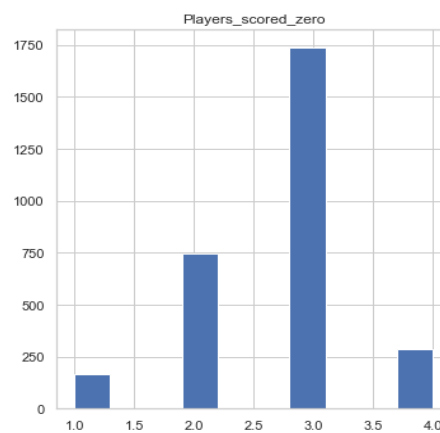
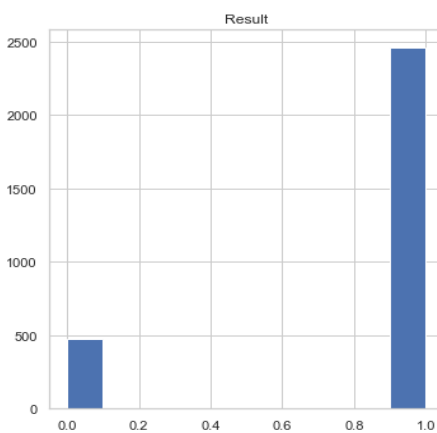
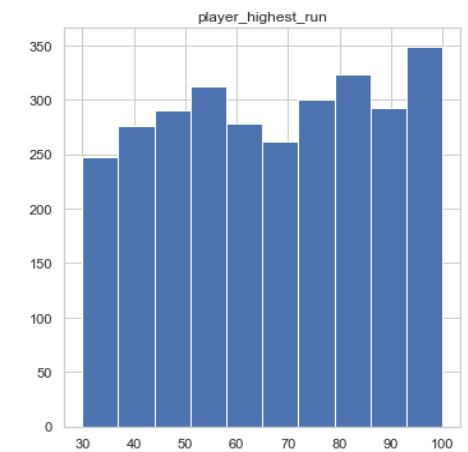
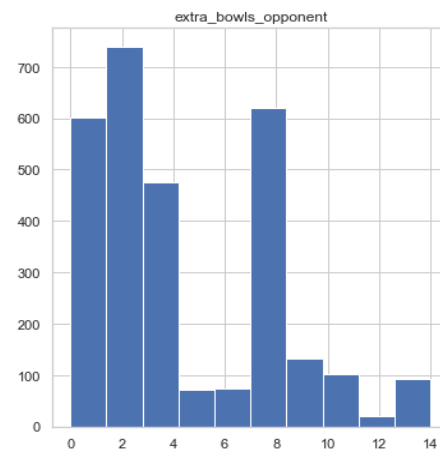
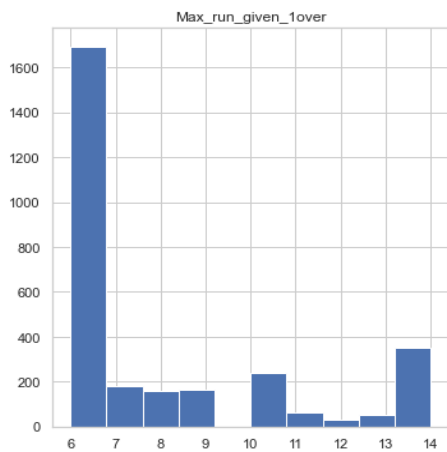
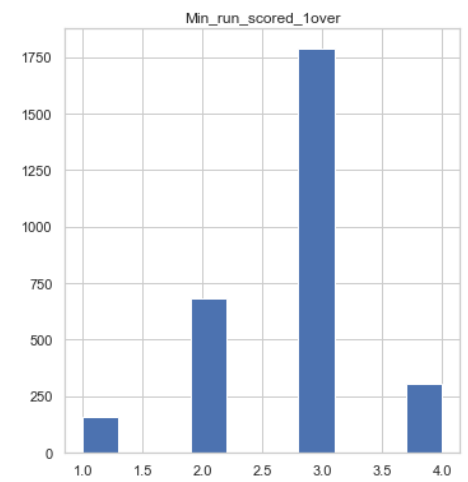
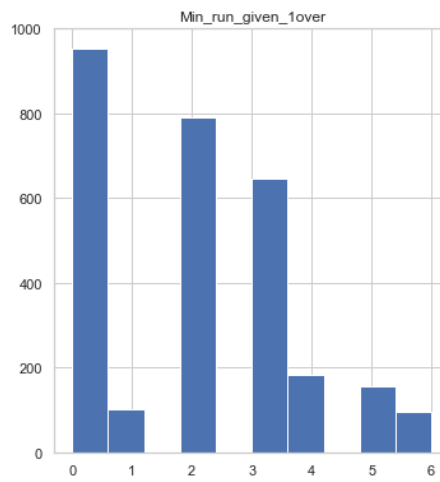
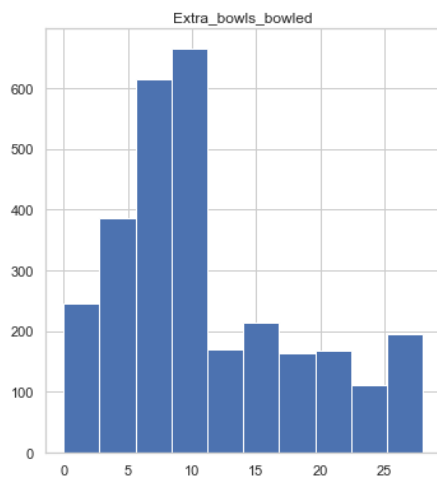
### Win/Loss Distribution against player\_scored zero



