Predicting The Final Score of First Innings Using Supervised Machine Learning at The End of 6 Overs of IPL Matches

By Naveen Ranapangu

Overview

1.Introduction

Cricket is a game of statistics. The **Indian Premier League** (IPL) is a T20 cricket league contested by 8 teams from 8 different cities of India. It was introduced by the Board of Control for Cricket in India (BCCI), in 2007. Each year, it is usually held between March and May. In IPL or T-20 cricket, multiple factors play a key role in estimating the final score of the innings. Some of the key factors include:

- Number of wickets left
- Number of overs/balls left
- Scores of the current batsman batting
- Runs scored in the last 5 overs?
- Wickets lost in the last 5 overs?
- The nature of the pitch (bowling pitch or batting), and
- How stronger the batting team and bowling team are

In this project, I **attempt to predict the final score** of the first innings at the end of 6 overs, using supervised machine learning. Regression analysis is to describe how well we can predict the behaviour of one variable based on the behaviour of another variable. I used regression analysis to predict the final score of the innings based on runs, wickets, striker, and non-striker, etc. As all the required features are not directly available from the data, hence I had to work on the data to get the required features from the data.

The models I used are Linear regression, Random forest, Lasso, Ridge, AdaBoost, Gradient Boosting, and XGB regressor. From all the models Linear Regression works best for the data with **a mean error of 12.79**. I have used residual plots for the Linear Regression to check the validation of the model. It has a high density of points close to the origin and a low density of points away from the origin. Residuals are independent and normally distributed.

2.Source of Input Data

The data is taken from the site Cricsheet that has ball by ball data for international and T20 cricket league matches for both men and women. I took the IPL data set for the prediction. The site provides the data in zip files. The IPL zip file contains 845 CSV files and one text file for all the matches of all the seasons from 2008 to 2021. Each CSV file is for a single match, contains the name of the batting team, bowling team, season and the date of the match happening, winner of the match, the man of the match, Toss winner, toss decision, umpires, match referee, and ball by ball data of the two innings of a match.

3.Data Cleaning

The problem with the data is that it has 845 CSV files each referring to a single match. I have combined all the CSV files into a single file and removed all the rows and columns from the data set that are no longer needed for the analysis. The text file contains all the match ids and the year of the match. I took all the years and combined them with the main data set. As a result, I got all the data of the ball by ball data for all 845 matches. Each file is saved with their corresponding match Id, I took the filename as a match id column for the corresponding CSV file.

The main data set consists of 2,00,664 rows and 19 columns. The data set consists of the following columns (features) :

1. **year:** Year of the match happening
2. **match_id:** Each match is given a unique number
3. **teams:** The playing teams (Batting and Bowling teams)
4. **innings:** Innings of each the match
5. **overs:** Overs of each match
6. **batting_team:** Batting team name
7. **bowling_team:** Bowling team name
8. **striker:** Batsman name who faced that ball

9. **non_striker:** Batsman name who is on another side of the crease
10. **bowler:** Bowler name who bowled that ball
11. **runs_off_bat:** Number of runs scored for that ball
12. **extras:** Extra run
13. **wides:** Extra type
14. **No balls:** Extra type
15. **byes:** Extra type
16. **leg byes:** Extra type
17. **penalty:** Is there any penalty for the ball
18. **kind_of_wicket:** Name of the player dismissal
19. **other_wicket_type:** Type of dismissal

```
In [32]: 1 ipl_matches.describe()
```

Out[32]:

	match_id	innings	runs_off_bat	extras	wides	noballs	byes	legbyes	penalty
count	2.006640e+05	200664.000000	200664.000000	200664.000000	6081.000000	810.000000	529.00000	3204.000000	2.0
mean	7.743539e+05	1.484013	1.242101	0.066449	1.206709	1.043210	1.84310	1.300250	5.0
std	3.143285e+05	0.503531	1.613072	0.340098	0.789025	0.364506	1.29975	0.839837	0.0
min	3.359820e+05	1.000000	0.000000	0.000000	1.000000	1.000000	1.00000	1.000000	5.0
25%	5.012350e+05	1.000000	0.000000	0.000000	1.000000	1.000000	1.00000	1.000000	5.0
50%	7.339790e+05	1.000000	1.000000	0.000000	1.000000	1.000000	1.00000	1.000000	5.0
75%	1.136561e+06	2.000000	1.000000	0.000000	1.000000	1.000000	4.00000	1.000000	5.0
max	1.254086e+06	6.000000	6.000000	7.000000	5.000000	5.000000	4.00000	5.000000	5.0

- There are **a maximum of 6 innings** of a match, it may be because the match was tied (i.e. 3*2 = 6)
- The maximum possible runs for a single ball is **six**

3.1 Preparing required features from raw data

As we don't have the following features I had to prepare from the data:

3.1.1 runs: Runs scored by the team at that instance

3.1.2 wickets: Total wickets fallen at that instance

3.1.3 Total_runs: Total runs scored by the team at the end of each innings

3.1.4 Striker_score: Score of the striker t that instance

3.1.5 Non-striker_score: Non-striker score at that instance

3.1.6 runs_last_5: Runs scored in the previous 5 overs

3.1.7 wickets_last_5: Wickets fallen in the previous 5 overs

4. Exploratory Data Analysis

Check the shape of the dataset:

```
In [53]:  1 # shape of the ipl data set
          2 ipl_matches.shape

Out[53]: (200664, 28)
```

After adding all the required features the data has 28 columns and 2,00,664 rows.

```
In [56]:  1 ipl_matches.describe()
```

Out[56]:

	runs	wickets	instant_wicket	Total_runs	striker_score	nonstriker_score	runs_last_5	wickets_last_5
	200664.000000	200664.000000	200664.000000	200664.000000	200664.000000	200664.000000	200664.000000	200664.000000
	74.245485	0.049162	2.418182	157.103790	17.819828	16.267507	33.338033	1.137494
	48.002383	0.216206	2.072966	29.949099	18.202190	17.582661	14.846165	1.074883
	0.000000	0.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000
	34.000000	0.000000	1.000000	138.000000	4.000000	3.000000	25.000000	0.000000
	70.000000	0.000000	2.000000	159.000000	12.000000	10.000000	34.000000	1.000000
	110.000000	0.000000	4.000000	177.000000	26.000000	24.000000	43.000000	2.000000
	263.000000	1.000000	10.000000	263.000000	175.000000	174.000000	113.000000	8.000000

- The highest score by a team is 263.
- The highest individual score in the history of the IPL is 175.

Checking for the null values and removing null columns with more than 60%

```
1 ipl_matches.isnull().sum().sort_values(ascending = False)
```

```
penalty             200662
byes                200135
noballs             199854
legbyes             197460
wides               194583
other_wicket_type   190799
kind_of_wicket      190799
non_striker              0
striker                  0
runs_off_bat             0
bowling_team             0
batting_team             0
overs                    0
innings                  0
teams                    0
match_id                 0
bowler                   0
wickets_last_5           0
extras                   0
runs_last_5              0
ball_score               0
runs                     0
wickets                  0
instant_wicket           0
Total_runs               0
striker_score            0
```

We can see that variables *penalty, byes, no balls, leg byes, wides, other_wicket_type, kind_of_wicket* have more than 60% of null values. We will remove those columns.

