Capstone project

On

Cricket match prediction

By: - Pratheek U

1) Introduction of the business problem

a) Defining problem statement

BCCI has hired us as an external analytics consulting firm for data analytics. The major objective of this tie up is to extract actionable insights from the historical match data and make strategic changes to make India win. Primary objective is to create Machine Learning models which correctly predicts a win for the Indian Cricket Team. Once a model is developed then you have to extract actionable insights and recommendation.

b) Need of the study/project

The goal of a research proposal is to present and justify the need to study a research problem and to present the practical ways in which this research should be conducted. The design elements and procedures for conducting the research are governed by standards within the predominant discipline in which the problem resides, so guidelines for research proposals are more exacting and less formal than a general project proposal.

c)Understanding business/social opportunity

- As we know that cricket is a multinational sport and since the introduction of T-20 cricket and the inauguration of short game format tournaments like the Indian Premier League, the sport of cricket has become much more competitive and fast-paced. Teams constantly strive to be better every year and invest heavily in their resources.

- A cricket analysis is one such tool through which teams keep improving, and today, almost every international cricket team or a big-league team has a cricket analyst or a team of them.

- The primary role of a cricket analyst is to analyze a team and share his results with the team management. The analyst gathers data from a match and studies what are the key areas of improvement and how the team can work on them.

- A cricket analyst is expected to find out on which pitch a player does well on, which shot is a batsman good and bad at, which is the ideal length for a bowler and one on which he needs to work, find out key player battles or matchups before a game, etc.

- The analyst is expected to scout new and unheard talents; point out weaknesses of players and how they can work on them; help in creating match strategies before a game, and improve the overall team's performance.

- A cricket analyst is a very important part of the team and has a big impact on a team's failure or success

2)Data Report

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

The information was collected from 2930 matches that India played against different teams in different conditions