- India has won 50.7% of times with 3 player scoring zero and 20.6% of times with 2 player scoring zero.

## Histogram distribution

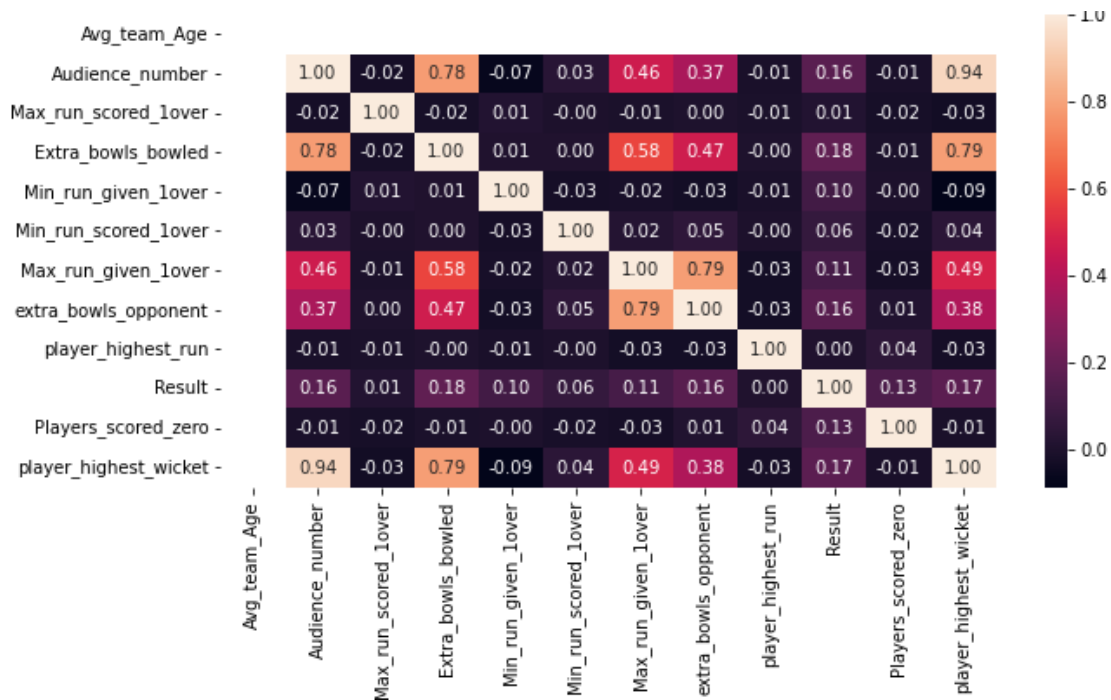




Out of all the histograms available above Player Highest run and Audience number are continuous and have a distribution on histogram. For the features other than the mentioned are showing discrete nature. Player highest run is a uniform distribution whereas audience number is a skewed distribution to the right.