Check the consistent teams

year	Number of teams played
2008	8
2009	8
2010	8
2011	10
2012	9
2013	9
2014	8
2015	8
2016	8
2017	8
2018	8
2019	8
2020	8

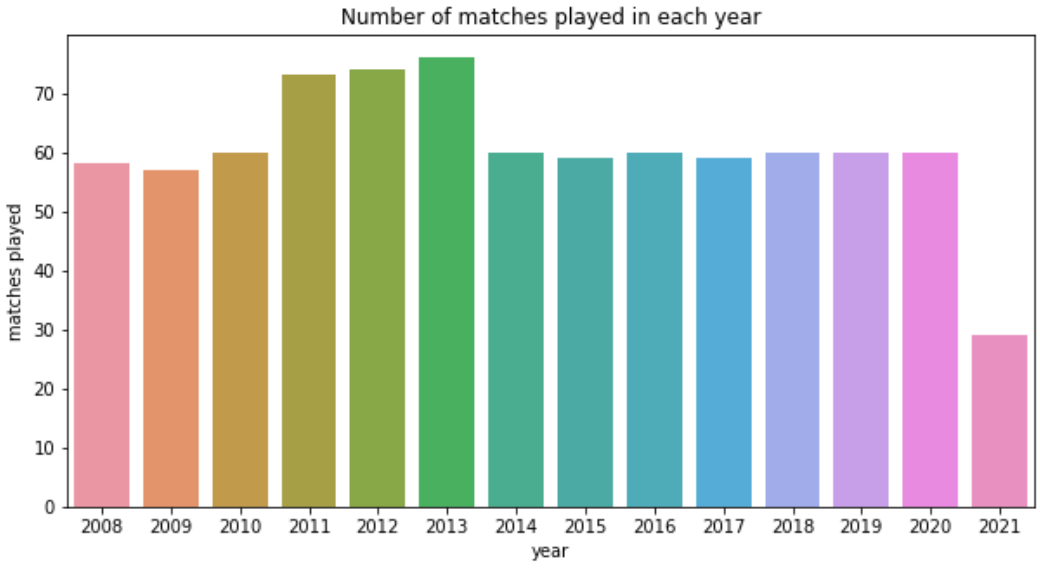
2021	8

In 2008 IPL started with 8 teams. In 2011 two new teams were added. In 2012 and 2013 one team was removed and there were a total of 9 teams. Again in 2014, one more team was removed, finally, till 2021 there are 8 consistent teams. For predicting the total first innings score, we will use only consistent teams.

```
In [59]: 1 ipl_matches.batting_team.unique()

Out[59]: array(['Kolkata Knight Riders', 'Royal Challengers Bangalore',
                'Deccan Chargers', 'Kings XI Punjab', 'Delhi Daredevils',
                'Chennai Super Kings', 'Mumbai Indians', 'Rajasthan Royals',
                'Pune Warriors', 'Kochi Tuskers Kerala', 'Sunrisers Hyderabad',
                'Rising Pune Supergiants', 'Gujarat Lions',
                'Rising Pune Supergiant', 'Delhi Capitals', 'Punjab Kings'],
               dtype=object)
```

- We need to update Delhi Daredevils to Delhi Capitals, as the Delhi Daredevils team changed their name to Delhi Capitals in 2018.
- We need to update Kings XI Punjab to Punjab kings, as the team is renamed as Punjab Kings in 2021.
- Consistent teams in 2021 are Kolkata Knight Riders, Royal Challengers Bangalore, Kings XI Punjab, Chennai Super Kings, Mumbai Indians, Rajasthan Royals, Sunrisers Hyderabad, Delhi Capitals. We will remove all the other teams as we don't need them for prediction.



The number of matches played during the years 2011, 2012, and 2013 were high compared to other years because new teams were added in those years. In the other years, the number of matches played was high and low due to tied matches. Due to super overs, there was a low and high number of matches played each year. In the year 2021, only half of the matches were held so it was lower than other years.

Remove extra innings

```
In [53]: 1 ipl_matches.innings.unique()
Out[53]: array([1, 2, 3, 4, 5, 6])
```

- We have 6 innings, this is because of super over matches when the match is tied. We have to remove extra innings except innings 1, as we need to predict the final score of the first innings.

Target variable [Total runs]

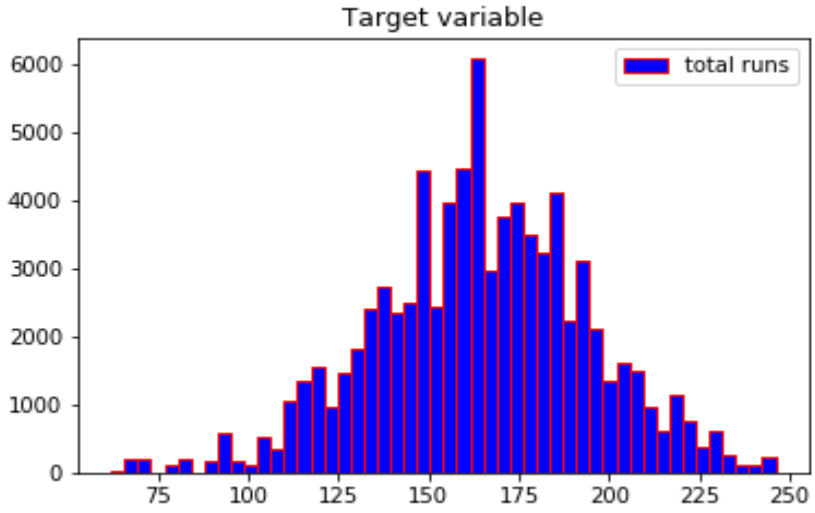
The 5 highest total scores recorded in the first innings by teams in Ipl are :

Batting team	Bowling team	Total runs
Chennai Super Kings	Rajasthan Royals	246
Kolkata Knight Riders	Punjab Kings	245
Chennai Super Kings	Punjab Kings	240
Royal Challengers Bangalore	Mumbai Indians	235
Kolkata Knight Riders	Mumbai Indians	232

