# PREDICTION ALGORITHM FOR CRICKET USING MACHINE LEARNING

*A Graduate Project Report submitted to Manipal Academy of Higher Education in partial fulfillment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

*In*

**Mechatronics Engineering**

*Submitted by*

**Chinmay Datar** (160929052)
**Siatnshu Sah** (160907176)

*Under the guidance of*

**Vibha Damodara Kevala**
**Assistant Professor**
**Department of Mechatronics Engineering**
**MANIPAL INSTITUTE OF TECHNOLOGY**

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
*(A constituent unit of MAHE, Manipal)*

**June – 2020**

# PREDICTION ALGORITHM FOR CRICKET USING MACHINE LEARNING

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
*(A constituent unit of MAHE, Manipal)*

Manipal
31/05/2020

# CERTIFICATE

This is to certify that the project titled **PREDICTION ALGORITHM FOR CRICKET USING MACHINE LEARNING** is a record of the bonafide work done by **Chinmay Datar (160929052)**, submitted in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **Mechatronics** of Manipal Institute of Technology, Manipal, Karnataka (A constituent Institute of Manipal Academy of Higher Education) during the year 2019-2020.

**Vibha Damodara Kevala**

**Project Guide** **Head of the department**

# ACKNOWLEDGMENT

I have taken a lot of effort into this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

Firstly I would like to thank Dr. Chandrashekhar Bhat, HOD of the Mechatronics department, for providing me an opportunity to do the project work and giving me all support and guidance, which made me complete the project duly.

I owe my sincere gratitude to our project guide Mr. Shashi Kumar G S, department of ECE, who took a keen interest in our project work and guided us all along till the completion of our project work by providing all the necessary information for developing an excellent robust system.

I would not forget to remember Mr. Sitanshu Sah, my project partner, for his continuous help and support until the completion of our project work.

I heartily thank our internal project guide, Mrs. Vibha Damodara Kevala, for her guidance, suggestions, and continuous engagement during this project work.

I am thankful for and fortunate enough to get constant encouragement, support, and guidance from all Teaching staff of the department of Mechatronics, which helped us in successfully completing our project work. Also, I would like to extend our sincere appreciation to all staff in the laboratory for their timely support.

# ABSTRACT

The modern-day sport is being analyzed to gain a better understanding and help the players and the support staff to make better decisions. Techniques such as DLS and VJD provide a better experience for the audience in matches interrupted due to external circumstances.

Throughout the years there has been a shift in the style the game is played, with the game growing more complex, the existing models seem to fail at certain instances. An algorithm-based approach is required, where the scores after interruption would be calculated based on the history between the teams, of matches played on the ground, and in the condition would be accounted for. To ensure the algorithm designed doesn't overfit (generalize), proper precautions and understanding of the game are required.

# CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1. INTRODUCTION

## 1.1.INTRODUCTION

Predictive analytics is the study that involves many different statistical methods from machine learning, data mining, and predictive modeling that reads and evaluates the current and historical facts and data to make predictions or other unknown events which cannot be assessed otherwise.[1]

In business, predictive models find a pattern in historical and transactional data that can help find new opportunities and risks. Models check and capture relationships among many different attributes to assess the risk or potential opportunity linked with a specific set of conditions, guiding decision-making for future transactions. AI tools, like the ones used in self-driving cars, often are dependent on predictive algorithms for decision making to less of a risk on the road.

## 1.2.PRESENT DAY SCENARIO

Cricket is one of the most thrilling and fascinating games that the people of all age groups love to see and play. It is considered to be the most exciting and uncertain game. It is also a billion-dollar market as they speculate financially, hoping to be able to earn a profit.

In present-day cricket matches are supposed to be completed in one day, and there is not enough spare time when conditions to play are suitable to cover up the loss of more than a very few overs. If there is a delay to start, then the match is shortened. If the game is interrupted after it has begun, then the constraint is imposed on two resources. They have a maximum number of allotted overs. The score is reset using the DLS method [7].

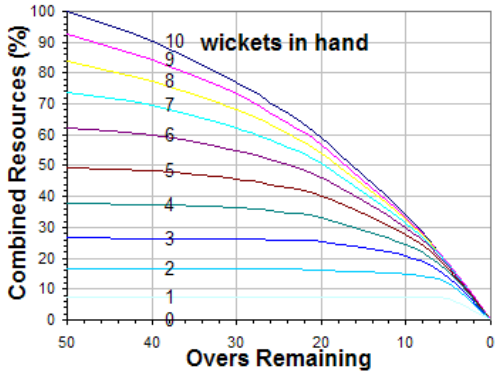



Fig. 1. Duckworth Lewis Method Graph

Table 1. Duckworth Lewis Resource Table

| OVERS LEFT | WICKETS LOST | | | | | | | | | | OVERS LEFT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| 50 | 100.0 | 93.4 | 85.1 | 74.9 | 62.7 | 49.0 | 34.9 | 22.0 | 11.9 | 4.7 | 50 |
| 49 | 99.1 | 92.6 | 84.5 | 74.4 | 62.5 | 48.9 | 34.9 | 22.0 | 11.9 | 4.7 | 49 |
| 48 | 98.1 | 91.7 | 83.8 | 74.0 | 62.2 | 48.8 | 34.9 | 22.0 | 11.9 | 4.7 | 48 |
| 47 | 97.1 | 90.9 | 83.2 | 73.5 | 61.9 | 48.6 | 34.9 | 22.0 | 11.9 | 4.7 | 47 |
| 46 | 96.1 | 90.0 | 82.5 | 73.0 | 61.6 | 48.5 | 34.8 | 22.0 | 11.9 | 4.7 | 46 |
| 45 | 95.0 | 89.1 | 81.8 | 72.5 | 61.3 | 48.4 | 34.8 | 22.0 | 11.9 | 4.7 | 45 |
| 44 | 93.9 | 88.2 | 81.0 | 72.0 | 61.0 | 48.3 | 34.8 | 22.0 | 11.9 | 4.7 | 44 |
| 43 | 92.8 | 87.3 | 80.3 | 71.4 | 60.7 | 48.1 | 34.7 | 22.0 | 11.9 | 4.7 | 43 |
| 42 | 91.7 | 86.3 | 79.5 | 70.9 | 60.3 | 47.9 | 34.7 | 22.0 | 11.9 | 4.7 | 42 |
| 41 | 90.5 | 85.3 | 78.7 | 70.3 | 59.9 | 47.8 | 34.6 | 22.0 | 11.9 | 4.7 | 41 |
| 40 | 89.3 | 84.2 | 77.8 | 69.6 | 59.5 | 47.6 | 34.6 | 22.0 | 11.9 | 4.7 | 40 |
| 39 | 88.0 | 83.1 | 76.9 | 69.0 | 59.1 | 47.4 | 34.5 | 22.0 | 11.9 | 4.7 | 39 |
| 38 | 86.7 | 82.0 | 76.0 | 68.3 | 58.7 | 47.1 | 34.5 | 21.9 | 11.9 | 4.7 | 38 |
| 37 | 85.4 | 80.9 | 75.0 | 67.6 | 58.2 | 46.9 | 34.4 | 21.9 | 11.9 | 4.7 | 37 |
| 36 | 84.1 | 79.7 | 74.1 | 66.8 | 57.7 | 46.6 | 34.3 | 21.9 | 11.9 | 4.7 | 36 |
| 35 | 82.7 | 78.5 | 73.0 | 66.0 | 57.2 | 46.4 | 34.2 | 21.9 | 11.9 | 4.7 | 35 |
| 34 | 81.3 | 77.2 | 72.0 | 65.2 | 56.6 | 46.1 | 34.1 | 21.9 | 11.9 | 4.7 | 34 |
| 33 | 79.8 | 75.9 | 70.9 | 64.4 | 56.0 | 45.8 | 34.0 | 21.9 | 11.9 | 4.7 | 33 |
| 32 | 78.3 | 74.6 | 69.7 | 63.5 | 55.4 | 45.4 | 33.9 | 21.9 | 11.9 | 4.7 | 32 |
| 31 | 76.7 | 73.2 | 68.6 | 62.5 | 54.8 | 45.1 | 33.7 | 21.9 | 11.9 | 4.7 | 31 |
| 30 | 75.1 | 71.8 | 67.3 | 61.6 | 54.1 | 44.7 | 33.6 | 21.8 | 11.9 | 4.7 | 30 |
| 29 | 73.5 | 70.3 | 66.1 | 60.5 | 53.4 | 44.2 | 33.4 | 21.8 | 11.9 | 4.7 | 29 |
| 28 | 71.8 | 68.8 | 64.8 | 59.5 | 52.6 | 43.8 | 33.2 | 21.8 | 11.9 | 4.7 | 28 |
| 27 | 70.1 | 67.2 | 63.4 | 58.4 | 51.8 | 43.3 | 33.0 | 21.7 | 11.9 | 4.7 | 27 |
| 26 | 68.3 | 65.6 | 62.0 | 57.2 | 50.9 | 42.8 | 32.8 | 21.7 | 11.9 | 4.7 | 26 |
| 25 | 66.5 | 63.9 | 60.5 | 56.0 | 50.0 | 42.2 | 32.6 | 21.6 | 11.9 | 4.7 | 25 |
| 24 | 64.6 | 62.2 | 59.0 | 54.7 | 49.0 | 41.6 | 32.3 | 21.6 | 11.9 | 4.7 | 24 |
| 23 | 62.7 | 60.4 | 57.4 | 53.4 | 48.0 | 40.9 | 32.0 | 21.5 | 11.9 | 4.7 | 23 |
| 22 | 60.7 | 58.6 | 55.8 | 52.0 | 47.0 | 40.2 | 31.6 | 21.4 | 11.9 | 4.7 | 22 |
| 21 | 58.7 | 56.7 | 54.1 | 50.6 | 45.8 | 39.4 | 31.2 | 21.3 | 11.9 | 4.7 | 21 |
| 20 | 56.6 | 54.8 | 52.4 | 49.1 | 44.6 | 38.6 | 30.8 | 21.2 | 11.9 | 4.7 | 20 |
| 19 | 54.4 | 52.8 | 50.5 | 47.5 | 43.4 | 37.7 | 30.3 | 21.1 | 11.9 | 4.7 | 19 |
| 18 | 52.2 | 50.7 | 48.6 | 45.9 | 42.0 | 36.8 | 29.8 | 20.9 | 11.9 | 4.7 | 18 |
| 17 | 49.9 | 48.5 | 46.7 | 44.1 | 40.6 | 35.8 | 29.2 | 20.7 | 11.9 | 4.7 | 17 |
| 16 | 47.6 | 46.3 | 44.7 | 42.3 | 39.1 | 34.7 | 28.5 | 20.5 | 11.8 | 4.7 | 16 |
| 15 | 45.2 | 44.1 | 42.6 | 40.5 | 37.6 | 33.5 | 27.8 | 20.2 | 11.8 | 4.7 | 15 |
| 14 | 42.7 | 41.7 | 40.4 | 38.5 | 35.9 | 32.2 | 27.0 | 19.9 | 11.8 | 4.7 | 14 |
| 13 | 40.2 | 39.3 | 38.1 | 36.5 | 34.2 | 30.8 | 26.1 | 19.5 | 11.7 | 4.7 | 13 |
| 12 | 37.6 | 36.8 | 35.8 | 34.3 | 32.3 | 29.4 | 25.1 | 19.0 | 11.6 | 4.7 | 12 |
| 11 | 34.9 | 34.2 | 33.4 | 32.1 | 30.4 | 27.8 | 24.0 | 18.5 | 11.5 | 4.7 | 11 |
| 10 | 32.1 | 31.6 | 30.8 | 29.8 | 28.3 | 26.1 | 22.8 | 17.9 | 11.4 | 4.7 | 10 |
| 9 | 29.3 | 28.9 | 28.2 | 27.4 | 26.1 | 24.2 | 21.4 | 17.1 | 11.2 | 4.7 | 9 |
| 8 | 26.4 | 26.0 | 25.5 | 24.8 | 23.8 | 22.3 | 19.9 | 16.2 | 10.9 | 4.7 | 8 |
| 7 | 23.4 | 23.1 | 22.7 | 22.2 | 21.4 | 20.1 | 18.2 | 15.2 | 10.5 | 4.7 | 7 |
| 6 | 20.3 | 20.1 | 19.8 | 19.4 | 18.8 | 17.8 | 16.4 | 13.9 | 10.1 | 4.6 | 6 |
| 5 | 17.2 | 17.0 | 16.8 | 16.5 | 16.1 | 15.4 | 14.3 | 12.5 | 9.4 | 4.6 | 5 |
| 4 | 13.9 | 13.8 | 13.7 | 13.5 | 13.2 | 12.7 | 12.0 | 10.7 | 8.4 | 4.5 | 4 |
| 3 | 10.6 | 10.5 | 10.4 | 10.3 | 10.2 | 9.9 | 9.5 | 8.7 | 7.2 | 4.2 | 3 |
| 2 | 7.2 | 7.1 | 7.1 | 7.0 | 7.0 | 6.8 | 6.6 | 6.2 | 5.5 | 3.7 | 2 |
| 1 | 3.6 | 3.6 | 3.6 | 3.6 | 3.6 | 3.5 | 3.5 | 3.4 | 3.2 | 2.5 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## 1.3. IMPORTANCE IN PRESENT DAY SCENARIO