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

Data set have 2930 rows and 23 columns

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

This data set contain 10 object columns and other are integer columns

In [44]:
```
data.info()
```

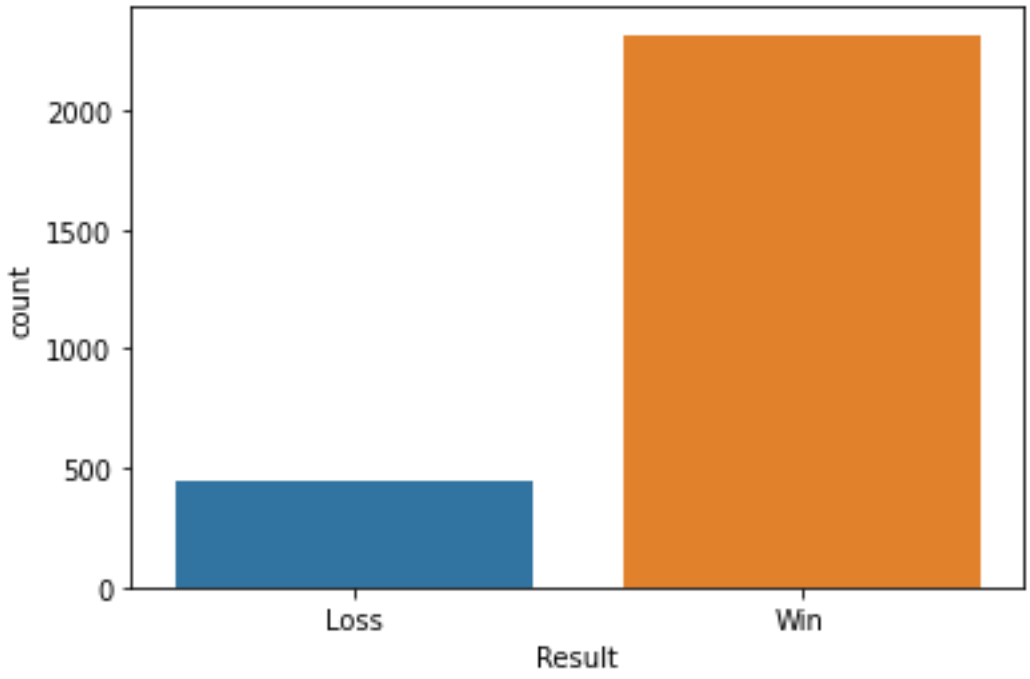
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2762 entries, 0 to 2929
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Game_number            2762 non-null   object
 1   Result                 2762 non-null   object
 2   Avg_team_Age           2762 non-null   float64
 3   Match_light_type       2762 non-null   object
 4   Match_format           2762 non-null   object
 5   Bowlers_in_team        2762 non-null   float64
 6   Wicket_keeper_in_team  2762 non-null   int64
 7   All_rounder_in_team    2762 non-null   float64
 8   First_selection        2762 non-null   object
 9   Opponent               2762 non-null   object
 10  Season                 2762 non-null   object
 11  Audience_number        2762 non-null   float64
 12  Offshore               2762 non-null   object
 13  Max_run_scored_1over   2762 non-null   float64
 14  Max_wicket_taken_1over 2762 non-null   int64
 15  Extra_bowls_bowled     2762 non-null   float64
 16  Min_run_given_1over    2762 non-null   int64
 17  Min_run_scored_1over   2762 non-null   float64
 18  Max_run_given_1over    2762 non-null   float64
 19  extra_bowls_opponent   2762 non-null   int64
 20  player_highest_run     2762 non-null   float64
 21  Players_scored_zero    2762 non-null   object
 22  player_highest_wicket  2762 non-null   object
dtypes: float64(9), int64(4), object(10)
memory usage: 517.9+ KB
```

3)Eda

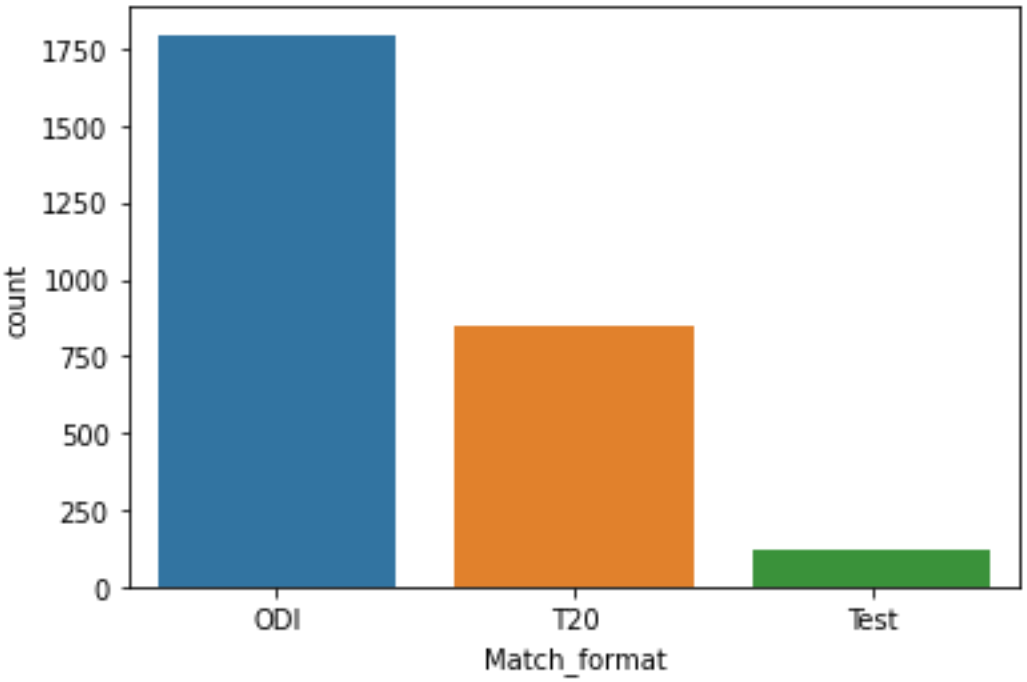
a) Univariate analysis

Result