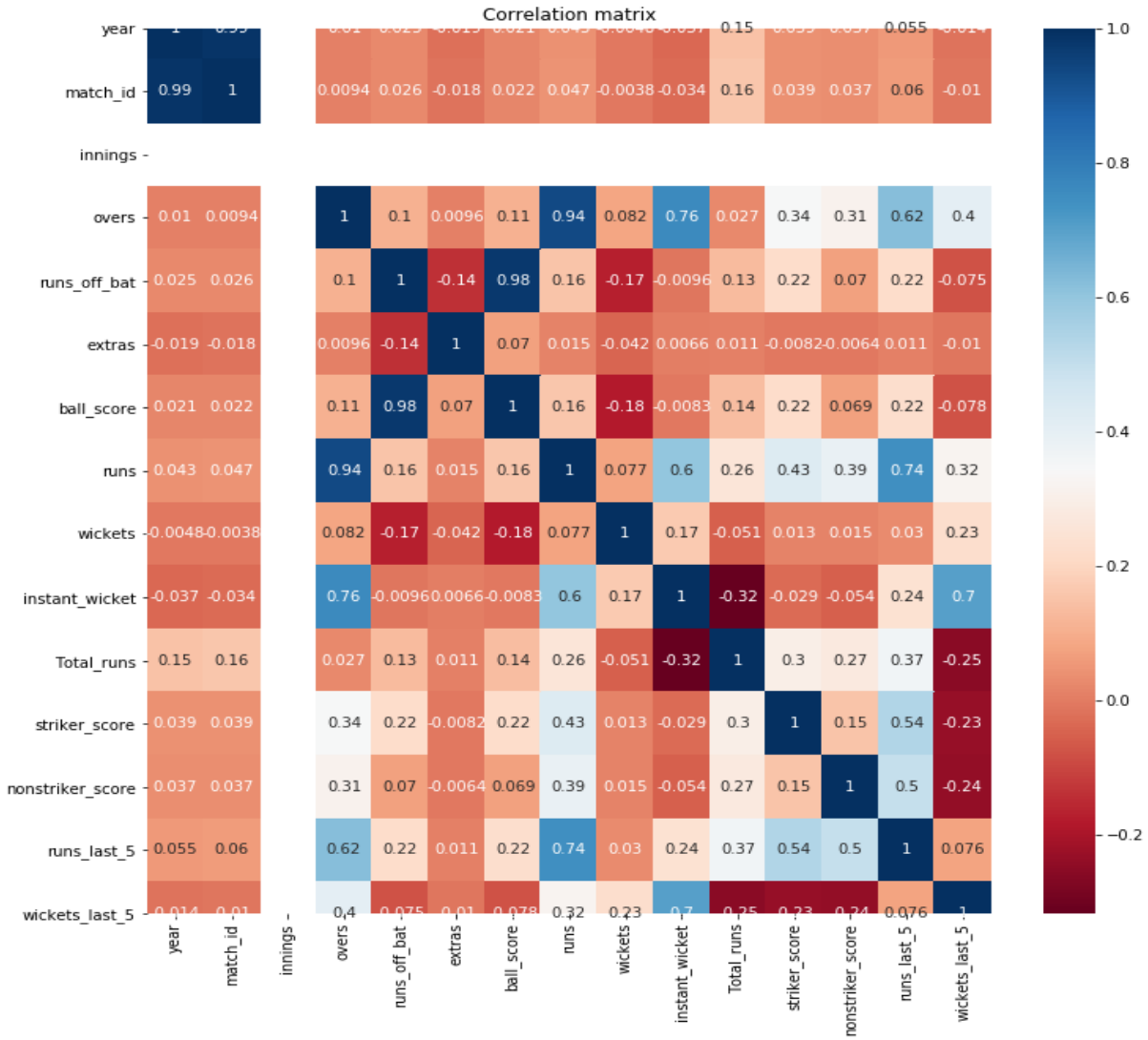
The 5 lowest total scores recorded in the first innings

Batting team	Bowling team	Total runs
Royal Challengers Bangalore	Rajasthan Royals	62
Delhi Capitals	Punjab Kings	67
Kolkata Knight Riders	Mumbai Indians	67
Royal Challengers Bangalore	Chennai Super Kings	70
Royal Challengers Bangalore	Rajasthan Royals	70

The highest score by a team in the first innings of IPL is Chennai Super Kings with 246 against Rajasthan Royals. Royal Challengers Bangalore had the lowest score of the first innings with 62 against Rajasthan Royals.
We will look at the distribution of our target variable Total runs scored in the first innings.



5. Correlation With the Target Variable



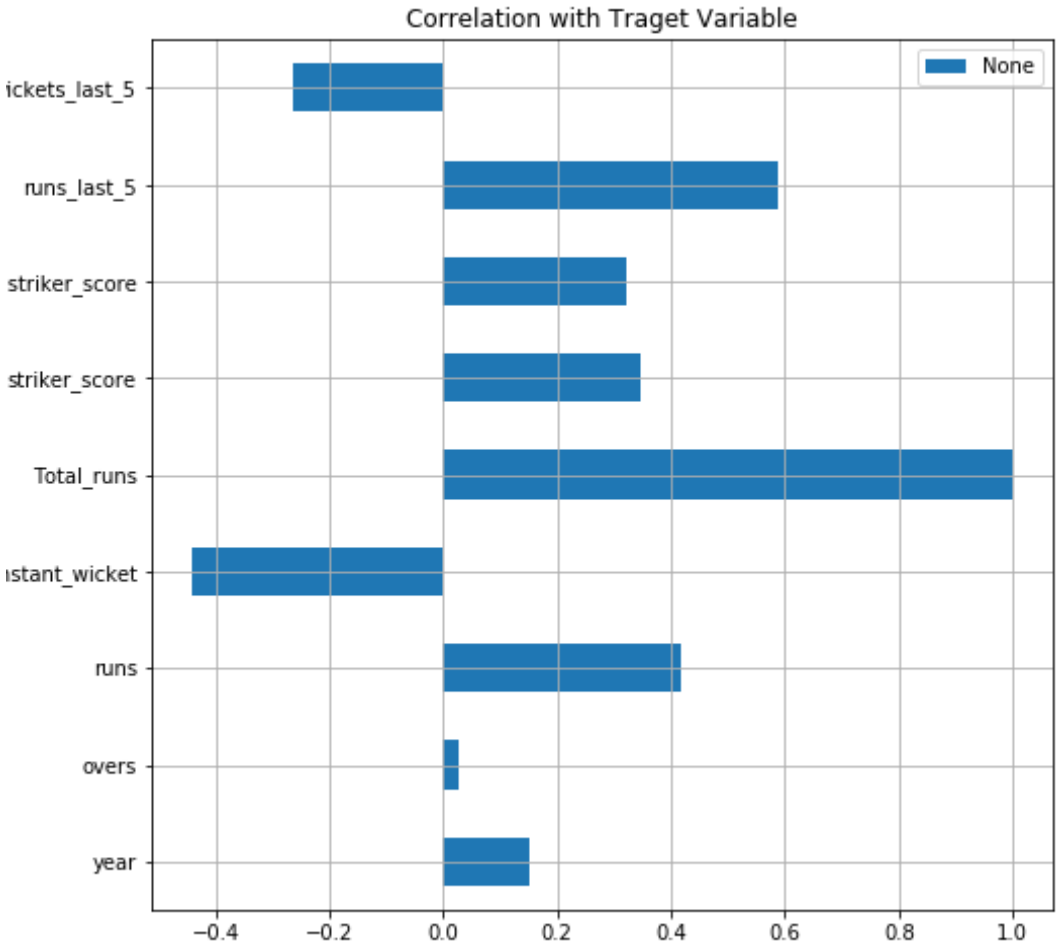
- We will remove the features which have a low correlation with Total runs [Target variable].

Features with low correlation are

'year','innings','extras','match_id','teams','runs_off_bat','ball_score'.

- We will remove the overs up to 6 as we are predicting the score after 6 overs.