By Seeing the Bi Variate Analysis we will get to have better understanding

## Bivariate Analysis

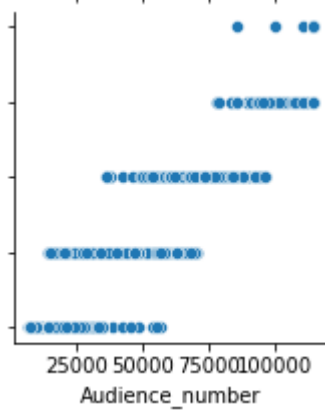


From the heat map we cannot see any significant correlation of any feature with our target variable that is the result. Although we can see some significant correlation between the features.

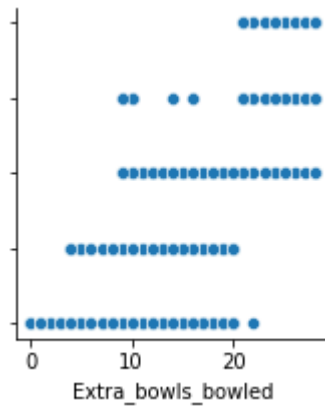
1. Audience number and player highest wicket-94%
2. Players highest wicket and extra bowls bowled-79%

But both these correlations are for discrete in nature as we can see in the scatterplot.