The Duckworth–Lewis–Stern method (DLS) is a mathematical formula designed in 1997 to revise the score for the team batting in the second inning in a rain-interrupted cricket match. When a game is shortened due to unfavorable weather interruption and overs are lost, setting a revised

target score for the team batting in the second innings is not straightforward because reducing the runs target proportionally to the loss in overs does not give us the acceptable results. It is because a team with all ten wickets remaining and 25 overs left to bat can score at a faster rate than if they had ten wickets and a full 50 overs. The main issue with the method is that it factors in only the percentage of resources remaining and not the quality. Also, there have been cases in the past where the usage of the DL method led to a target that was impossible to achieve, whereas, in the situation at which the play had been stopped, all three results were possible.

Table 1 is used to calculate the revised target. The formula for the calculation is given in [7]. Fig.1 is a graphical representation of the table.

We aimed to develop an algorithm to analyze better and predict scores for a match in cricket after factoring in more variables and comparing their performance with the techniques which are currently being used. This is because the current methods are based on the number of individual factors and not the quality of those factors remaining, which sometimes leads to an unfair change in the revised scores in the match.

### 1.4. SHORTCOMINGS OF THE PREVIOUS METHODS

All the research done before is mainly on predicting the final scores and outcome of the match by observing the playing pattern of the players and teams. But we found very few researches on the score prediction for interrupted matches. Since the present methods take very few variables into account, we aimed to device an approach that considers more parameters while predicting the score and is closer to the real-world scenario.

The new proposed model aims to predict the score more accurately and closer to what it may have been in case the match wasn't interrupted. This will help make the game fairer. It will also help in improving the viewer experience of the game.

As stated by many experts, current methods used to reset the score of rain-interrupted match (Duckworth Lewis Method) only takes into account the percentage of resources remaining (number of overs remaining and number of wickets down) and no other variable is considered. Hence, **we aim to develop an algorithm to analyze better and predict scores for a match that is shortened due to rain in cricket after factoring in more variables**.

# CHAPTER 2. LITERATURE REVIEW

Geddam Jaishankar Harshit [2] has researched on analyzing match outcomes using machine learning techniques such as supervised learning algorithms to predict the outcome of ODI cricket matches. The user just needs to select the two teams to predict the winning team. The four primary attributes used were the name of both the teams, home, and opponent team, toss results, match results.

T. Prabhakar Reddy [3] has come up with a simple formula for resetting the score for interrupted T20 matches. The formula is based on the existing rain rules. It scales the resources left with the teams at the time of interruption to 50 over format. This method can also be applied to several other types of interrupted matches in which interruption may occur in the first innings or the second innings of a game.

Table 2. Duckworth Lewis Resource Table scaled for twenty-20 matches

| | Wickets Lost | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Overs Available | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 20 | 100.0 | 96.8 | 92.6 | 86.7 | 78.8 | 68.2 | 54.4 | 37.5 | 21.3 | 8.3 |
| 19 | 96.1 | 93.3 | 89.2 | 83.9 | 76.7 | 66.6 | 53.5 | 37.3 | 21.0 | 8.3 |
| 18 | 92.2 | 89.6 | 85.9 | 81.1 | 74.2 | 65.0 | 52.7 | 36.9 | 21.0 | 8.3 |
| 17 | 88.2 | 85.7 | 82.5 | 77.9 | 71.7 | 63.3 | 51.6 | 36.6 | 21.0 | 8.3 |
| 16 | 84.1 | 81.8 | 79.0 | 74.7 | 69.1 | 61.3 | 50.4 | 36.2 | 20.8 | 8.3 |
| 15 | 79.9 | 77.9 | 75.3 | 71.6 | 66.4 | 59.2 | 49.1 | 35.7 | 20.8 | 8.3 |
| 14 | 75.4 | 73.7 | 71.4 | 68.0 | 63.4 | 56.9 | 47.7 | 35.2 | 20.8 | 8.3 |
| 13 | 71.0 | 69.4 | 67.3 | 64.5 | 60.4 | 54.4 | 46.1 | 34.5 | 20.7 | 8.3 |
| 12 | 66.4 | 65.0 | 63.3 | 60.6 | 57.1 | 51.9 | 44.3 | 33.6 | 20.5 | 8.3 |
| 11 | 61.7 | 60.4 | 59.0 | 56.7 | 53.7 | 49.1 | 42.4 | 32.7 | 20.3 | 8.3 |
| 10 | 56.7 | 55.8 | 54.4 | 52.7 | 50.0 | 46.1 | 40.3 | 31.6 | 20.1 | 8.3 |
| 9 | 51.8 | 51.1 | 49.8 | 48.4 | 46.1 | 42.8 | 37.8 | 30.2 | 19.8 | 8.3 |
| 8 | 46.6 | 45.9 | 45.1 | 43.8 | 42.0 | 39.4 | 35.2 | 28.6 | 19.3 | 8.3 |
| 7 | 41.3 | 40.8 | 40.1 | 39.2 | 37.8 | 35.5 | 32.2 | 26.9 | 18.6 | 8.3 |
| 6 | 35.9 | 35.5 | 35.0 | 34.3 | 33.2 | 31.4 | 29.0 | 24.6 | 17.8 | 8.1 |
| 5 | 30.4 | 30.0 | 29.7 | 29.2 | 28.4 | 27.2 | 25.3 | 22.1 | 16.6 | 8.1 |
| 4 | 24.6 | 24.4 | 24.2 | 23.9 | 23.3 | 22.4 | 21.2 | 18.9 | 14.8 | 8.0 |
| 3 | 18.7 | 18.6 | 18.4 | 18.2 | 18.0 | 17.5 | 16.8 | 15.4 | 12.7 | 7.4 |
| 2 | 12.7 | 12.5 | 12.5 | 12.4 | 12.4 | 12.0 | 11.7 | 11.0 | 9.7 | 6.5 |
| 1 | 6.4 | 6.4 | 6.4 | 6.4 | 6.4 | 6.2 | 6.2 | 6.0 | 5.7 | 4.4 |

Kalpdrum Passi [4] has researched on predicting the players' performances in one-day matches by studying the player statistics using supervised machine learning techniques. The performance of

every player is analyzed as the number of runs scored by a batsman and the number of wickets taken by a bowler in a match. The attributes considered for this are batting hand, bowling hand, batting position, match type, match time. Classification algorithms used for this purpose were Naive Bayes theorem, decision tree, random forest, and support vector machine. The most accurate classifier for both datasets turned out to be Random Forest.