- In last 2930 matches , we analyze that our team won around 2900 matches and we lost around 480 matchs.

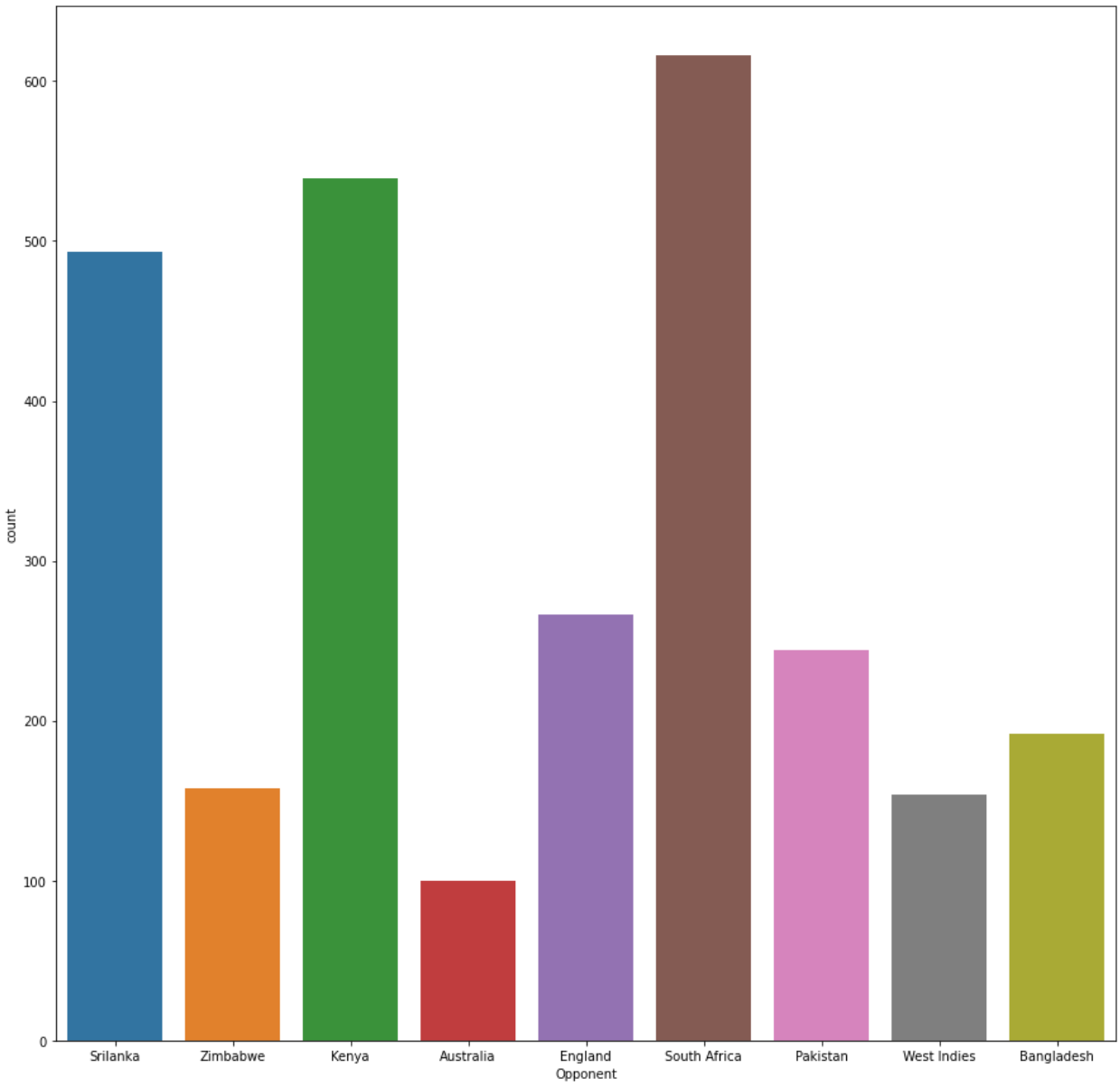


2)Match format



- This figure state that we have 3 formats of the game.
- The most played format is one day international (ODI).
- The least played format is test matches.
- The fast-growing format is T20. we called it as modern savior of the game.

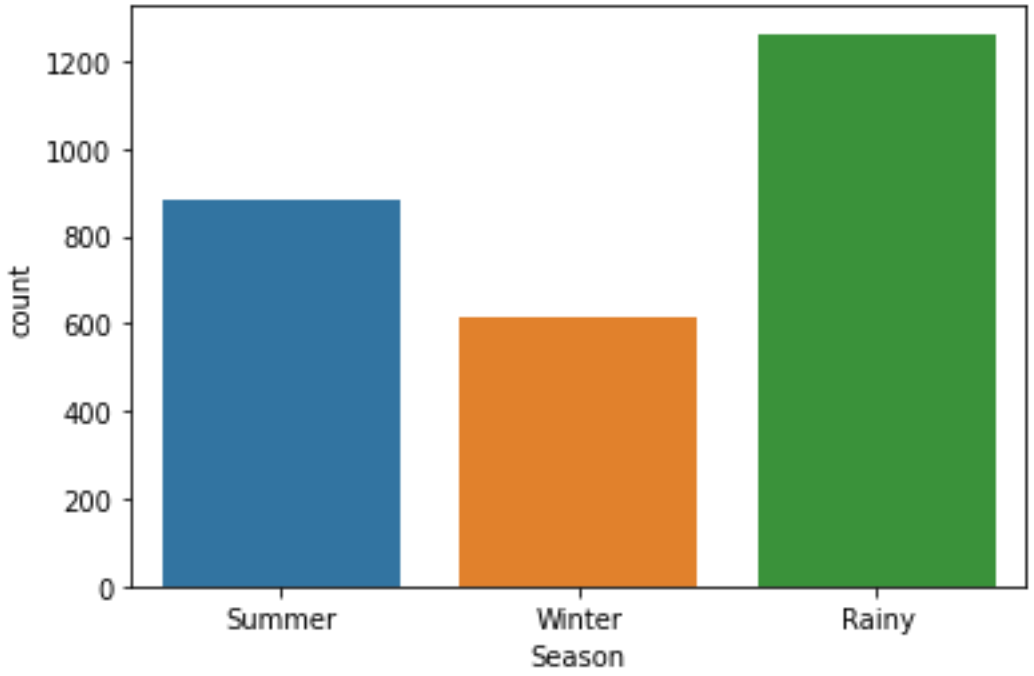
3)Opponent



- It state that we played our most number of matches with South Africa , Kenya , Sri Lanka,

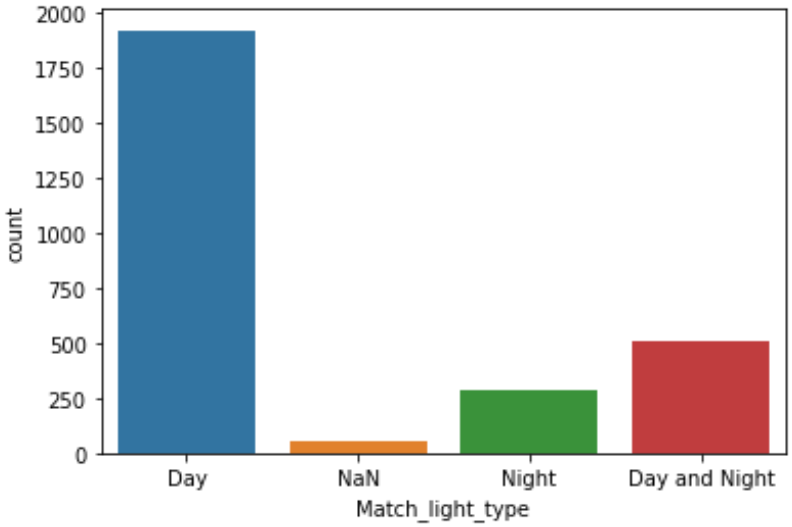
- We played least number of matches with Australia , west indies , Bangladesh

4) Season



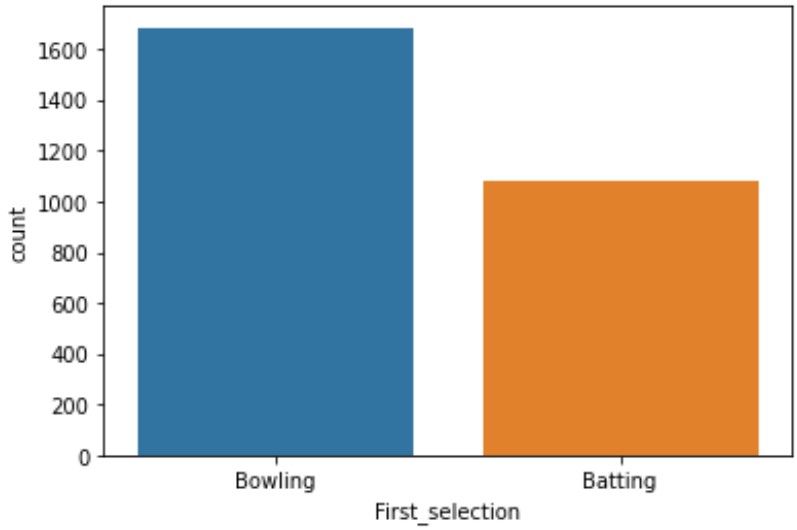
- Is state that mostly our matches are played during rainy seasons
- While summer hold 2 sopt and winter season hold 3 spot

5) Match_light_type



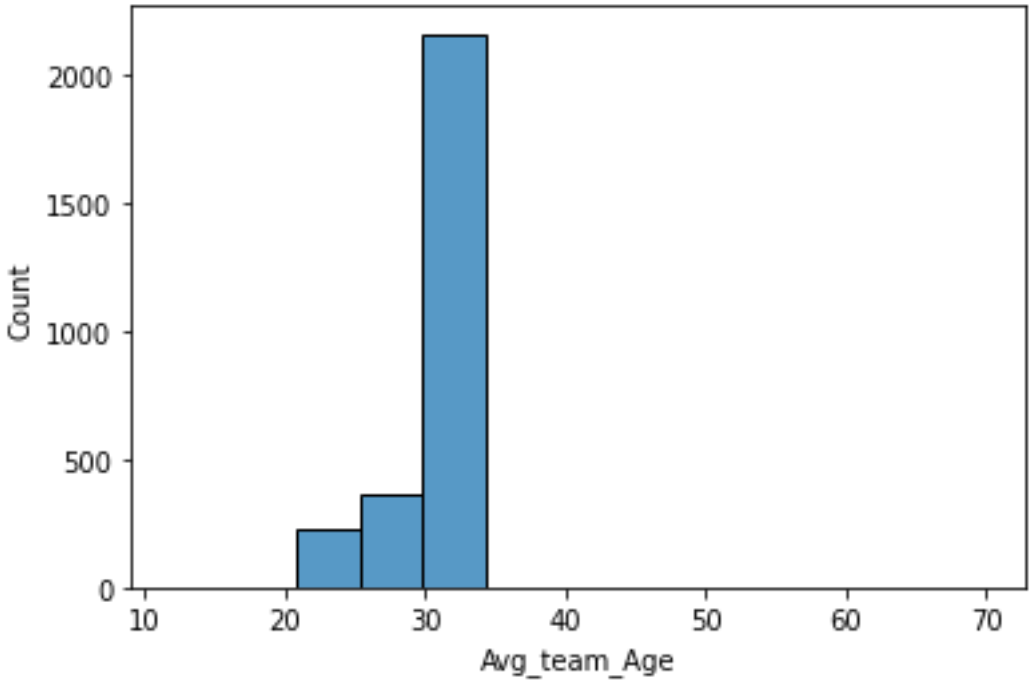
- Mostly our matches like test and ODI are played in between sunlight's.
- But format like T20 are played at night.
- Day and night are suitable for ODI's.

6) First_selection



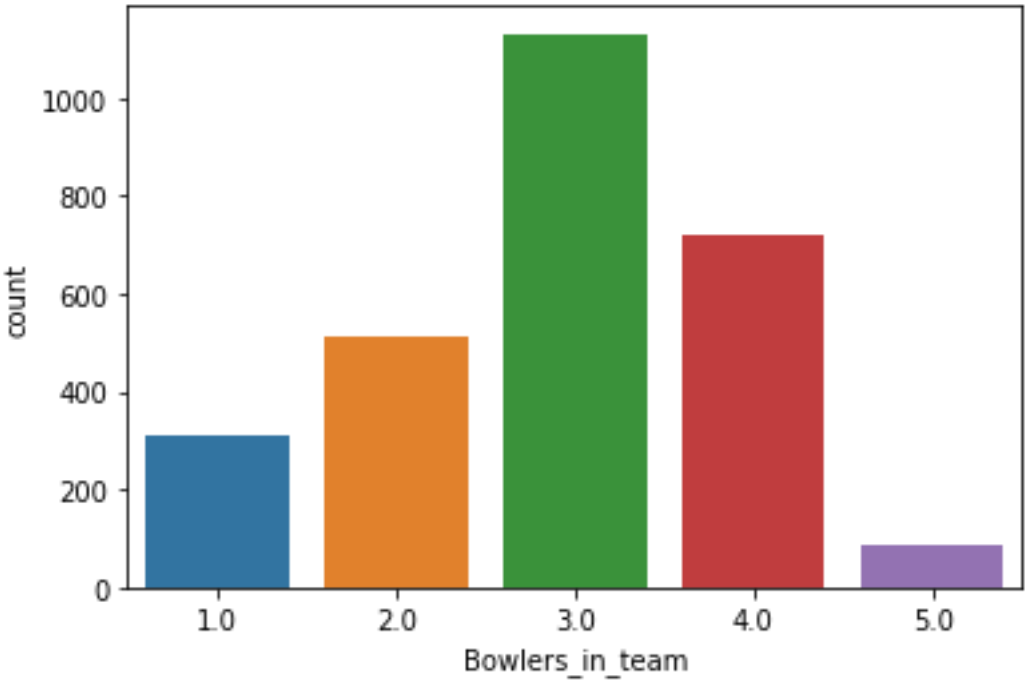
- Mostly team select bowling as there main preference after winning the toss
- They prefer to chase the target

7) Average age in team



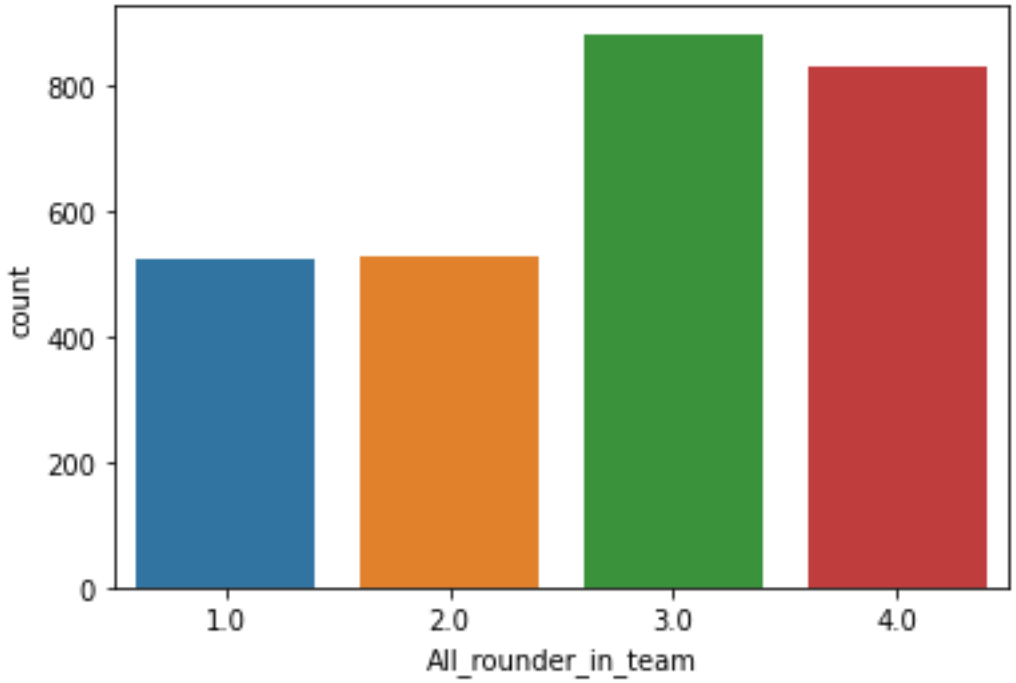
- The mean age in team is 29.
- And mostly player are above 30 years.

8) **Bowler in team**



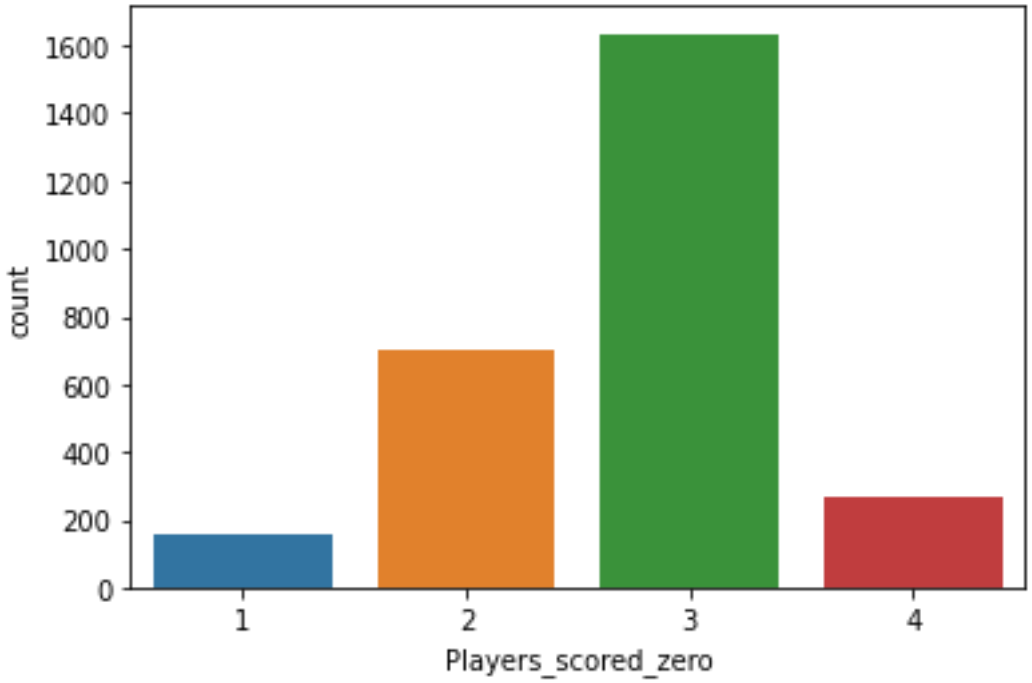
- Teams mostly opt 3and 4 bowlers in there team.

9) **All rounder in team**



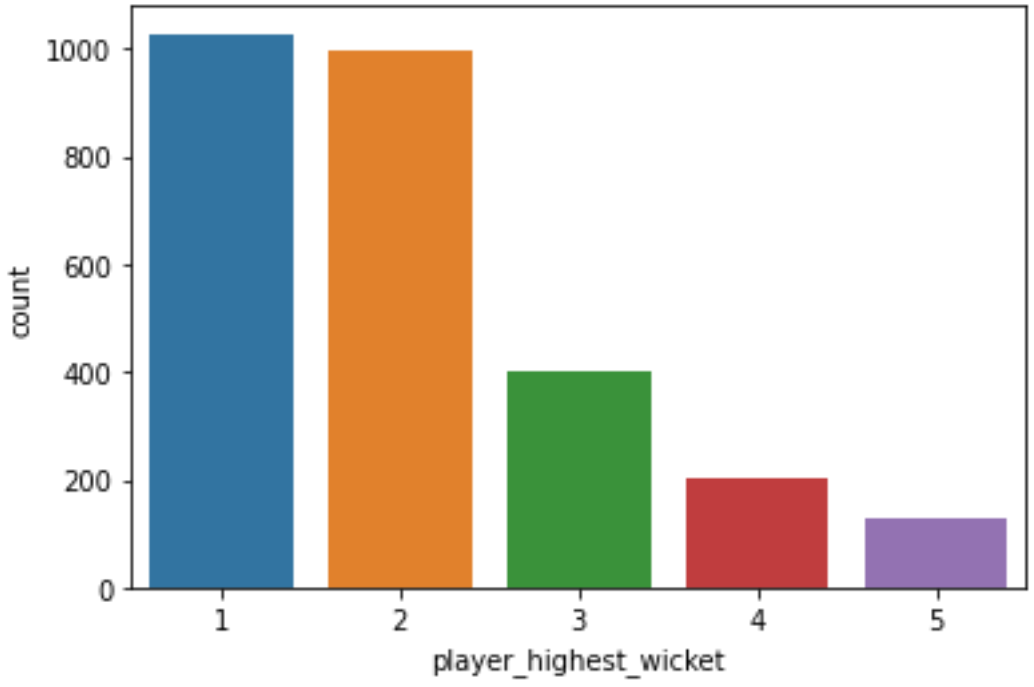
- Choice of all rounder in team is not that smooth according to situation team prefer all rounder in team.

10) Players_scored_zero



- In around 1600 matches 3 players out on zero