Now we can observe the rise in the correlation with target variable



Correlation lies in the range of -1 and 1. A negative correlation implies the input and output move in opposite directions. As the input increases, the output decreases and vice versa. Zero correlation implies that the two variables are completely unrelated. Positive correlation implies that the input and output move in the same direction.

wickets_last_5: Wickets in the previous 5 overs have a negative correlation of -0.3 with total runs (Target variable). As the number of wickets in the previous five overs increases the total score of the innings decreases.

Runs_last_5: Runs scored in the previous 5 overs have a positive correlation of +0.6 with total runs (Target variable). As the runs scored in the previous five overs increase the total score of the innings increases.

Striker and Non_striker: Striker and nonstriker (players on two sides of the crease) have a positive correlation of +0.3 with total runs (Target variable). As the scores of the striker and nonstriker increase, the total score also increases.

Instant_wickets: Wickets at each instance have a negative correlation greater than -0.4 with total runs (Target variable). As the number of wickets at each instance increases the total score of the innings decreases.

Runs: Runs at each instance have a positive correlation of +0.4 with total runs (Target variable). As the runs at each instance increase the total score of the innings increases.

Independent variables (Input features): overs, batting team, Bowling team, striker, the non-striker, bowler, striker score, non-striker score, runs, wickets, runs in last 5 overs, wickets in last 5 overs.
Dependent variable (Output feature): Total runs of the innings
Other variables (which are not used): year, match id, runs off the bat, extras, ball score, wickets.

To use a particular model and predict the outcome, we need to convert all the categorical data into numerical data. Identify types of variables

- **Categorical variables:** Striker, Non-striker, Bowler, Batting team, Bowling team
- **Numerical variables**: overs, striker score, non-striker score, runs, wickets, runs in last five overs, wickets in last five overs.

6. Data Pre-processing

Most machine learning algorithms require data to be formatted in a very specific way. For this to take place, datasets generally require some amount of preparation before making useful insights. Missing and invalid values cause difficulty for an algorithm to process. Without preparation, the algorithm produces less accurate outcomes. Good data preparation produces clean and well-curated data which leads to more accurate model outcomes.

6.1 OneHot Encoding method for Batting team and Bowling team

For the variables batting team and bowling team, there is no relationship among the categories. So we will use one-hot encoding. There are eight consistent teams in 2021. They are Kolkata Knight Riders(KKR), Royal Challengers Bangalore(RCB), Chennai Super Kings(CSK), Kings XI Punjab(KXIP), Rajasthan Royals(RR), Mumbai Indians(MI), Sunrisers Hyderabad(SRH), Delhi Capitals(DC).

One Hot encoding for batting and bowling teams

KKR	1	0	0	0	0	0	0	0
RCB	0	1	0	0	0	0	0	0
CSK	0	0	1	0	0	0	0	0
KXIP	0	0	0	1	0	0	0	0
RR	0	0	0	0	1	0	0	0
MI	0	0	0	0	0	1	0	0
SRH	0	0	0	0	0	0	1	0
DC	0	0	0	0	0	0	0	1

6.2 Ordinal Encoding for batsman and Bowler ranks

In Ordinal Encoding, each unique category is assigned an integer value. By default, Ordinal encoding will assign integers to labels in the specified order. For this reason, I have arranged the ranks of bowlers and batsmen in descending order so that the player with the highest rank gets the first rank. In the same way, the least ranked player would be assigned with the highest rank.

Initially, I thought of ranking batsmen based on their strike rate. But, what if a player had faced only one ball and had hit a six? His strike rate will be 600, and it seems unfair to give ranks purely based on strike rate. So I have decided to rate the players based on all the statistics, based on ratings I ranked all the players.

My other encounter with a significant concern was about the retired players like Sachin and Sehwag who got better ranks than the current consistent players. To address this issue, I have decided to rate players only from the last four seasons i.e., 2016 to 2020 to achieve the consistency component of the players.

Batsman Ranks

Player	ratings	Ranks
KL Rahul	58.0	1
DA Warner	57.0	2
AB de Villiers	55.0	3
SV Samson	53.0	4
CH Gayle	53.0	5
...	...	...

Bowler Ranks

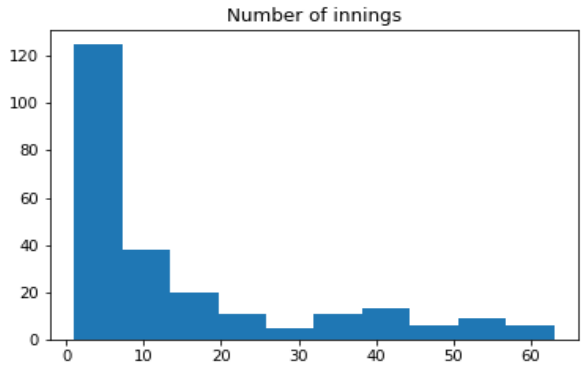
bowler	ratings	ranks
Rashid Khan	48.0	1
JJ Bumrah	48.0	2
YS Chahal	40.0	3
KH Pandya	39.0	4
Sandeep Sharma	38.0	5
...	...	...

6.2.1 Batsmen rankings

- Number of innings played
- Number of runs scored
- Number of fours
- Number of sixes
- Number centuries
- Number of fifties (Half-centuries)
- Highest score
- Batting average
- Batting strike rate

Ratings for the number of matches played by each player

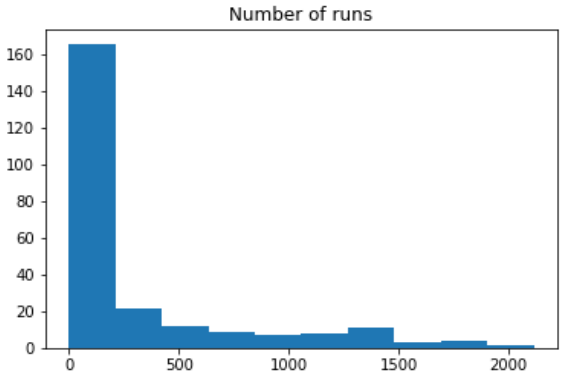
Number of innings



Ratings for number of innings played

- 0 - 6 : 1
- 7 - 12 : 2
- 13 - 18 : 3
- 18 - 24 : 4
- 25 - 30 : 5
- 31 - 36 : 6
- 37 - 42 : 7
- 43 - 48 : 8
- 49 - 54 : 9
- 55 > : 10

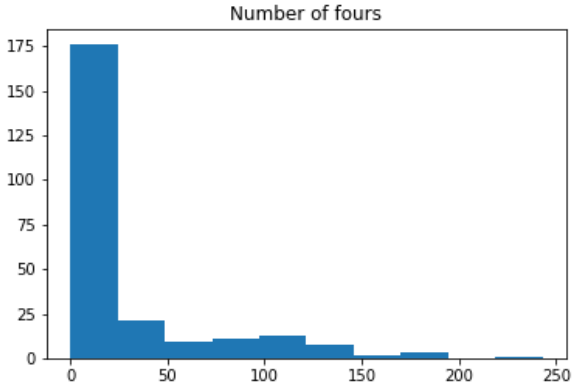
Number of runs



Ratings for number of runs scored

- 0 - 100 : 1
- 101 - 200 : 2
- 201 - 300 : 3
- 301 - 400 : 4
- 401 - 500 : 5
- 501 - 750 : 6
- 751 - 1000 : 7
- 1001 - 1500 : 8
- 1501 - 2000 : 9
- 2000 > : 10