Madan Gopal Jhanwar and Vikram Pudi [5] have estimated the potentials of the 22 players (playing 11 of each team) playing using their career stats and current form determined by their performance in recent matches. This focuses on figuring out which team has relative dominance over the other. It takes into account other base features like toss result and the venue where the match is taking place, which, along with other career statics of players, helps in determining the winner of the match.

Ayush Kalla [6] has used a linear regression algorithm for predictive analysis. It aims to determine the individual performances of the players, which in turn will determine the approximate runs that a team might score in the match. The graph is plotted of previous data, and then the best-fit algorithm is implemented to predict the slope and intercept of a line that is closest to most of the points. This helps in predicting the future using the available variable in the equation of the line.

Aminul Islam Anik's paper [8] presents an analytical method that aims to predict a cricket player's performance in an upcoming match with the help of ML algorithms. To use the proposed model for predictions, the statistical data was processed into numerical values so it can be applied in the algorithms. It also helps in predicting runs scored by batsman and wickets taken by a bowler in a match. Therefore, this helps in predicting a player's future performance and thus ensures a better selection of the team for cricket matches played in the future.

V. Jayadevan [10] proposed a new method based on a mathematical model that refers to the natural flow of the innings. It is based on the concept of normal scores and target scores. Regression equations obtained from the analysis of the dataset of closely fought matches are used to construct an easy-to-use table for employing the method in the field. The method can handle multiple interruptions. In cases where the D/L method leads to too inappropriate targets, this method gives satisfactory results. The VJD method is currently being used by BCCI since 2010. It is considered superior to the D/L method.


Fig. 2. Normal curve and target curve

# CHAPTER 3. THEORETICAL BACKGROUND

Some other methods that were used before the DLS method are:

1. Average run Rate (ARR)

The team which has scored runs at a higher run rate for the duration of the game is declared as the winner. It is a straightforward calculation, but the drawback of this method is that it usually favors the chasing team.

2. Most Productive Overs (MPO)

The target is obtained for the remaining overs of the chasing team by totaling the equal number of the highest-scoring overs of Team 1. The process of determining the target involves extensive record-keeping and calculations for match officials, and the pattern of scoring for Team 1 is a criterion to decide the winner. The method tends to favor Team 1. One of the most noteworthy examples came in the 1992 Cricket World Cup, where the MPO method was used; in the semi-final between England and South Africa, the rain stopped play for less than 15 minutes. South Africa required 22 runs off 13 balls. England's score was 252/6 in 45 overs. After the target was revised for South Africa, they needed 21 runs off 1 ball. This was a reduction of only one run and 2 overs. With the D/L method, this kind of error is avoided. If the D/L method were applied in this situation, the target would have been 5 runs off 1 ball.

3. Discounted Most Productive overs (DMPO)

The total runs from the overs in which most runs were scored are reduced by 0.5% for each lost over. This method reduces the advantage that the MPO method gives to Team 1 by a little margin. But DMPO still has the same fundamental flaw of the MPO method.

4. Parabola (PARAB)

This method was introduced by a young South African (do Rego), which calculates a table of norms y, for the overs of an inning, s, using the parabola with the equation "$y = 7.46x + 0.059x^2$" to model, appropriately. The parabola has a turning point at a little over 60 overs (nearly 63 overs), the reducing return's nature of the relationship between average total runs and the overs to be bowled. This method is a significant improvement over ARR, but it does not take into account the number of overs that are lost or of the number of wickets that are fallen.

5. Clark Curves (CLARK)

Clark Curves method, as explained online, tries to rectify the drawbacks of the previous method. It talks about six types of interruptions in a game, three for each innings, and interruptions occurring before the innings start. It applies a different set of rules for a different type of interruption. Some of the rules allow for wickets that have already fallen. There are breaks between the resetted target at the two adjacent types of interruptions.

Table 3. Calculation of revised target score
in hypothetical 50 over examples

| *Hypothetical example no.* | *I* | *II* | *III* | *IV* | *V* | *VI* |
|---|---|---|---|---|---|---|
| Team 2 score, chasing 250(=$S$), $R_1 = 1$ | 0 | 75 | 120 | 75 | 191 | 180 |
| Wickets lost, $w$ | 0 | 0 | 0 | 2 | 9 | 4 |
| Overs left at the stoppage, $u_1$ | 50 | 30 | 20 | 30 | 20 | 20 |
| Overs left at the stoppage, $u_2$ | 30 | 10 | 0 | 10 | 0 | 0 |
| Proportion of resources left at resumption $P(u_1, w)$ | 1 | 0.771 | 0.589 | 0.682 | 0.076 | 0.461 |
| Proportion of resources left at resumption $P(u_2, w)$ | 0.771 | 0.341 | 0 | 0.325 | 0 | 0 |
| Proportion lost in $(u_1 - u_2)$ overs $P(u_1, w) - P(u_2, w)$ | 0.229 | 0.430 | 0.589 | 0.357 | 0.076 | 0.461 |
| Proportion available $R_2 = 1 - P(u_1, w) + P(u_2, w)$ | 0.771 | 0.570 | 0.411 | 0.643 | 0.924 | 0.539 |
| Revised target score $T = SR_2$ | 192.8 | 142.5 | 102.8 | 160.8 | 231.0 | 134.8 |
| D/L target to win | **193** | **143** | **103** | **161** | **232** | **135** |
| OTarget to win from other methods: | | | | | | |
| ARR | 151 | **151** | 151 | **151** | 151 | 151 |
| WC96 | **191** | 191 | 191 | 191 | 191 | 191 |
| MPO[a] | **201** | 201 | 201 | 201 | 201 | 201 |
| DMPO[a] | **181** | 181 | 181 | 181 | 181 | 181 |
| CLARK | **182** | 162 | 134 | **162** | 201 | **134** |

[a] The targets by the MPO and DMPO methods cannot be evaluated properly without the actual score cards to find the total of the 30 most productive overs. To obtain some comparative figures we have assumed here that the 20 *least* productive overs yielded 50 runs, half the average run rate. Therefore, the 30 most productive overs yielded 200 runs.

After 1997, the DLS method is being used for all interrupted matches.

The DLS method [12], as we know, considers two essential factors of a match while resetting or revising the target for the team batting second in case the match is interrupted due to any circumstances. The factors are, i) the number of overs remaining and ii) the number of wickets lost.

The DLS table consists of these two factors. The number of overs remaining reduces from top to bottom (50 -> 0). The number of wickets lost increases from left to right (0 -> 9).

This forms a matrix, and each (x,y) cell gives a percentage value of resources remaining at that instance. In terms of the factors, x is the number of overs remaining, and y is the number of wickets lost.

*Case 1 (Match Interrupted in the 1st innings)*

In case the match is stopped during the first innings, and it results in the loss of overs, the second team (Team B) would have a clear advantage over the first team, let's say Team A. To ensure that it is fair for both the teams X runs are added to Team A's final score.

X is calculated as follows:

Let us assume that the revised target = A

225 is taken as the Global average for DLS calculations.

Let us say that when Team A's innings was interrupted, they had used up Y% of resources according to the DLS table.

After the rain has stopped, the umpires decide to reduce a certain number of overs from the total quota of 50 overs a side, depending upon the time lost.

Based on the reduced overs, the resources available for Team B are calculated from the DLS table as Z%.

The difference between Z and Y, i.e., (Z-Y) % is calculated. (Z-Y) % of 225 gives us the value of X. The final target for Team B is then calculated as the sum of Team A's final score and additional value X incremented by 1, i.e., Revised Target = Team A's score + X + 1.

*Case 2 (Match Interrupted in the 2nd innings)*

In case the match is stopped during the second innings. It results in loss of overs. The target for Team B should be reduced as they would be left with lesser resources than Team A. If M and N are the percentage resources available to Team B and Team A respectively, then the ratio of M:N multiplied by the original target gives us the revised target.

In case the first team plays its entire quota of 50 overs, it is said to have used 100% of its resources, i.e., N=100%.

The second innings is interrupted due to rain. At that instance, Team B would have utilized a certain amount of resources.

The remaining resources at this point are taken from the DLS table as Z%. Once the rain stops and match starts, there are chances of there being a reduced game. Based on the reduced conditions, again, the remaining resources with Team B is calculated from the table as W%.

The (Z-W) % is calculated. M, which is the final resources left with Team B is calculated as M = 100-(Z-W).

Revised Target is then calculated using the formula, (M/N) *(Team A's Score).

*Case 3 (Match abandoned in the middle of the 2nd Innings)*

In case the first team plays its entire quota of 50 overs, it is said to have used 100% of its resources, i.e., N=100%.

The second innings is interrupted due to rain. At that instance, Team B would have utilized a certain amount of resources. The remaining resources at this point are taken from the DLS table as Z%.

This means that Team B has effectively utilized (100-Z) % of the resources [Let's say M].

Due to some unavoidable circumstances, the match has to be called off at that instance. To determine the winning team, we calculate a Par Score for Team B at the point when the match had been stopped.

This Par Score is determined using the formula, (M/N) *(Team A's Score).