- While we saw a huge difference between others.

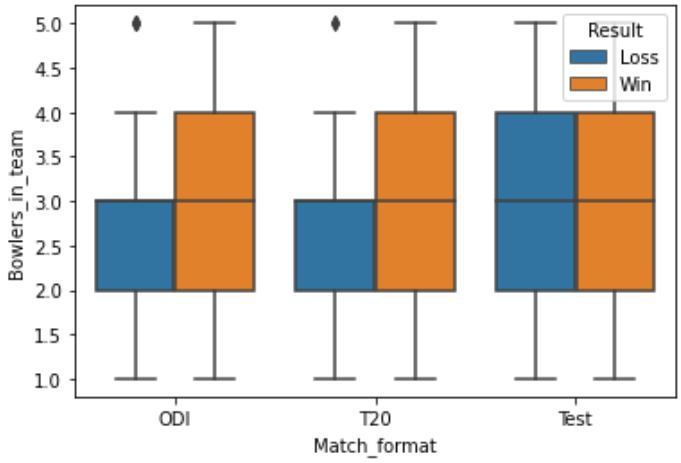
11) player_highest_wicket



In around 200 matches we saw one or 2 wickets are highest a bowler can pick.

Bivariate analysis

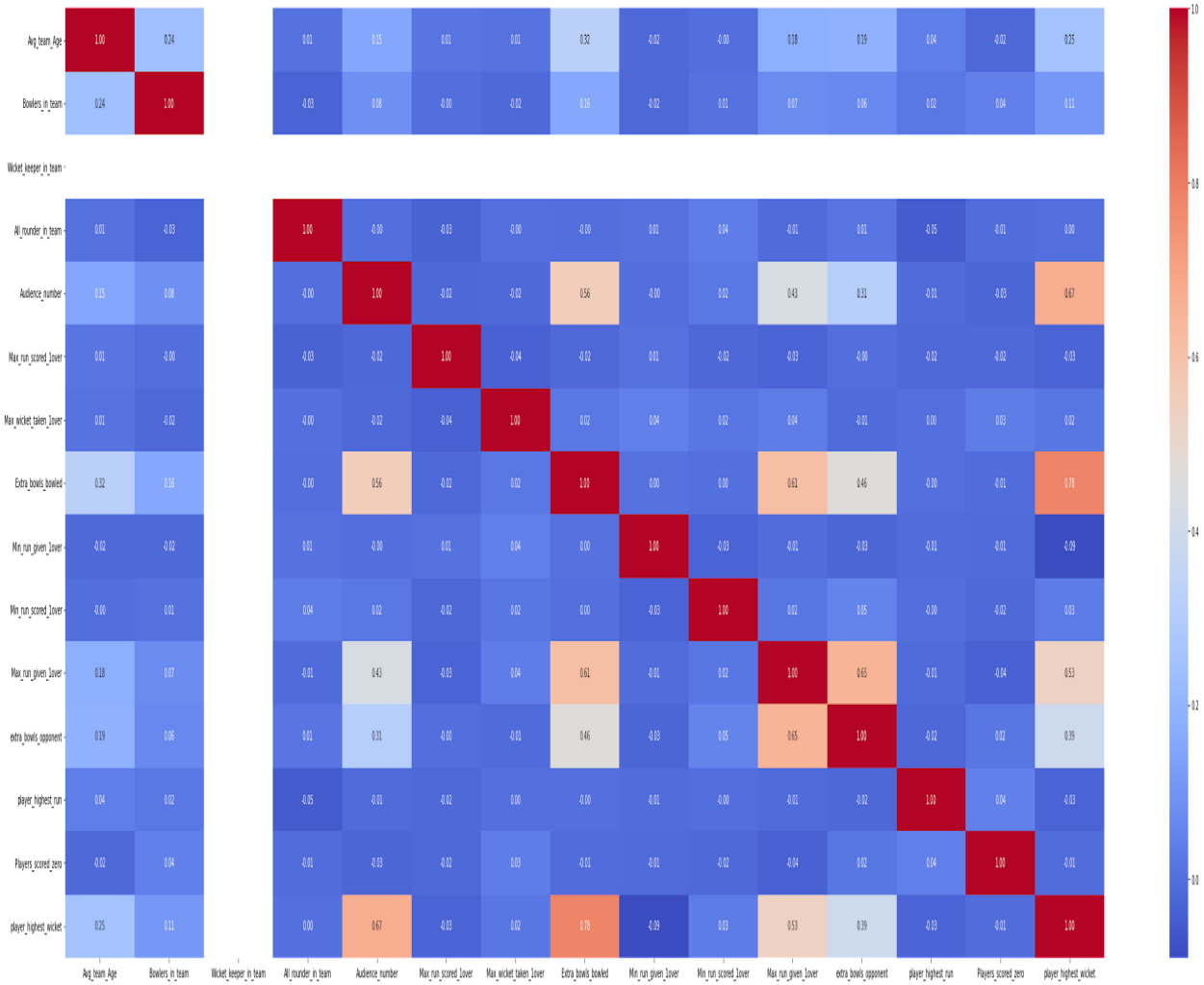
1) Match format vs bowler in team vs result



- In T20 and ODI the more number of bowlers we have there are high chance of winning the match.
- But in test match we don't saw any impact of bowlers in winning or losing a match

Correlation

- There is high correlation between extra balls bowled and highest wicket taken by player.
- There is high correlation between audience number and extra balls bowled, highest run in 1 over.
- Minimum runs given in one over and format of match are highly correlated.
- Extra balls bowled by opponent and high runs scored in 1 over are correlated.
- Result is correlated with onshore or offshore.



Removal of Unwanted variable

The three columns may be removed for the analysis,

- 'Game_number' : This is just a running serial number for each match
- 'Wicket_keeper_in_team': All the matches has 1 as value hence may be removed.

- “Off shore”

Missing Value Treatment

- There are Null Values in the following variables,
- 'Avg_team_Age',
- 'Bowlers_in_team',
- 'All_rounder_in_team',
- 'Audience_number',
- 'Max_run_scored_1over',
- 'Extra_bowls_bowled',
- 'Min_run_scored_1over',
- 'Max_run_given_1over',
- 'player_highest_run'
- The missing values are treated with mean , median and mode of the respective column.

Outlier Treatment

- There are outliers present in Averege Age, Extra balls bowled, Audience number, extras_bowled_opponent, Max_runs_given_1over.
- As dropping outliers may lead to loss of data it can be treated with mean imputation method in this case**.**