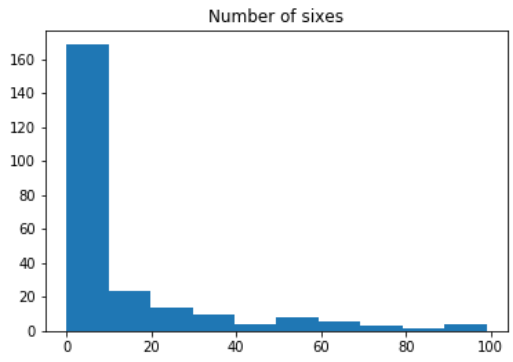
Number of fours



Ratings for number of fours

- 0 - 25 : 1
- 26 - 50 : 2
- 51 - 75 : 3
- 76 - 100 : 4
- 101 - 125 : 5
- 126 - 150 : 6
- 151 - 175 : 7
- 176 - 200 : 8
- 201 - 225 : 9
- 226 > : 10

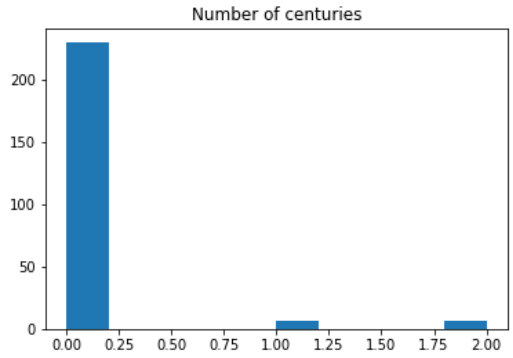
Number of sixes



Ratings for number of sixes

- 0 - 10 : 1
- 11 - 20 : 2
- 21 - 30 : 3
- 31 - 40 : 4
- 41 - 50 : 5
- 51 - 60 : 6
- 61 - 70 : 7
- 71 - 80 : 8
- 81 - 90 : 9
- 91 > : 10

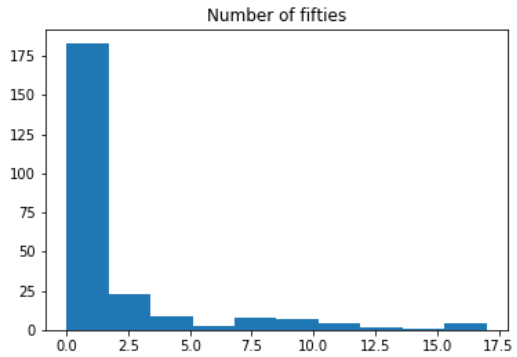
Number of centuries



Ratings for number of centuries

- 0: 0
- 1: 1
- 2: 2
- 3: 3

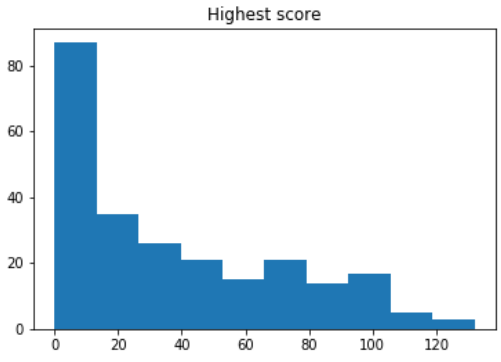
Number of the fifties



Ratings for the number of fifties

- 0 - 2 : 1
- 3 - 4 : 2
- 5 - 6 : 3
- 7 - 8 : 4
- 9 - 10 : 5
- 11 - 12 : 6
- 13 - 14 : 7
- 15 - 16 : 8
- 17 - 18 : 9
- 19 > : 10

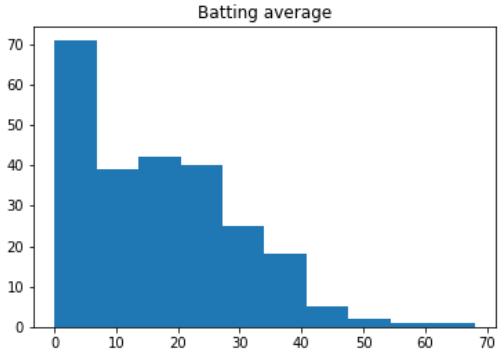
Highest score



Ratings for Highest score

- 0 - 12 : 1
- 13 - 24 : 2
- 25 - 36 : 3
- 37 - 48 : 4
- 49 - 60 : 5
- 61 - 72 : 6
- 73 - 84 : 7
- 85 - 96 : 8
- 97 - 108 : 9
- 108 > 10

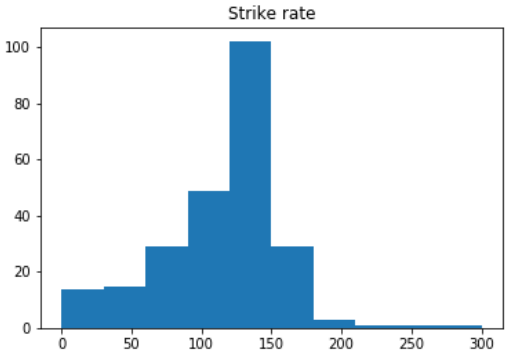
Batting average



Ratings for Batting average

- 0 - 7 : 1
- 8 - 14 : 2
- 15 - 21 : 3
- 22 - 28 : 4
- 29 - 35 : 5
- 36 - 42 : 6
- 43 - 49 : 7
- 50 - 56 : 8
- 57 - 63 : 9
- 64 > : 10

Strike rate



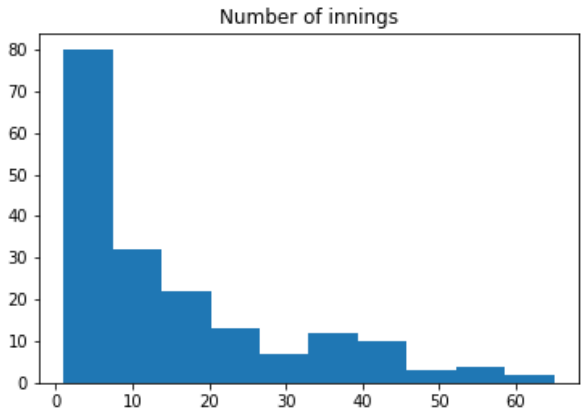
Ratings for Striker Rate

- 0 - 30 : 1
- 31 - 60 : 2
- 61 - 90 : 3
- 91 - 120 : 4
- 121 - 150 : 5
- 151 - 180 : 6
- 181 - 210 : 7
- 211 - 240 : 8
- 241 - 270 : 9
- 271 > : 10

6.1.2 Bowler Rankings

For the consistency of bowlers, I have rated for the last four seasons from 2017 to 2021