# CHAPTER 4. OBJECTIVES & METHODOLOGY

## 4.1. OBJECTIVES

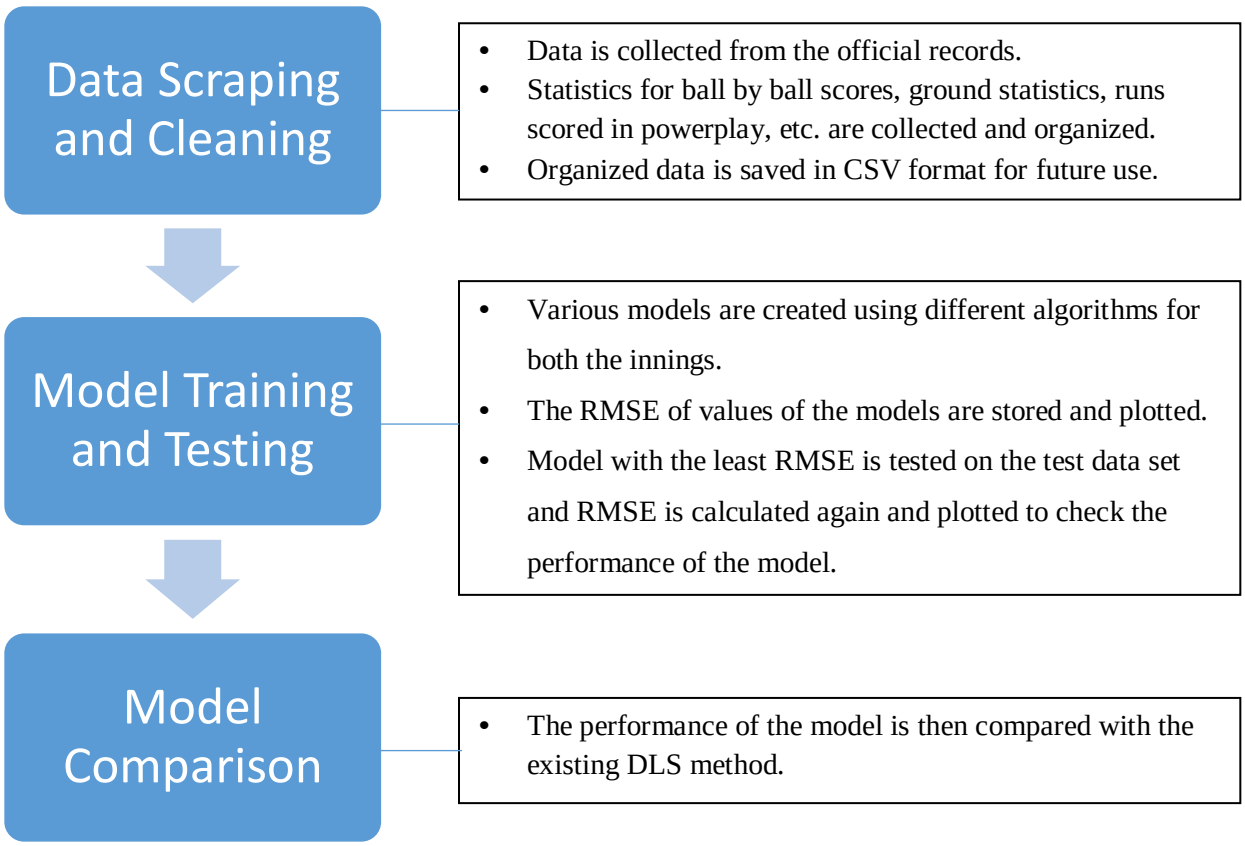
- To scrape the data set off espncricinfo.com, clean the data, and convert and save it into CSV format.
- To use the data obtained to train the models for both the innings separately.
- To add other variables to make a more accurate prediction of the score and calculate the error.
- To compare the performance of the models and choose the one which gives the least RMSE.
- To compare the models with the performance of the DLS method.

## 4.2. METHODOLOGY

Existing models such as DLS and VJD, help calculate the proportion of a team's runs that it is projected to have scored, depending on the number of overs already faced and the number of wickets lost before the interruption. This is a simple approach, not factoring in the playing conditions or history. As has been noted by many experts, this is a controversial approach. This study aims to design a predictive model that forecasts the score given a certain number of factors, such as runs scored by the opponents, historical results between the teams, conditional results, type of resources left. This would be done by analyzing the significance of each covariate and mapping the highly significant covariate to find a causal link between the numbers of runs required by the chasing team in the number of balls of left.

The project can be divided into 3 part:

1. Data scraping and cleaning
2. Model training
3. Model comparisons and conclusions




### 4.2.1. DATA SCRAPING AND CLEANING

Data for this study is collected from official records of the games that have been played. Proper cleaning of the data would require planning and organization of the data. The data is scraped off http://stats.espncricinfo.com [11]. The files scraped are in yaml format. This data is cleaned, converted into the desired format, organized, and saved in CSV format.

The match data is divided for 2 innings and further into training and testing set. We also collected ground averages of 100 stadiums. The ball by ball scores of each inning is paired with the ground average of the respective match.

| | match_id | balls | runs | wickets | ground |
|---|---|---|---|---|---|
| 17645 | 65 | 19 | 4 | 1 | New Wanderers Stadium |
| 17646 | 65 | 20 | 5 | 1 | New Wanderers Stadium |
| 17647 | 65 | 21 | 5 | 1 | New Wanderers Stadium |
| 17648 | 65 | 22 | 5 | 1 | New Wanderers Stadium |
| 17649 | 65 | 23 | 6 | 1 | New Wanderers Stadium |
| 17650 | 65 | 24 | 6 | 1 | New Wanderers Stadium |
| 17651 | 65 | 25 | 6 | 1 | New Wanderers Stadium |
| 17652 | 65 | 26 | 10 | 1 | New Wanderers Stadium |
| 17653 | 65 | 27 | 13 | 1 | New Wanderers Stadium |
| 17654 | 65 | 28 | 13 | 1 | New Wanderers Stadium |
| 17655 | 65 | 29 | 13 | 1 | New Wanderers Stadium |

Fig. 3. Cleaned data of 1st Innings

| | match_id | balls | runs | wickets | ground_average | pp_balls_left | total_overs |
|---|---|---|---|---|---|---|---|
| 101 | 1 | 102 | 80 | 0 | 242 | 0 | 50 |
| 102 | 1 | 103 | 80 | 1 | 242 | 0 | 50 |
| 103 | 1 | 104 | 80 | 1 | 242 | 0 | 50 |
| 104 | 1 | 105 | 80 | 1 | 242 | 0 | 50 |
| 105 | 1 | 106 | 80 | 1 | 242 | 0 | 50 |
| 106 | 1 | 107 | 80 | 1 | 242 | 0 | 50 |
| 107 | 1 | 108 | 80 | 2 | 242 | 0 | 50 |
| 108 | 1 | 109 | 80 | 2 | 242 | 0 | 50 |
| 109 | 1 | 110 | 80 | 2 | 242 | 0 | 50 |
| 110 | 1 | 111 | 80 | 2 | 242 | 0 | 50 |
| 111 | 1 | 112 | 80 | 3 | 242 | 0 | 50 |

Fig. 4. Sample of the cleaned data for 1st innings

| | match_id | balls | runs | wickets | ground_av | pp_balls_l | target | total_overs |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 242 | -59 | 271 | 50 |
| 1 | 1 | 2 | 0 | 0 | 242 | -58 | 271 | 50 |
| 2 | 1 | 3 | 0 | 0 | 242 | -57 | 271 | 50 |
| 3 | 1 | 3 | 1 | 0 | 242 | -57 | 271 | 50 |
| 4 | 1 | 4 | 1 | 0 | 242 | -56 | 271 | 50 |
| 5 | 1 | 5 | 1 | 0 | 242 | -55 | 271 | 50 |
| 6 | 1 | 6 | 1 | 0 | 242 | -54 | 271 | 50 |
| 7 | 1 | 7 | 3 | 0 | 242 | -53 | 271 | 50 |
| 8 | 1 | 8 | 3 | 0 | 242 | -52 | 271 | 50 |
| 9 | 1 | 9 | 3 | 0 | 242 | -51 | 271 | 50 |
| 10 | 1 | 10 | 3 | 0 | 242 | -50 | 271 | 50 |

Fig. 5. Sample of the cleaned data for 2nd innings

| | ground | ground avergae |
|---|---|---|
| 0 | Civil Service Cricket Club, Stormont, Belfast - Ireland | 181.5967742 |
| 1 | b"Lord s, London - England" | 237.2692308 |
| 2 | Kennington Oval, London - England | 247.0588235 |
| 3 | Riverside Ground, Chester-le-Street - England | 233.8333333 |
| 4 | Grange Cricket Club, Raeburn Place, Edinburgh - Scotland | 225.5526316 |
| 5 | Old Trafford, Manchester - England | 230.3235294 |
| 6 | Headingley, Leeds - England | 277.5666667 |
| 7 | VRA Ground, Amstelveen - Netherlands | 221.0588235 |
| 8 | Harare Sports Club - Zimbabwe | 212.5681818 |
| 9 | Cambusdoon New Ground, Ayr - Scotland | 163.1428571 |
| 10 | Toronto Cricket, Skating and Curling Club - Canada | 189.0625 |

Fig. 6. Sample of the ground averages csv file

### 4.2.2. MODEL TRAINING AND TESTING

The data is divided into two parts. The first half is used for training the model, and the second half is used for testing the model. The models are designed to predict the score at every ball and tested to cross-check with the actual final score.

The models are trained using different algorithms to find the best fit and to make sure that the model does not overfit. Different algorithms used are linear regression, lasso regression, random forest, gradient boost regression, and decision tree.