Variable Transformation

- In these two variable player_highest_wicket, Players_scored_zero, there are values with Three which are replaced with 3 and transformed to Numeric variable.
- Opponent, There are 9 opponents and they are encoded as follows,

  Country Code
  South Africa 1
  Kenya 2
  Srilanka 3
  England 4
  Pakistan 5
  Bangladesh 6
  Zimbabwe 7
  West Indies 8
  Australia 9
- In Offshore, No is encoded as 0 and Yes is encoded as 1.
- In Season, Rainy is encoded as 1, Summer is encoded as 2, Winter is encoded as 3.
- In First Selection, Bowling is encoded as 1 and Batting is encoded as 2. Also bat has been encoded as 2

- In Match format, ODI is encoded as 1, T20 is encoded as 2, Test is encoded as 3. Also 20-20 has been encoded as 2
- In Match_light_type, Day is encoded as 1, Day and Night is encoded as 2, and Night is encoded as3
- In Result, Win is encoded as 1 and Loss is encoded as 0.
  Also all these variables are converted into integer

4) Business insights from EDA

- The team played with most of the matches with average age of team is 30 and also the win% is better compared with any other average age. This may have a good mix of experienced and young players.
- The losing% is less in winter however the team has played less winter matches compared with other seasons.
- India has played Rainier season matches both in Offshore and onshore
- Summer Offshore matches the win% is very less almost 50% only.
- The team has played more day matches that is 69% than other formats. Though the win% more in ight matches the total number of matches is less. Day and night matches has the maximum loss% of the three
- The team played most of the matches against South Africa and has a good probability of winning the game.