For 1<sup>st</sup> significant correlation

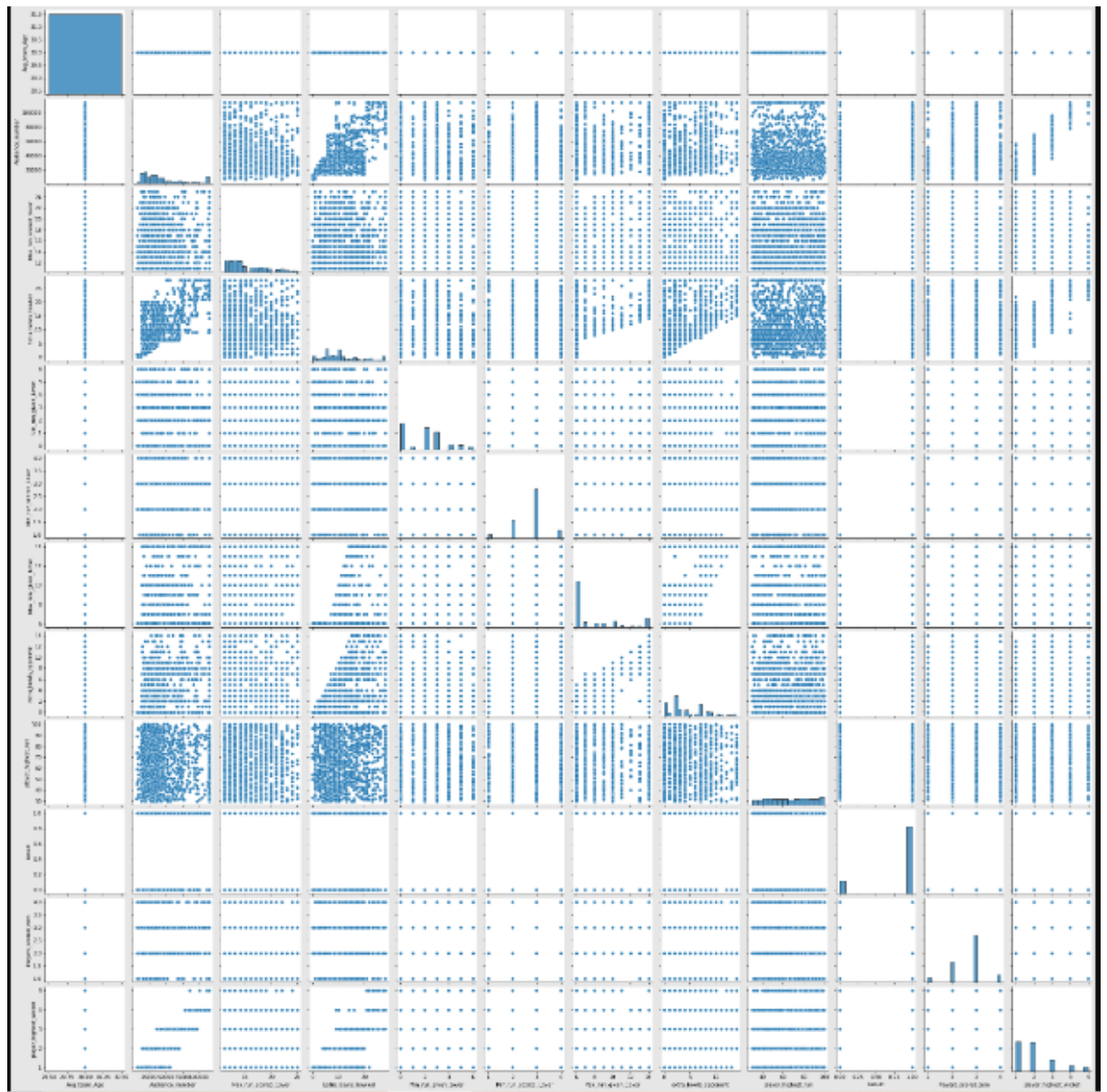


For 2<sup>nd</sup> significant Correlation



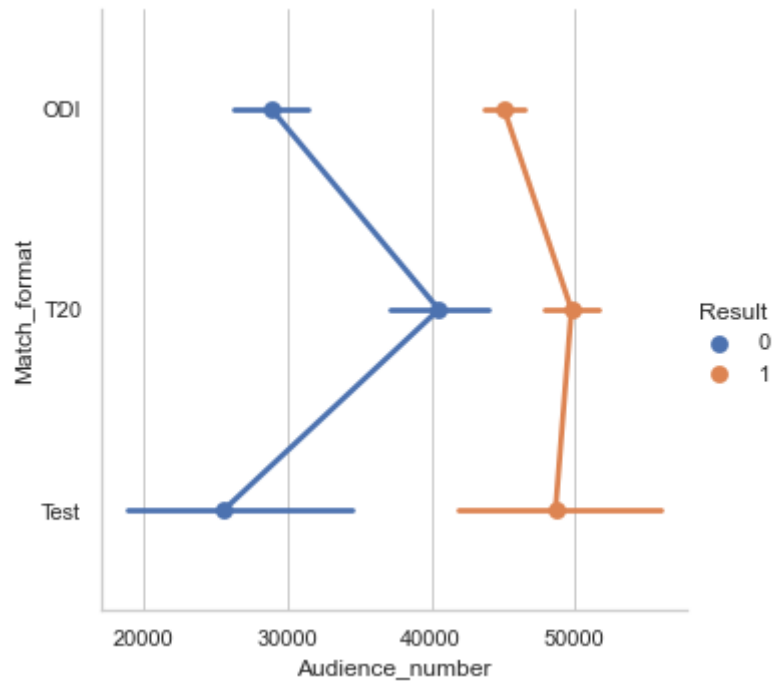
**From the pair plot we can see that the features don't have a significant linear relationship between the variables.**





## Multivariate Analysis

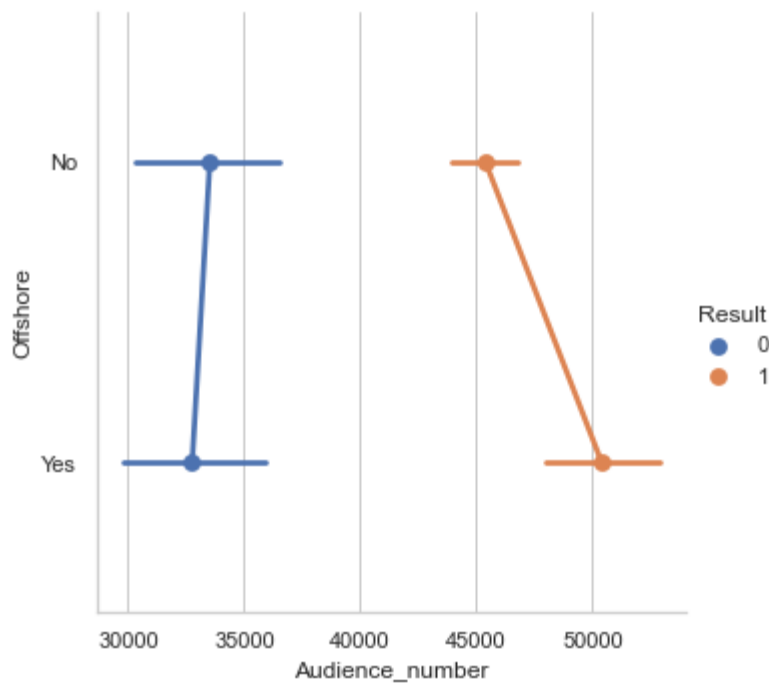
Relationship between audience number match win/loss and match\_format



From this we can interpret quite a few things

1. India won test, T20 match when its audience approx. 42000 to 55000.
2. For ODI it is between 42000 to 47000 approx

So from this we can assume that large audience really effects team performance and morale and may boost their chances of win. Lets confirm it with Offshore as well.