The algorithms used work in the following way:

1. Linear Regression

   Linear Regression estimates the true values (like number of calls, total sales, cost of houses, etc.) based on a continuous variable/variables. Here, we define an equation between independent variables and dependent variables by finding the best fit line. This line is called the regression line. It is represented by a linear equation "Y= aX + b" where:

   X: Independent variable

   Y: Dependent Variable

   a: Slope

   b: Intercept

2. LASSO Regression

   Least Absolute Shrinkage and Selection Operator, also called LASSO, performs both variable selections as well as regularization to improve the accuracy of prediction. Lasso regression is nothing but a type of linear regression that uses the shrinkage process. Shrinkage is where data values are shrunk towards a central point, like the mean. This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate some parts of model selection, like parameter elimination/ variable selection. Some coefficients can become zero and eliminate from the model due to regularization. Larger penalties result in coefficient values closer to zero, which is ideal for producing simpler models.

3. Decision Tree

   Decision Tree is one of the types of supervised machine learning algorithm that works for problems which require classification. It also works on both continuous and categorical dependent variables. In this algorithm, we split the data into at least two similar sets. The sets are made based on the most significant attributes/ independent variables to make as distinct sets as possible.

4. Gradient Boost Regression

   Gradient boosting is a technique of machine learning used for regression and classification problems. This method produces a prediction model in the form of a collection of weak prediction models, typically decision trees. It builds the model in a stage-wise manner as the other boosting methods do. It then generalizes them by allowing optimization of a random differentiable loss function.

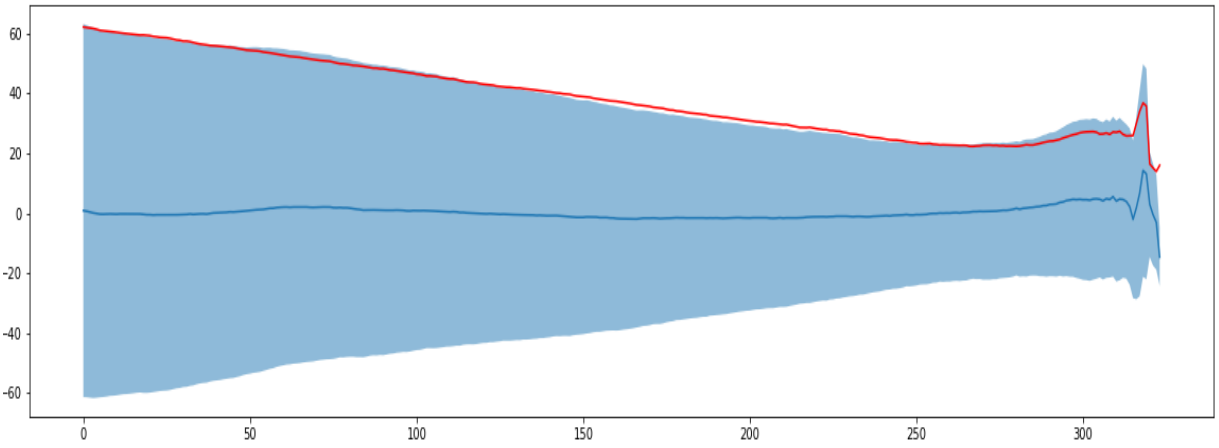
5. Random Forest

   Random Forest is a term for an ensemble of decision trees. In this, we have a collection of decision trees (also called "Forest"). To classify a new object based on features, each tree gives a classification. This classification given by the tree is said to be the "class voted by the tree." The forest selects the tree that has the most number of votes (over all the trees in the forest).

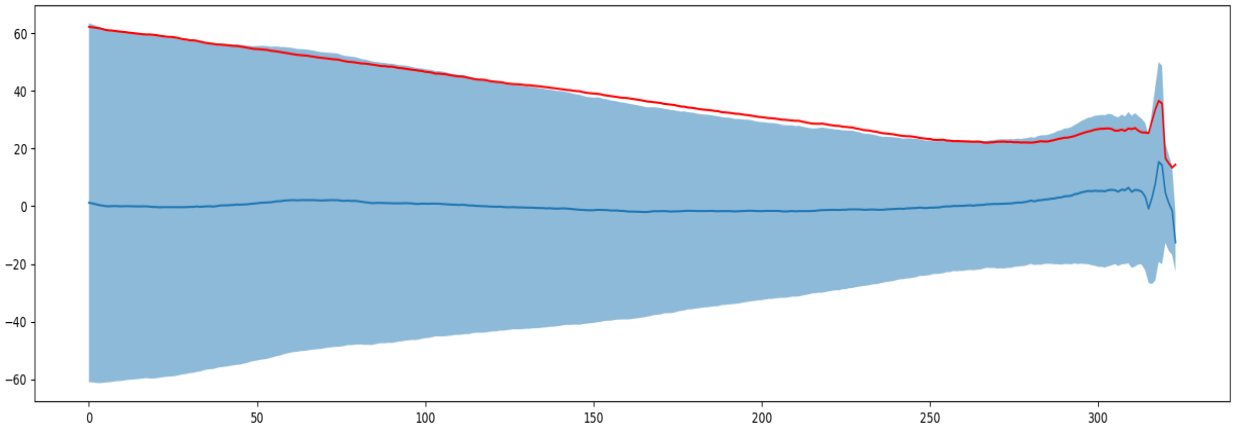
The models are trained on the Training data set. The data set is grouped by  Since we have only about 1000 match data in the training data set, we have done K-Fold cross-validation. In this, we take the data set and divide it into 10 parts(folds). We train the program on 9 folds and test the model on the remaining fold. The sequence of the data is never changed. This method helps in handling the overfitting of the models.

The predicted final score is compared with the actual final score, and the error is recorded. This error is used to calculate the RMSE and average error and plot it.

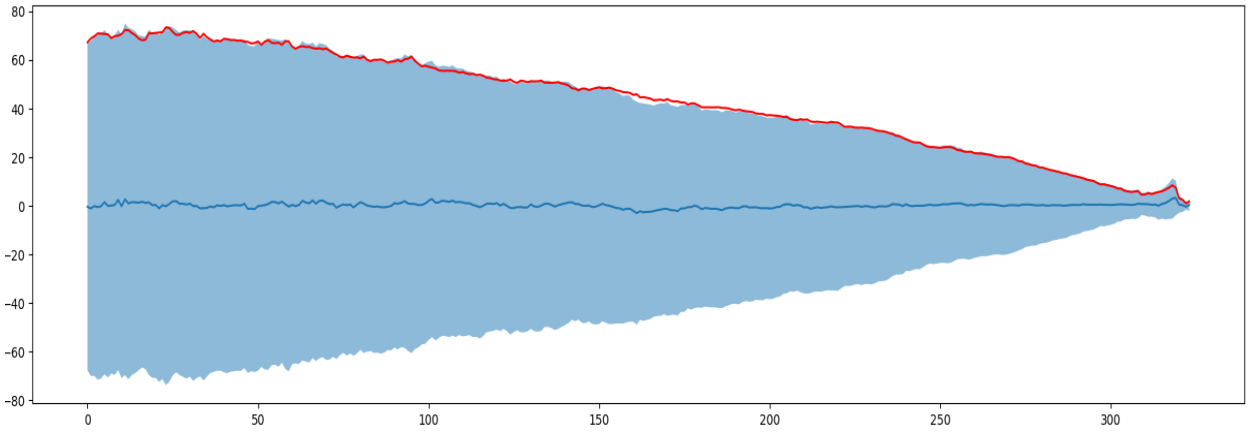
Below are the graphs of errors between the predicted final score and actual final score and RMSE plotted at every ball. The blue line marks the average error, and the red line marks the RMSE. The shaded region shows the region between the maximum and minimum deviation of error at each ball.



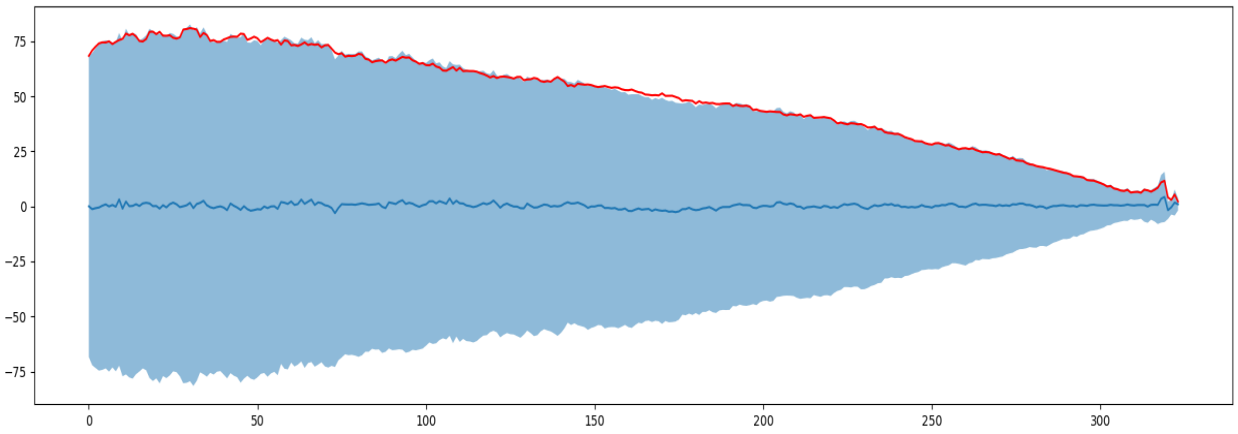

a.