- The dataset has more ODI matches and win% is also more in ODI matches.
- The number of matches played outside the country is less than the no. of matches played within the country.

- The chances of losing the match are high when played outside the country compared to when played in the country.
- As the Number of all rounders in the team increases the loss% is decreases.
- When the no. of full time bowlers in the team is less than 5 then there are 50-50 chances of winning the game.
- As the Highest wickets taken by single player in match increases the chance of losing the match is very less.
- When the Highest wickets taken by single player in match is very less there is high chance of losing the match.
- Rainy matches in South Africa are more favorable to win than other matches. 30% of summer matches in South Africa the team has lost.
- Most of the matches the team has bowled first and the chance of losing is also more compared to batting first.

5) Model building and interpretation.

5.1) train test split: -

- We use sklearn library to split the data into 4 different matrix (x_train , x_test , y_train , y_test).
- Test_size = .30

5.2)Model building

- We use two different models in this 1) logistic model and other is 2) decision tree

1) Logistic model

- The accuracy score of logistic model is .85524 .
- But this is not sufficient for us to judge the model. So, we use many different metrics to evaluate the model
- The matrix we used was confusion_matrix, classification_report .
- According to confusion matrix, we can say that true positive values are only 8 and is this model there are 111 values which classify as type 1 error and we have 9(false negative) type 2 error (false positive)