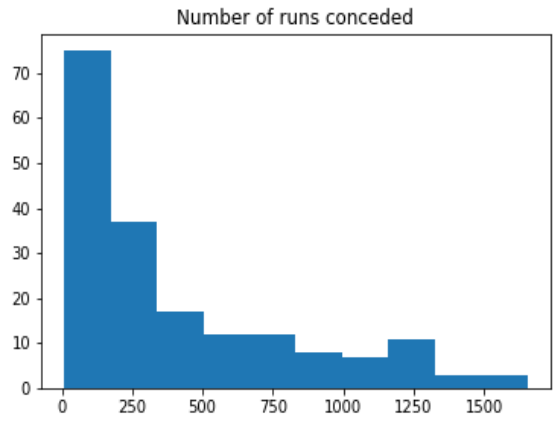
- Number of innings played
- Number of runs conceded
- Number of overs bowled
- Three wickets haul
- Five wickets haul
- Bowling economy



Ratings for number innings played

- 0 - 6 : 1
- 7 - 12 : 2
- 13 - 18 : 3
- 19 - 24 : 4
- 25 - 30 : 5
- 31 - 36 : 6
- 37 - 42 : 7
- 43 - 48 : 8
- 49 - 54 : 9
- 54 > : 10

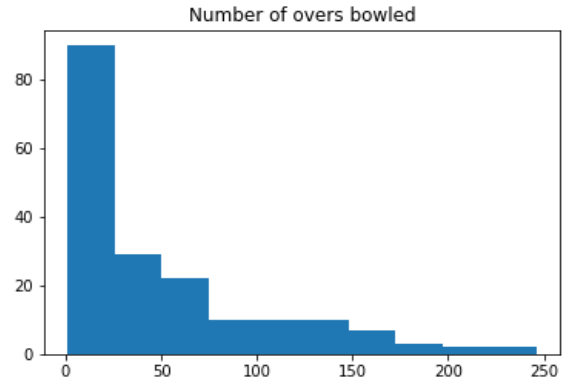
Number of runs conceded



Ratings for runs conceded

- 0 - 150 : 1
- 151 - 300 : 2
- 301 - 450 : 3
- 451 - 600 : 4
- 601 - 750 : 5
- 751 - 900 : 6
- 901 - 1050 : 7
- 1051 - 1200 : 8
- 1201 - 1350 : 9
- 1351 > : 10

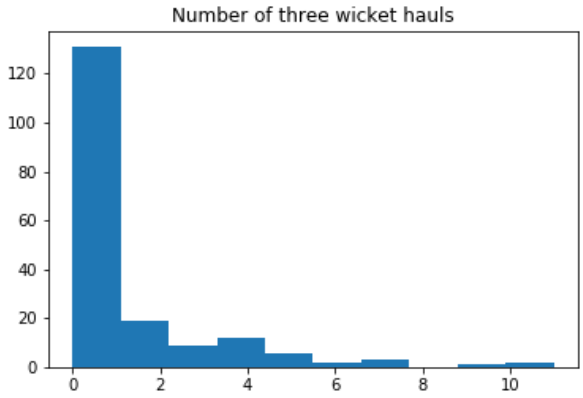
Number of overs bowled



Ratings for number of overs bowled

- 0 - 25 : 1
- 26 - 50 : 2
- 51 - 75 : 3
- 76 - 100 : 4
- 101 - 125 : 5
- 126 - 150 : 6
- 151 - 175 : 7
- 176 - 200 : 8
- 201 - 225 : 9
- 225 > : 10

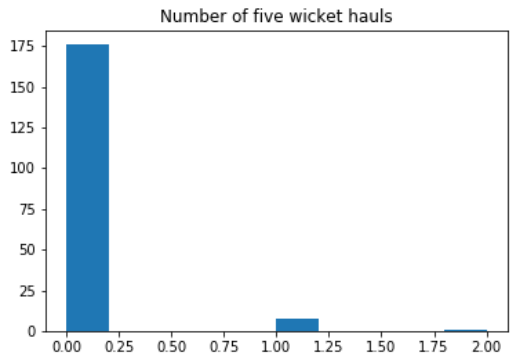
Number of three-wicket Hauls



Ratings for three-wicket haul

- 1: 1
- 2: 2
- 3 : 3
- 4: 4
- 5: 5
- 6: 6
- 7: 7
- 8: 8
- 9: 9
- 10 > : 10

Number of five-wicket hauls

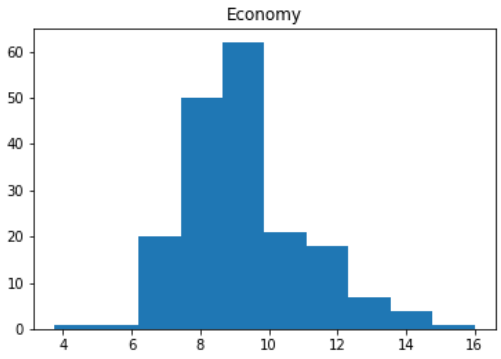


Ratings for five-wicket Hauls

- 0 : 0
- 1: 1
- 2: 2

Economy

A player's economy rate is the average number of runs they have conceded per a bowled over. The lower the economy rate is the better the bowler is performing.



Ratings for Economy

- < 5 : 10
- 6 : 9
- 7 : 8
- 8 : 7
- 9 : 6
- 10 : 5
- 11 : 4
- 12 : 3
- 13 : 2
- 14 > : 1

So now all the dependent and independent variables are numerical.

7. Splitting the data into training, testing, and validation sets

Create dependent and independent data sets based on our dependent and independent features. From 2008 to 2020 data, we will use 75% for training and 25% for testing. We will use 2021 data for model validation.

8. Model selection

R2-value: R-squared value is the measure of how much the change in output variable (total score) explained by the change in the input variable. R-square value is always between 0 and 1. 0 and that indicates the model explaining nil variability. 1 indicates the model explaining full variability. The higher the R-squared value the better the model fits the data. But the R-squared value does not always work and explain the goodness of the regression model. In multivariate linear regression, when we keep on adding new variables, the R square value will always increase irrespective of the variable significance. So, while doing the multivariate regression, we should look at the adjusted R-square value.

MAE: Mean Absolute Error is the average error in units of the predicted feature.
MSE: Mean Squared Error
RMSE: Root Mean Squared Error shows how far predictions fall from measured true values using Euclidean distance.

The best model in regression is defined as the model that has **the lowest prediction error**. The lower the RMSE the better the model. The higher the adjusted R-square, the better the model.

Linear Regression					
	MAE	MSE	RMSE	R2-value	Adjusted R2
Training data	12.80	294.41	17.15	0.666	0.666
Testing data	12.95	303.17	17.41	0.656	0.656
Validation	14.11	322.52	17.95	0.66	0.66

RandomForestRegressor					
	MAE	MSE	RMSE	R2-value	Adjusted R2
Training data	1.96	9.51	3.08	0.989	0.989
Testing data	5.26	68.68	8.28	0.922	0.922
Validation	14.85	389.48	19.73	0.598	0.594

Random Forest Regressor is overfitting the data. The Mean Absolute Error for training data is 1.96 and 5.26 for testing. It worked better for the training set, but it failed in working for testing. For validation (unknown data to the model, 2021 ball by ball data), Random Forest Regressor failed.