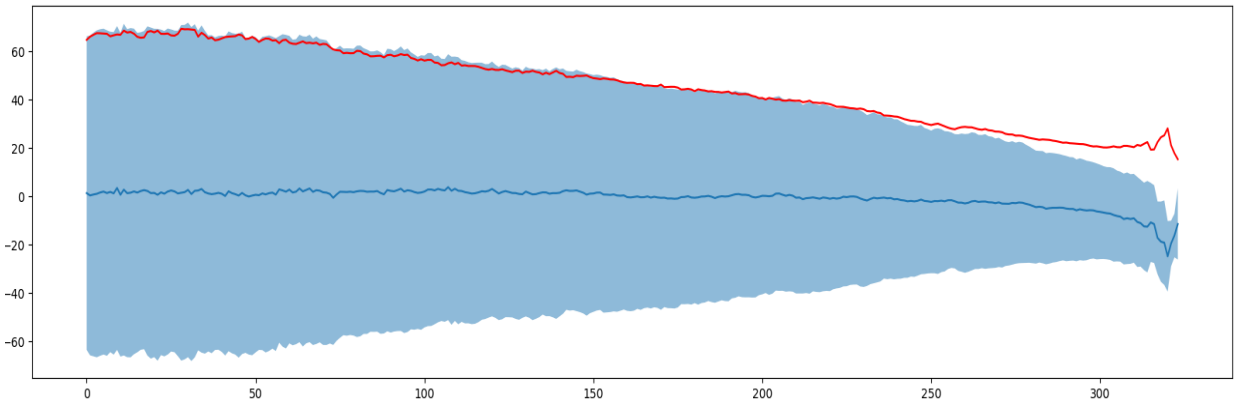

b.




c.




d.




e.

Fig. 7. RMSE and error graphs: a. Linear Regression, b. LASSO Regression, c. Decision tree Regression, d. Gradient Boost Regression, e. Random Forest Regression

Initially, the models were trained on a subset of the training data set( about 80 matches). After performing K-Fold validation and calculating the RMSE for the 1st innings, it is plotted on a graph along with the errors. Fig. 8 shows the result obtained.

It is clear from the graph that Linear regression and LASSO regression gave more accurate results with consistent lower RMSE values, while RMSE of other models is significantly higher with Decision Tree having the highest values.

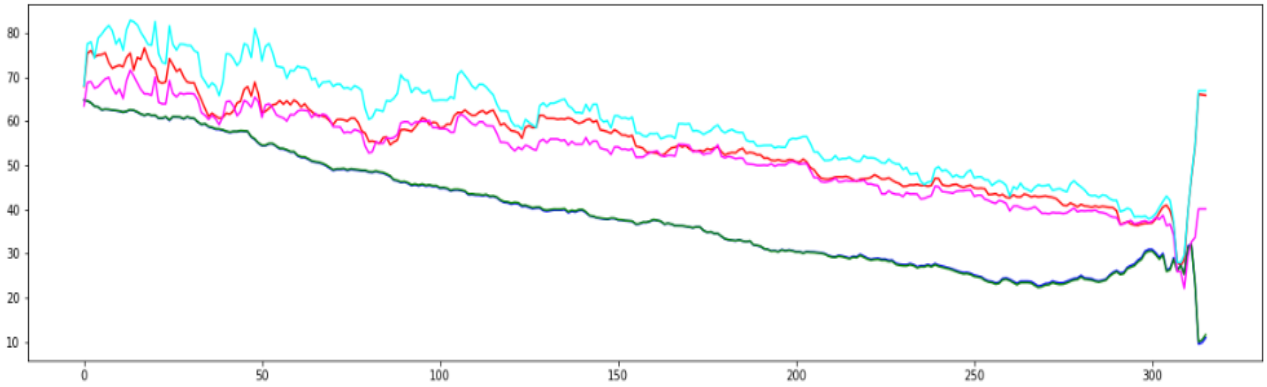



Fig. 8. Preliminary Result of RMSE graph. Red: Random Forest, blue: Linear Regression, green: LASSO Regression, magenta: Gradient boost Regression, cyan: Decision Tree

We expected to have similar results when tested on the complete training data set, although the end results were a little different. Linear Regression and Lasso Regression show consistent lower RMSE values till 40 overs, but then it spikes suddenly while Decision tree and random forest regression have lower RMSE after 40 overs, therefore giving a better performance than the other models.

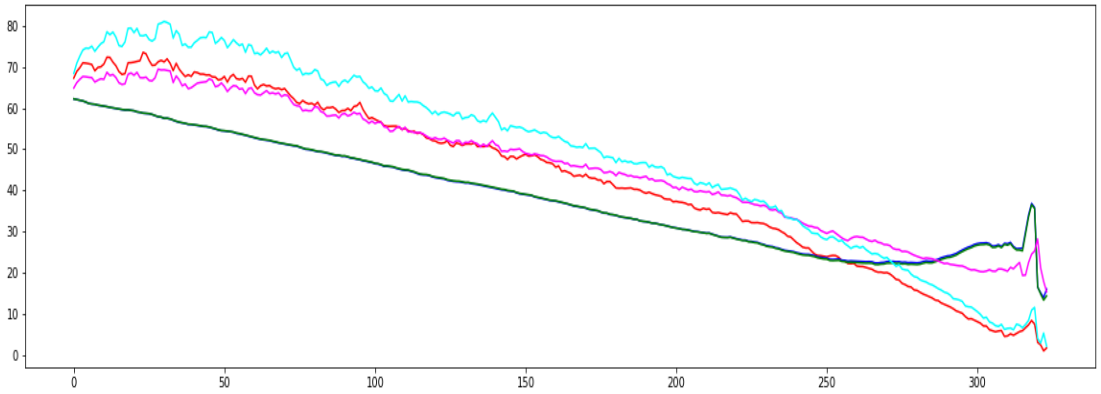



Fig. 9. RMSE result of all models graph. Red: Random Forest, blue: Linear Regression, green: LASSO Regression, magenta: Gradient boost Regression, cyan: Decision Tree

We take a match at random from the testing data set and try to predict the final score after each ball based on Lasso Regression. Fig. 10 illustrates the results obtained. The initial RMSE value is low(about 50) and reduces to almost by 12 overs(72 balls). But it increases again at about 130 balls, which is not acceptable.

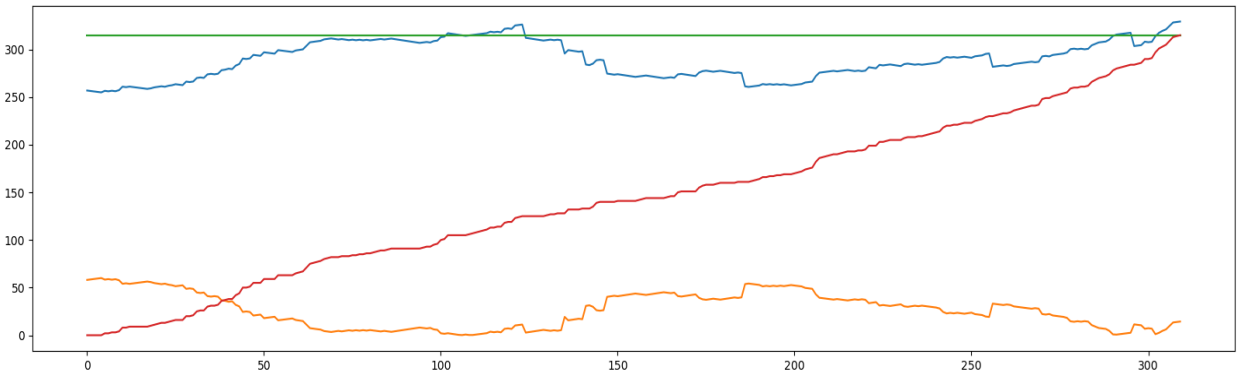



Fig. 10. Lasso regression model test result. Red:Runs scored till that ball, blue:Final Score prediction at that ball, green:Actual final score of the match, yellow:RMSE of predicted and actual final score

Since this model is not acceptable, we test other models on a random match. After testing, we found that the random forest regression model to give better performance than others. We made a continual training model, i.e., after predicting the final score of a match and calculating error, we retrain the model with this tested match. This makes our model a continuously updating model, which will help in future predictions. Fig. 11 shows the RMSE graph of continually updating the model tested on the complete test data set. Even though the initial RMSE is high, it is continuously decreasing and approaches zero by the end of the match.

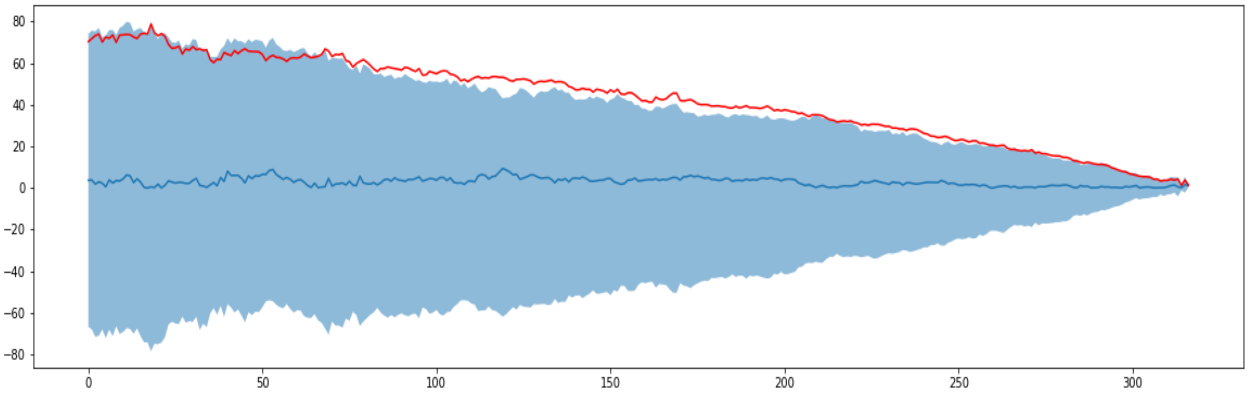
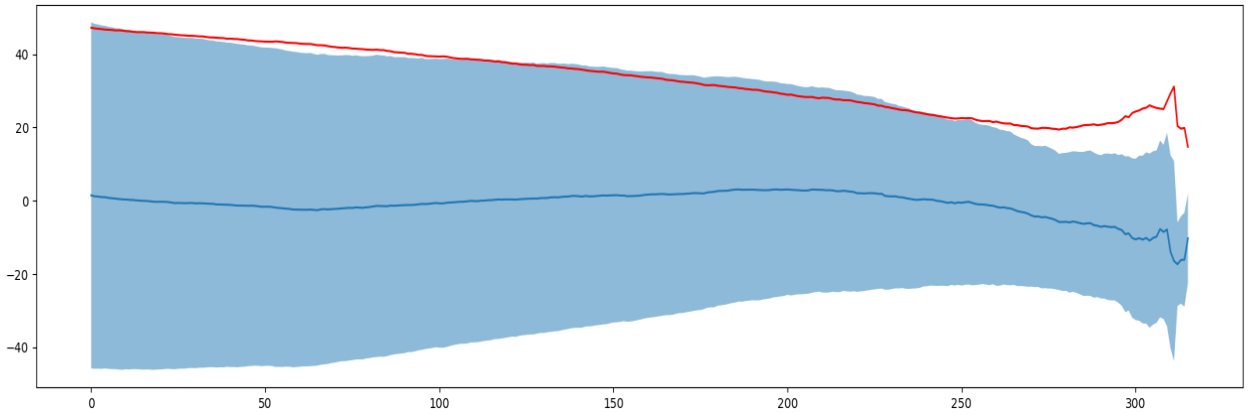