Here we can see that when India is playing at home with audience is around 47000 to 55000, which is quite acceptable as Cricket has huge fanbase in India, so India has won matches at that level of audience.

Although we didn't find any correlation with these features but we can bet on these features to boost Indian team performance

## **Impact on Business problem**

As far for our target column that is the result column, we cannot find any variable effecting with high correlation.

Although with Univariate, bivariate and multivariate analysis we can take some insights which we can use to improve our odds of winning for different matches. Although feature impact and their insights will reflect differently when we will see how machine learning models finding these features important.

1. India always performs best in when Playing home from every opponent.
2. India performs best when playing day matches.
3. With Audience of more than 40000, India performs best when playing offshore as well as for different match formats.

## 4. Model Building and Validation

### Model Performance, Model Selection

After EDA and cleaning the data, now our main task is training machine learning models in order to predict Loss and win of matches.

So we have set of predictive models which can help us in predicting this binary classification of Win and Loss. In this project we proceeded with training below mentioned models till we got the best result we want.

- a. Logistic Regression
- b. Naïve Bayes
- c. Decision Tree
- d. Bagging- Ensemble technique
- e. Boosting- Ensemble technique
- f. XGboost- Ensemble Technique

These models are great for classification problems like these on which we are working on. This model building sequence is quite essential as logistic regression have some linear regression assumptions and even Naïve Bayes has independent event assumption which kind make them less capable when working in different problems. Hence from these we move to decision tree and random forest models like bagging etc. which can handle complex data with less assumptions.

Before starting building models we first use **one hot encoding** in order to convert categorical data into numerical data. After encoding then we applied **Recursive feature elimination with Cross validation for selecting optimal number of features with cross validation.**

Optimal features coming are below:-

1. 'Min\_run\_given\_1over',
2. 'Min\_run\_scored\_1over',
3. 'Max\_run\_given\_1over',
4. 'extra\_bowls\_opponent',
5. 'Players\_scored\_zero',
6. 'Match\_light\_type\_Day',
7. 'Match\_light\_type\_Day  
and Night',
8. 'Match\_light\_type\_Night',
9. 'Match\_format\_ODI',
10. 'Match\_format\_Test',
11. 'Bowlers\_in\_team\_3.0',
12. 'Bowlers\_in\_team\_4.0',
13. 'Bowlers\_in\_team\_5.0',
14. 'Bowlers\_in\_team\_nan',
15. 'All\_rounder\_in\_team\_1.0'  
,
16. 'All\_rounder\_in\_team\_2.0'  
,
17. 'All\_rounder\_in\_team\_3.0'  
,
18. 'All\_rounder\_in\_team\_4.0'  
,
19. 'All\_rounder\_in\_team\_nan',
20. 'First\_selection\_Batting',
21. 'First\_selection\_Bowling',
22. 'Opponent\_Australia',
23. 'Opponent\_Bangladesh',
24. 'Opponent\_England',
25. 'Opponent\_Pakistan',
26. 'Opponent\_South Africa',
27. 'Opponent\_Srilanka',
28. 'Opponent\_West Indies',
29. 'Opponent\_Zimbabwe',
30. 'Season\_Rainy',
31. 'Season\_Summer',
32. 'Season\_Winter',
33. 'Offshore\_No',
34. 'Offshore\_Yes',
35. 'Max\_wicket\_taken\_1over  
\_1',
36. 'Max\_wicket\_taken\_1over  
\_2',
37. 'Max\_wicket\_taken\_1over  
\_3',
38. 'Max\_wicket\_taken\_1over  
\_4'

We used the above selected features for model training.

For the comparison we will take two metrics

1. F1 score
2. Accuracy

F1 score is harmonic mean of both precision and recall hence it will take the effect of both. Also F1 score metric will be used to check for overfitting.

During model building we observed sample imbalance.