```
In [97]: from sklearn.metrics import confusion_matrix, classification_report

In [98]: confusion_matrix(y_test,y_predict)

Out[98]: array([[  8, 111],
                [  9, 701]], dtype=int64)
```

- According to classification report, the precision and recall value are not up to the marks.

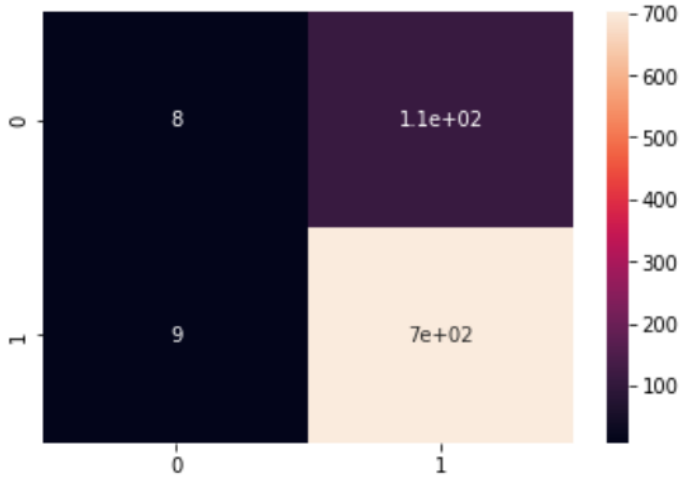
```
In [85]: print(classification_report(y_test , y_predict))
```

	precision	recall	f1-score	support
0	0.47	0.07	0.12	119
1	0.86	0.99	0.92	710
accuracy			0.86	829
macro avg	0.67	0.53	0.52	829
weighted avg	0.81	0.86	0.81	829

- This graph help us to understand the matrix.

```
n [105]: sas.heatmap(confusion_matrix(y_test,y_pred1),annot = True)

ut[105]: <AxesSubplot:>
```



2) Decicion tree

- The accuracy score of logistic model is .9022 .
- But this is not sufficient for us to judge the model. So, we use many different metrics to evaluate the model
- The matrix we used was confusion_matrix, classification_report .
- According to the confusion matrix the distribution of target variable are

```
]: confusion_matrix(y_test,y_pred)

]: array([[ 87,  32],
          [ 49, 661]], dtype=int64)
```

- As we clearly saw that the classification of target variable is way better than of logistic regression.
- According to classification report

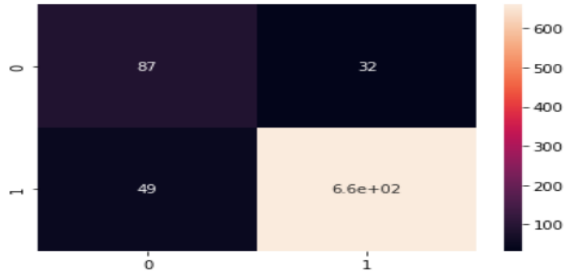
```
In [91]: print(classification_report(y_test , y_pred))
```