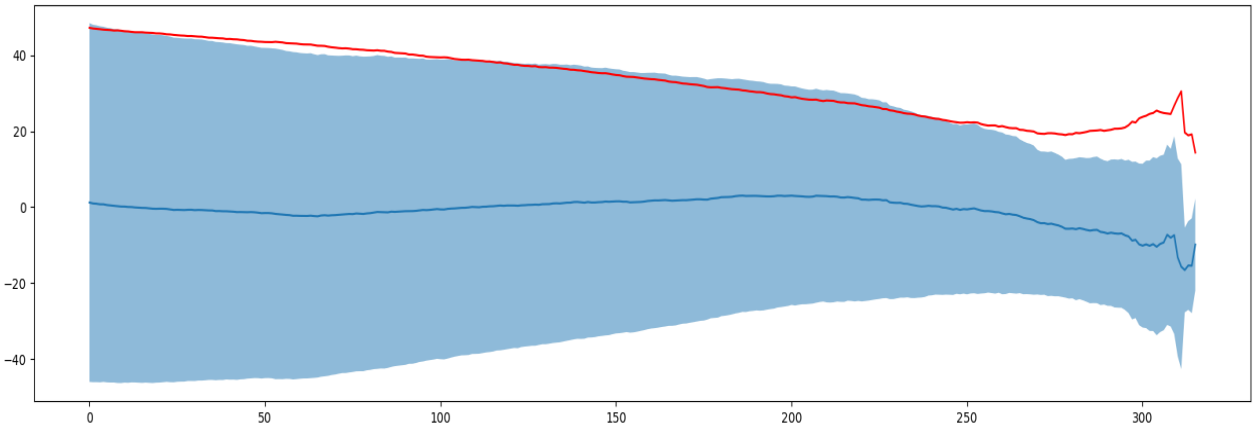

Fig. 11. Continuouly updating Random Forest Regression model performance graph of 1st innings

Similar steps have been taken to access the 2nd innings. We have taken the same 5 models and trained the on the training data set for 2nd inning. The final score of matches is predicted and is compared with the actual final score, and the error is recorded. This error is used to calculate the RMSE and average error and plot it. 2nd innings model training has an extra parameter of the target, which changes he playing strategy of the team batting second.

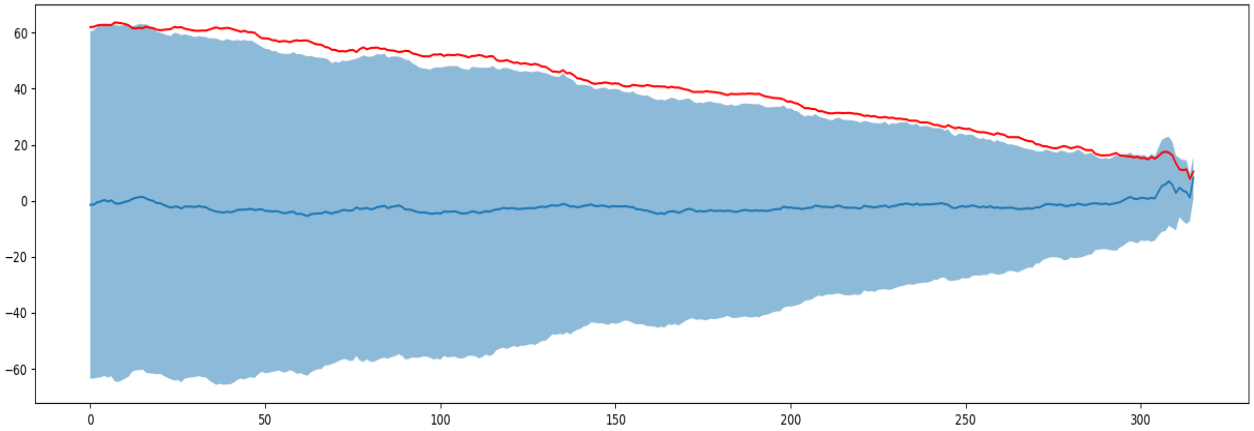
Below are the graphs of errors between the predicted final score and actual final score and RMSE plotted at every ball. The blue line marks the average error, and the red line marks the RMSE. The shaded region shows the region between the maximum and minimum deviation of error at each ball.



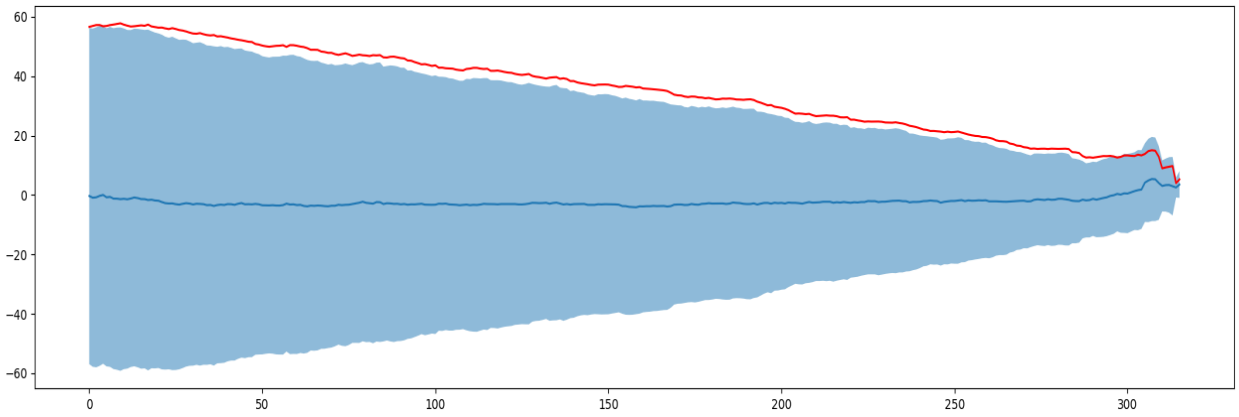

a.



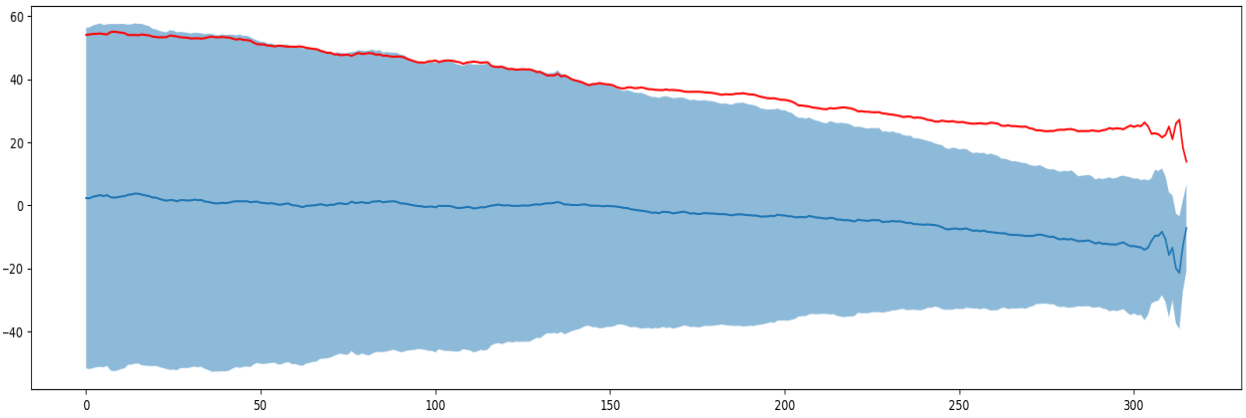

b.




c.




d.




e.

Fig. 12. RMSE and error graphs: a. Linear Regression, b. LASSO Regression, c. Decision tree Regression, d. Gradient Boost Regression, e. Random Forest Regression

It can be noticed from the above graphs that as in the 1st innings linear regression and Lasso regression have lower RMSE in the beginnings, but it does not reduce significantly towards the end of the innings, which is desired. The decision tree model also gives similar results.

These 2 models are taken and tested on a random match from the test data set. Fig.13 shows the results for the test of the decision tree model, and fig. 14 shows the result for the random forest model.