<b>Loss(0)</b>	473
<b>Win(1)</b>	2457

This quite a significant sample imbalance

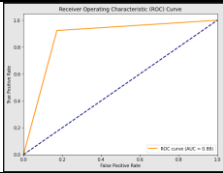
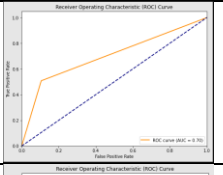
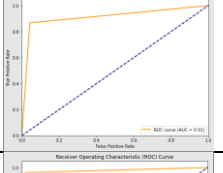
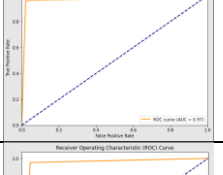
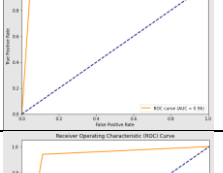
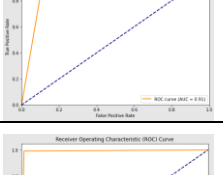
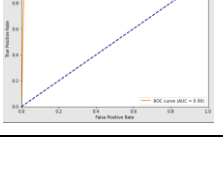
Hence we applied oversampling technique called SMOTE in order to balance the data. After applying the SMOTE the data looks like as follows

<b>Loss(0)</b>	2457
<b>Win(1)</b>	2457

After this we trained the models on balanced data

## Model Performance Comparison

Each model was checked for **overfitting** and the its result was validated using **5 fold Cross Validation** and ROC-AUC curve. Below table shows the performance of each model with best parameters to avoid overfitting.

Model	Parameters	Accuracy	F1 Score	Cross Validation mean accuracy	ROC AUC Curve
Logistic Regression	-	0.88	Loss(0)-0.87 Win(1)-0.88	0.86	
Naïve Bayes	-	0.7	Loss(0)-0.75 Win(1)-0.63	0.7	
Decision Tree	Max_depth =12	0.91	Loss(0)-0.92 Win(1)-0.91	0.9	
Bagging	-	0.93	Loss(0)-0.91 Win(1)-0.92	0.95	
Adaboosting	Max_depth =12	0.96	Loss(0)-0.96 Win(1)-0.96	0.91	
Gradient Boost	-	0.91	Loss(0)-0.91 Win(1)-0.92	0.88	
XGboost	'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200	0.98	Loss(0)-0.98 Win(1)-0.98	0.96	



Out of all the models we built **XGboost** model performs the best in cross validation with highest mean accuracy of 0.96 and accuracy of 0.98.

### **Model Tuning**

For Building XGboost model we used hyper-parameter tuning using below parameters.

'n\_estimators': [100, 200, 300],

'learning\_rate': [0.01, 0.1, 0.2],

'max\_depth': [3, 4, 5],

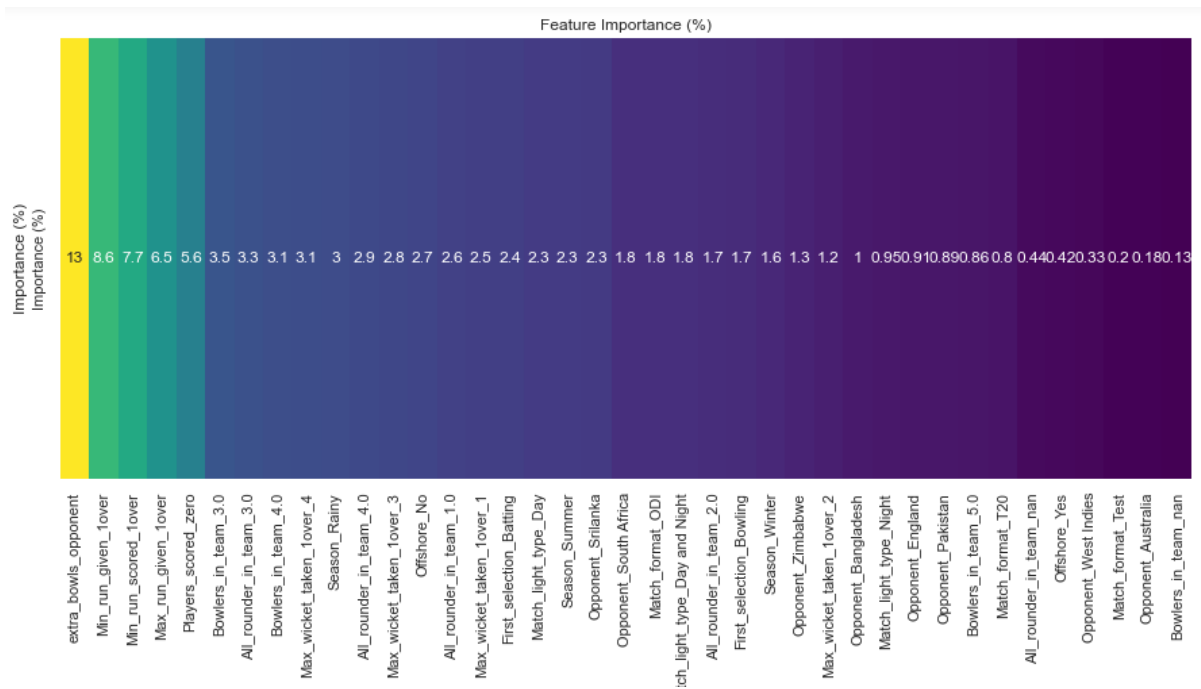
Used Grid Search with Cross Validation and got the best parameters of performance.