```
                  precision    recall  f1-score   support

               0       0.64      0.73      0.68       119
               1       0.95      0.93      0.94       710

        accuracy                           0.90       829
       macro avg       0.80      0.83      0.81       829
    weighted avg       0.91      0.90      0.90       829
```

```
In [100]: sas.heatmap(confusion_matrix(y_test,y_pred),annot = True)
Out[100]: <AxesSubplot:>
```



- The precision and recall are way more performing better than in logistic regression.

3)LDA

- The accuracy score of logistic model is .8673
- According to the confusion matrix the distribution of target variable are

```
In [118]: confusion_matrix(y_test,y1_pred)
Out[118]: array([[ 22,  97],
                 [ 13, 697]], dtype=int64)
```

- According to classification report

```
In [119]: print(classification_report(y_test , y1_pred))
```

	precision	recall	f1-score	support
0	0.63	0.18	0.29	119
1	0.88	0.98	0.93	710
accuracy			0.87	829
macro avg	0.75	0.58	0.61	829
weighted avg	0.84	0.87	0.83	829

- Performance of LDA is better than logistic but not better than decision tree

4)Random Forest

- The accuracy score we got was .86731
- The parameter we used in this were 'max_depth': [10], 'max_features': [11], 'min_samples_leaf':[10], 'min_samples_split' [50], 'n_estimators': [100]

- The below image show the classification report and confusion matrix.

```
Out[135]: 0.8673100120627262

In [136]: print(classification_report(y_test , y2_pred))
                  precision    recall  f1-score   support

               0       0.68      0.14      0.24       119
               1       0.87      0.99      0.93       710

        accuracy                           0.87       829
       macro avg       0.78      0.57      0.58       829
    weighted avg       0.85      0.87      0.83       829

In [137]: confusion_matrix(y_test,y1_pred)
Out[137]: array([[ 22,  97],
                 [ 13, 697]], dtype=int64)
```

5) KNN

- The accuracy score we got was .84731
- Parameter used n_neighbour = 15, weights = 'uniform', metric = 'minkowski'.
- The below Image give the detail about confusion matrix and classification report

```
: confusion_matrix(y_test,y3_pred)

: array([[  8, 111],
        [ 14, 696]], dtype=int64)

: print(classification_report(y_test,y3_pred))
```

	precision	recall	f1-score	support
0	0.36	0.07	0.11	119
1	0.86	0.98	0.92	710
accuracy			0.85	829
macro avg	0.61	0.52	0.52	829
weighted avg	0.79	0.85	0.80	829

6) Gaussian Naive

- The accuracy score we got was .80731
- The below Image give the detail about confusion matrix and classification report

```
C
```

	precision	recall	f1-score	support
0	0.34	0.42	0.38	119
1	0.90	0.87	0.88	710
accuracy			0.80	829
macro avg	0.62	0.64	0.63	829
weighted avg	0.82	0.80	0.81	829

```
confusion_matrix(y_test,y4_pred)
```

```
array([[ 50,  69],
       [ 95, 615]], dtype=int64)
```

7) xgboost

- Parameter used base_score = 0.5,
  colsample_bylevel = 1,
  colsample_bytree = 1,
  gamma = 0,
  learning_rate = 0.1,
  max_depth = 10,
  min_child_weight = 1,
  n_estimators = 100,
  objective ='binary:logistic',
  reg_alpha = 1,
  reg_lambda = 1,
  scale_pos_weight=1,
  subsample = 1
- The accuracy we got was.9445

```
In [167]: confusion_matrix(y_test,y5_pred)

Out[167]: array([[ 82,  37],
                 [  9, 701]], dtype=int64)

In [169]: print(classification_report(y_test,y5_pred))

              precision    recall  f1-score   support

           0       0.90      0.69      0.78       119
           1       0.95      0.99      0.97       710

    accuracy                           0.94       829
   macro avg       0.93      0.84      0.87       829
weighted avg       0.94      0.94      0.94       829
```

-

Important columns in datasets

Decision tree has special feature through which we can find out the importance and reliability of each columns / feature in the dataset.

```
       [ 49, 661]], dtype=int64)
```

```
[110]: # to find imp feature in data set
       print(pd.DataFrame(drt.feature_importances_,columns = [ "Imp"], index = x_t
```

	Imp
Avg_team_Age	0.024795
Match_light_type	0.025863
Match_format	0.002244
Bowlers_in_team	0.025793
All_rounder_in_team	0.062491
First_selection	0.013318
Opponent	0.050757
Season	0.048271
Max_run_scored_1over	0.110151
Max_wicket_taken_1over	0.067493
Extra_bowls_bowled	0.111958
Min_run_given_1over	0.044766
Min_run_scored_1over	0.064998
Max_run_given_1over	0.022629
extra_bowls_opponent	0.056641
player_highest_run	0.160665
Players_scored_zero	0.073594
player_highest_wicket	0.033572

```
In [ ]:
```

Model Tuning and business implication

Algo	Accuracy	Precision(0/1)	Recall(0/1)
Xgboost	0.9445	.90/.95	.69/.99
Logistic	0.85524	0.47/0.86	0.07/0.99
Decision tree	0.9022	0.64/.95	0.73/0.93
Lda	0.8673	0.63/0.88	0.18/.98
Random forest	0.8673	0.68/0.87	0.14/0.99
knn	0.8492	0.36/0.86	0.07/0.98
naive	0.8021	0.34/0.90	0.42/0.87

1) Interpretation of the most optimum model and its implication on the business

- Xgboost is the most convenient model with accuracy of .954 . which is highest among all the all models.
- The recall is also good .

- All the previous methods other than feature permutation are inconsistent! This is because they assign less importance

Implication on business

- When added together into an ensemble these weak models perform with excellent predictive accuracy. This performance comes at a cost of high model complexity which makes them hard to analyse and **can lead to overfitting**

6. Final interpretation / recommendation

- In T20 and ODI the more number of bowlers we have there are high chance of winning the match.
- Batting second have higher chance of winning
- Mean age of team players should be 29