Lasso Regression					
	MAE	MSE	RMSE	R2-value	Adjusted R2
Training data	12.89	301.5	17.36	0.658	0.657
Testing data	13.05	310.40	17.61	0.648	0.647
Validation	13.92	308.61	17.56	0.682	0.678

Ridge Regression					
	MAE	MSE	RMSE	R2-value	Adjusted R2
Training data	12.80	294.41	17.15	0.66	0.66
Testing data	12.95	303.17	17.41	0.656	0.656
Validation	14.11	322.52	17.95	0.66	0.66

AdaBoostRegressor					
	MAE	MSE	RMSE	R2-value	Adjusted R2
Training data	16.40	428.22	20.69	0.514	0.514
Testing data	16.49	434.69	20.84	0.507	0.507
Validation	17.97	480.18	21.19	0.505	0.500

GradientBoostingRegressor					
	MAE	MSE	RMSE	R2-value	Adjusted R2
Training data	11.43	235.74	15.35	0.732	0.732
Testing data	11.76	251.54	15.86	0.715	0.714
Validation	13.76	324.63	18.01	0.66	0.66

XGBRegressor					
	MAE	MSE	RMSE	R2-value	Adjusted R2
Training data	12.32	273.05	16.52	0.690	0.690
Testing data	12.60	285.80	16.90	0.676	0.675
Validation	14.09	331.21	18.19	0.658	0.655

9. Training with Linear Regression model

Linear regression is a way of explaining the relationship between the dependent variable and one or more independent variables using a straight line.

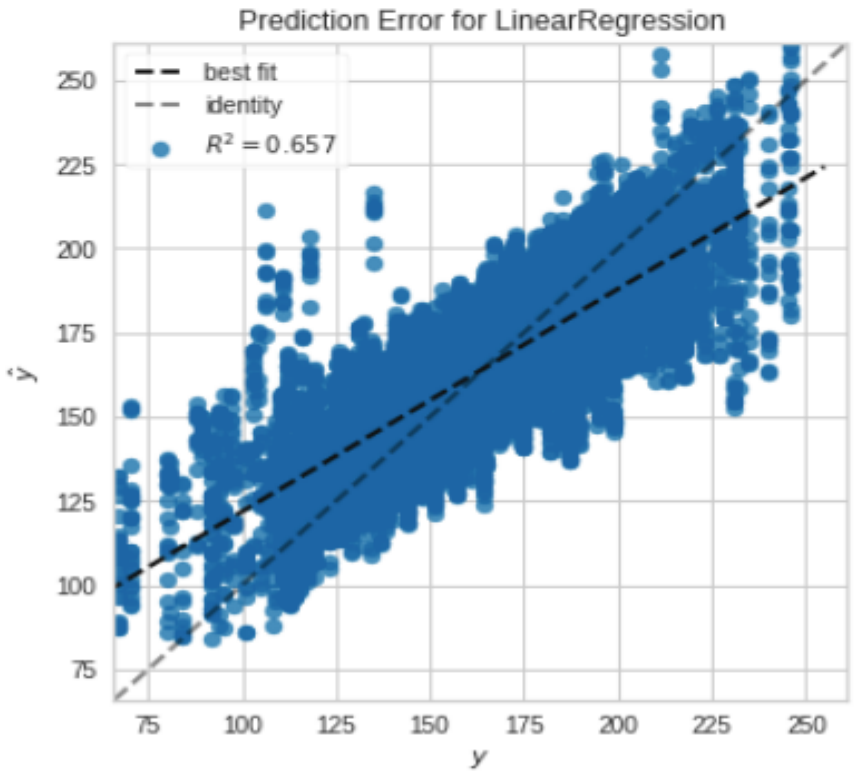
$$y = mx + c$$

y - predicted value

x - known value

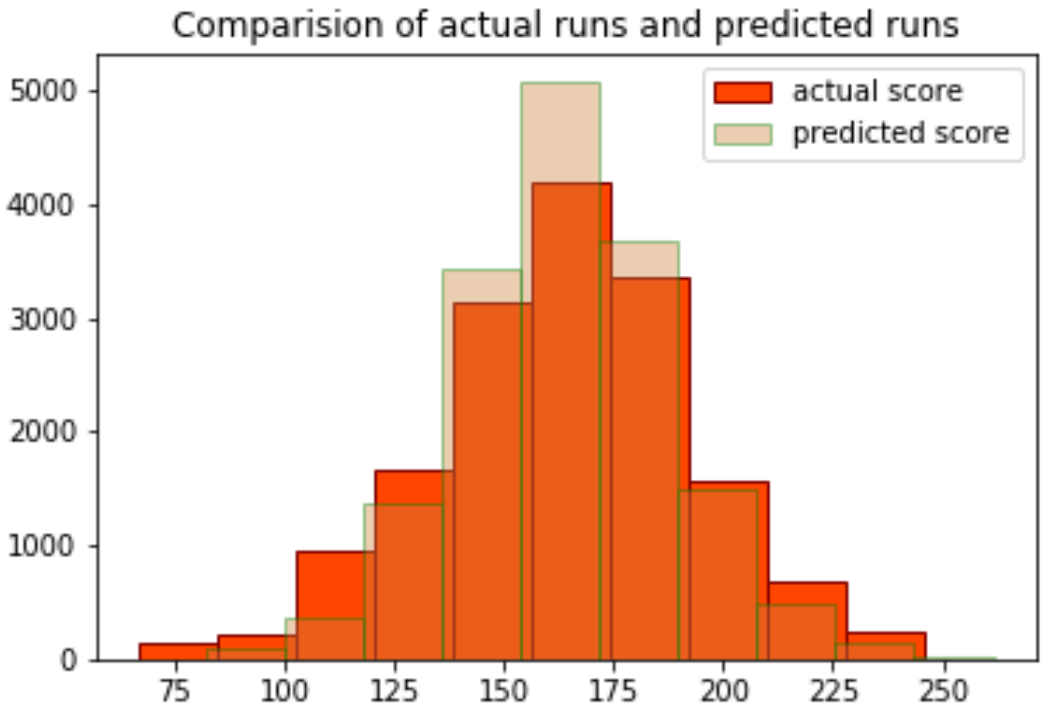
m - regression coefficient

c - constant/intercept



Comparing actual and predicted scores		
	Actual score	Predicted score
count	16155	16155
mean	163.97	164.07
std	29.71	24.09
min	67	81.93
25%	145	148.56
50%	164	164.08
75%	184	178.47
max	246	261.53

Graphical representation of a comparison of actual and predicted scores



Sometimes the predicted score is higher than the actual score and vice versa. Therefore, we cannot predict the future of games like cricket whose future could potentially go beyond our prediction in very little time.