**'learning\_rate': 0.2, 'max\_depth': 5, 'n\_estimators': 200**

Using these parameters we have built our model.

### **Feature Importance and Prediction.**

After creating the beat model on our data we found the main important features which model finds the most I order to reduce the complexity of model and get better prediction using those. We know values for only few future for future matches and for model to perform best in prediction requires all the features values on which it I trained. So for that we found the important features by model.



From this heat map we got 5 top important features

1. extra\_bowls\_opponent
2. Min\_run\_given\_1over
3. Min\_run\_scored\_1over
4. Max\_run\_given\_1over
5. Players\_scored\_zero

Now we will use all the values for these features from data and using these values we will get win probabilities for our future matches.

Values we found in the data for above features:-

extra\_bowls\_opponent\_values = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14]

Min\_run\_given\_1over\_values = [0, 1, 2, 3, 4, 5, 6]

Min\_run\_scored\_1over\_values = [1, 2, 3, 4]

Max\_run\_given\_1over\_values = [6, 7, 8, 9, 10, 11, 13, 14]

Players\_scored\_zero\_values = [1, 2, 3, 4]

## 5. Final Interpretations and Implementation

Using the final important features we predicted win probability got the best result for ultimate win against the teams India is going to play.

For consideration we took win probability greater than 0.5

### **For Match against England**

extra\_bowls\_opponent- 9

Min\_run\_given\_1over- 6

Min\_run\_scored\_1over- 4.

Max\_run\_given\_1over- 13

Players\_scored\_zero- 2

### **For Matches Against Australia**

extra\_bowls\_opponent- 9.000000

Min\_run\_given\_1over - 6.000000

Min\_run\_scored\_1over- 4.000000

Max\_run\_given\_1over - 13.000000

Players\_scored\_zero - 2.000000

### **For Matches Against Srilanka**

extra\_bowls\_opponent- 5.000000  
Min\_run\_given\_1over- 6.000000  
Min\_run\_scored\_1over- 3.000000  
Max\_run\_given\_1over- 8.000000  
Players\_scored\_zero- 4.000000

Based on the analysis, we have identified specific combinations of important features that maximize the win probability for upcoming matches against England, Australia, and Sri Lanka. These combinations represent the conditions under which the Indian cricket team is more likely to win.

For the match against England, the optimal conditions include:

Extra Bowls by the Opponent: 9  
Minimum Runs Given in 1 Over: 6  
Minimum Runs Scored in 1 Over: 4  
Maximum Runs Given in 1 Over: 13  
Players Scoring Zero: 2

For matches against Australia, the recommended conditions are the same as those for England.

However, for matches against Sri Lanka, the optimal conditions are slightly different:

Extra Bowls by the Opponent: 5  
Minimum Runs Given in 1 Over: 6  
Minimum Runs Scored in 1 Over: 3  
Maximum Runs Given in 1 Over: 8  
Players Scoring Zero: 4

These conditions represent the specific scenarios where the Indian cricket team has the highest chances of winning. Implementing these strategies could lead to a more favorable outcome in these upcoming matches.

It's important to note that these conditions are based on historical data and statistical analysis. The actual performance in future matches may vary due to various factors, including changes in team composition, weather conditions, and opponent strategies. Therefore, these conditions should be considered as guidelines rather than guaranteed outcomes, and flexibility in strategy is essential to adapt to evolving match situations.

This is the result on the basis of data available to us which didn't show any significant correlation of result with other features like season etc. which can affect real world results.