10. Model validation

Residual (error): It is the distance between a predicted value and the actual value

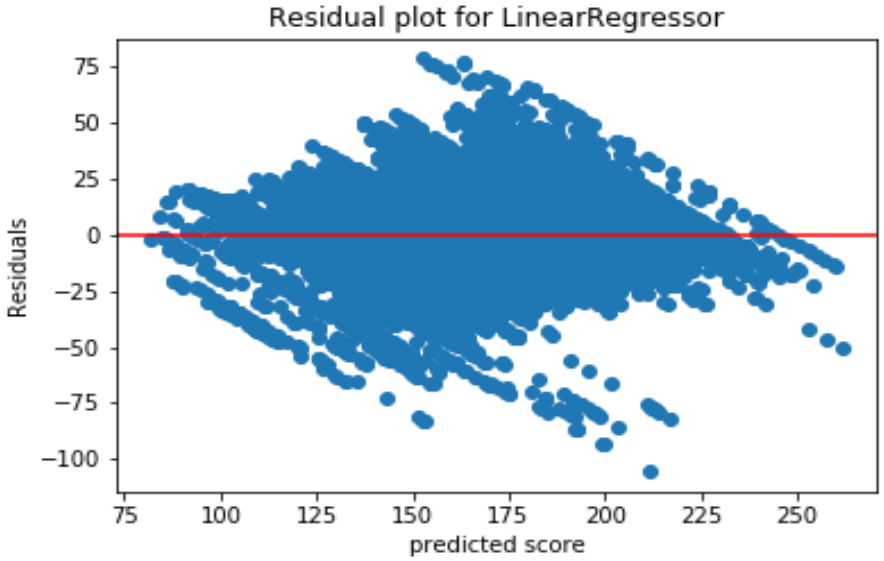
Residual (E) = actual - predicted

8.1 Summary statistics of residuals

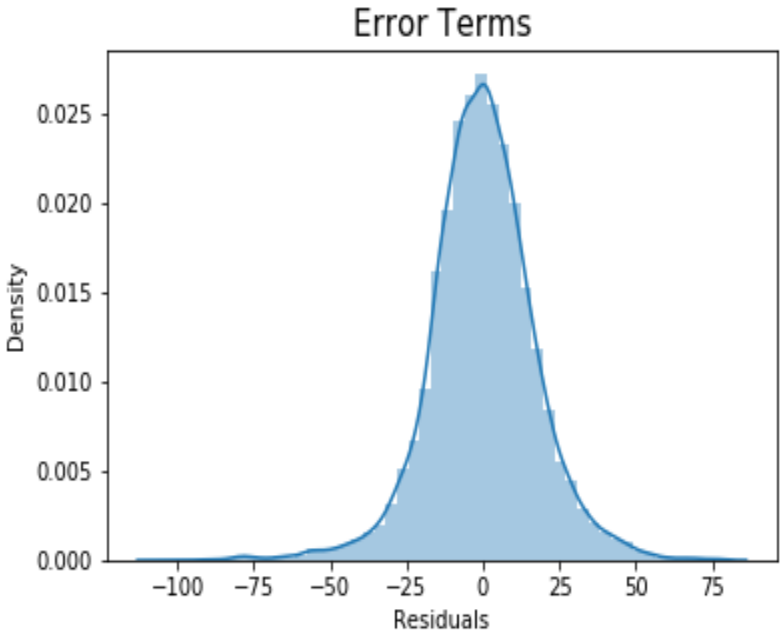
]:

	residuals	abs_residuals
count	16155.000000	16155.000000
mean	-0.098840	12.956709
std	17.412196	11.632180
min	-105.454344	0.000439
25%	-10.042244	4.770109
50%	-0.131449	10.008907
75%	9.974478	17.426310
max	78.534313	105.454344

8.2 Residual plot for Linear Regression model validation



Prediction made by the model is on the X-axis. The accuracy of the prediction is on the Y-axis. The distance from the line at o is how bad the prediction was for that value. Positive values for the residuals indicates that the prediction was too low. In the same way, the negative value indicates that the prediction was too high. Zero tells that the guess was exactly correct. It has a high density of points closer to the origin and a low density of points away from the origin. The variance of the residuals is constant with the response variable. The random pattern indicates that the regression model provides a decent fit to the data.



11. Prediction Interval

A prediction interval is a quantification of the uncertainty on the prediction. The prediction interval can be calculated prediction +- z * sigma.

Z is the number of standard deviations from Gaussian distribution. Sigma is the standard deviation of the prediction.

12. Testing with input

Prediction 1

year: 2021

match number: 1

batting team name: KKR

bowling team name: RR

overs: 9.3

who is on strike: SP Narine

who is on non_strike: RA Tripathi

bowler: JD Unadkat

score at that instant: 53

number of wickets at that instant: 2

striker score at that instant: 6

non-striker score at that instant: 12

score in last five overs: 32

number of wickets lost in last five overs: 2

Total score: 133

The predicted score is between 158.0 and 187.0

Prediction 2

year: 2021

match number: 2

batting team name: DC

bowling team name: SRH

overs: 8.4

who is on strike: PP Shaw

who is on non_strike: S Dhawan

bowler: Rashid Khan

score at that instant: 72

number of wickets at that instant: 0

striker score at that instant: 44

non-striker score at that instant: 26

score in last five overs: 34

number of wickets lost in last five overs: 0

Total score: 159

The predicted score is between 145.0 and 174.0

Prediction 3

year: 2021

match number : 3

batting team name: CSK

bowling team name: RCB

overs: 11.4

who is on strike: SK Raina

who is on non_strike: F du Plessis

bowler: Washington Sundar

score at that instant: 97

number of wickets at that instant: 1

striker score at that instant: 17

non-striker score at that instant: 43

score in last five overs: 43

number of wickets lost in last five overs: 1

Total score: 191

The predicted score is between 173.0 and 203.0

References:-

1. Cricsheet, https://cricsheet.org/downloads/. Last accessed on 26 May, 2021.
2. Delhi Capitals, Wikipedia, https://en.wikipedia.org/wiki/Delhi_Capitals. Last accessed on 4 June, 2021.
3. IPL2021: Teams And Squads List For Indian Premier League, Outlook Web Bureau,https://www.outlookindia.com/website/story/sports-news-ipl-2021-teams-and-squads-list-for-indian-premier-league/379563. Last updated on 7 April, 2021. Last accessed on 4 June 2021.
4. Strike Rate, Wikipedia, https://en.wikipedia.org/wiki/Strike_rate. Last accessed on 26 May, 2021.
5. Batting average (cricket), Wikipedia, https://en.wikipedia.org/wiki/Batting_average_(cricket). Last accessed on 27 May, 2021.
6. Economy rate, Wikipedia,https://en.wikipedia.org/wiki/Economy_rate. Last accessed on 27 May, 2021.
7. Punjab Kings, Wikipedia, https://en.wikipedia.org/wiki/Punjab_Kings. Last accessed 09 June, 2021.
8. Ordinal and One-Hot Encodings for Categorical Data,Jason Brownlee, https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/, published on 12 June, 2020, Last accessed on 28 May,2021.
9. How to use Residual Plots for regression model validation, Usman Gohar, https://towardsdatascience.com/how-to-use-residual-plots-for-regression-model-validation-c3c70e8ab378. 5 March,2020, Last accessed 29 May,2021.