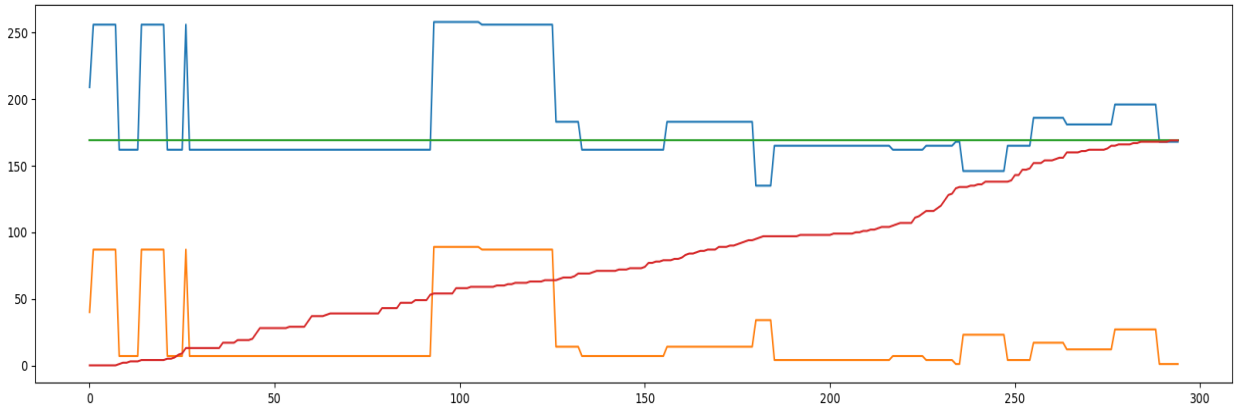



Fig. 13. Decision Tree model test result. Red: Runs scored till that ball, blue: Final Score prediction at that ball, green: Actual final score of the match, yellow: RMSE of predicted and actual final score

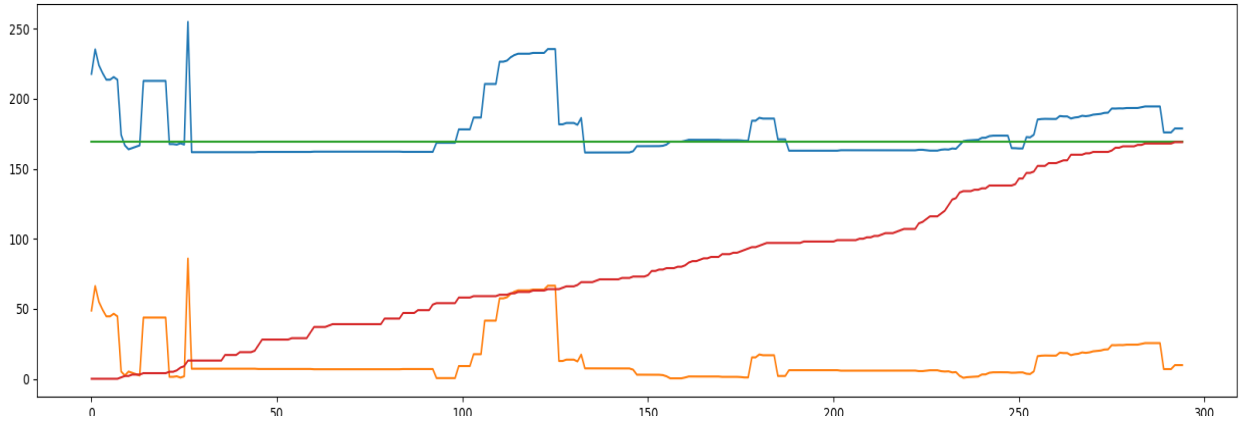



Fig. 14. Random Forest regression model test result. Red: Runs scored till that ball, blue: Final Score prediction at that ball, green: Actual final score of the match, yellow: RMSE of predicted and actual final score

Comparing the 2 examples, we see that random forest regression gives a more consistent output, which is closer to the actual result as compared to the decision tree. So we decide to use random

forest regressor model to test the whole test data set and train on it match by match. This makes it a continuous updating model. Fig. 15 shows the plot of average RMSE values at each ball. The red line denotes RMSE values. The blue line shows the average error, and the shaded blue region shows the deviation from the mean.

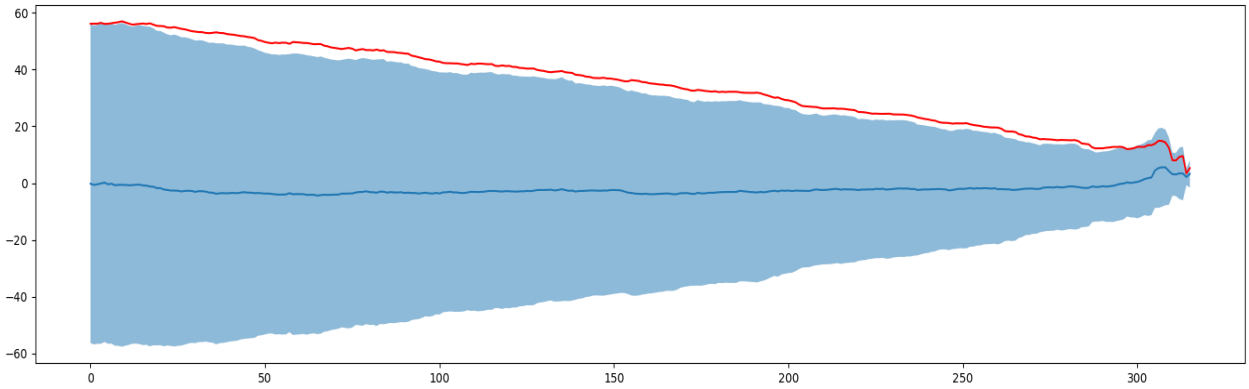


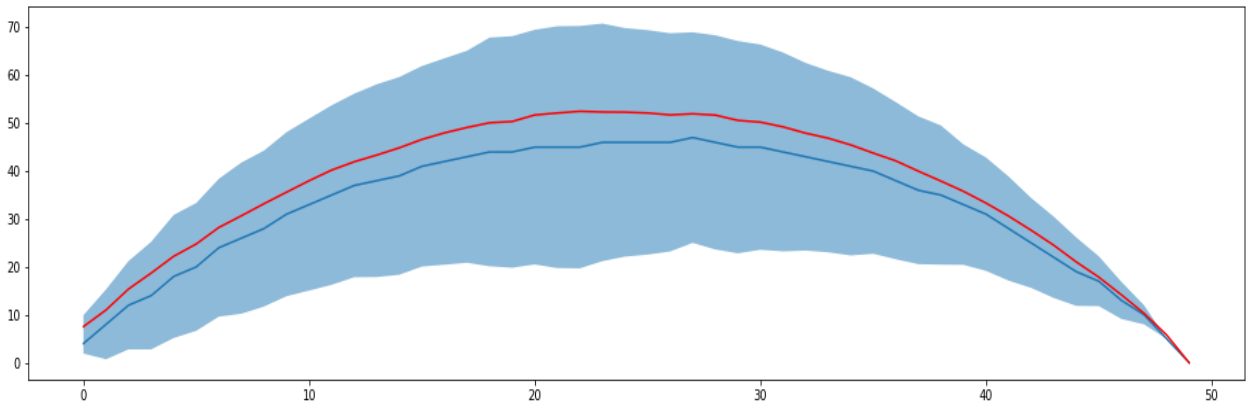

Fig. 15. Continuously updating Random Forest Regression model performance graph on test data set of 2nd innings

### 4.2.3. MODEL COMPARISON

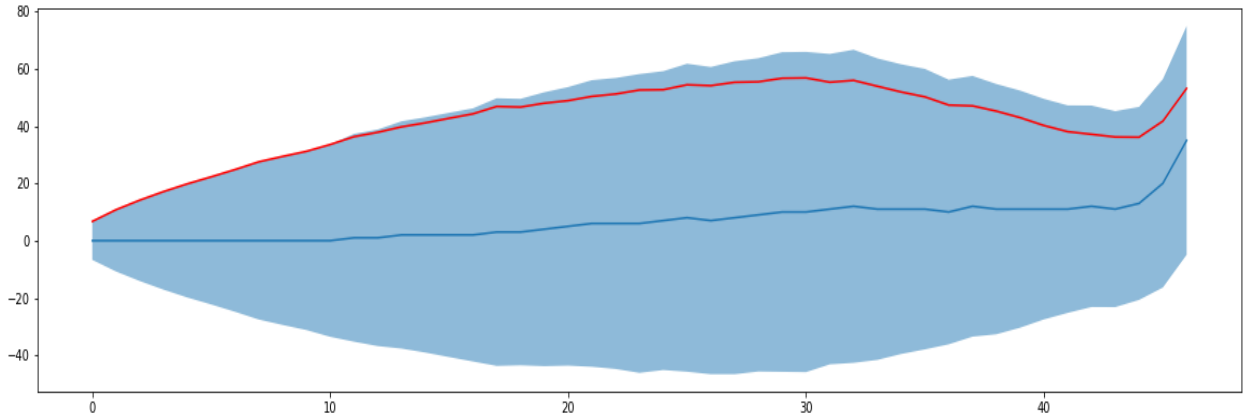
Since we took the DLS method as the basis for our project, to compare our models, we made a model for the DLS method. Since the mathematical formula for the DLS method is confidential and not disclosed online, only a table is available with guidelines on how to use it.

We have used the same guidelines to develop a model of the DLS method, which we could use to compare our model.

Fig. 16 shows the graphs of the DLS method tested our training dataset to see how it performs.




a.




b.

Fig. 16. a. DLS method prediction error in 1st innings b. DLS method prediction error in 2nd innings

As can be noticed, the error in both the graphs follows a somewhat bell-shaped curve, which has a minima at the 25th over in 1st innings and 30th over in the 2nd innings.

Fig. 17 shows the predicted score by our method and target set by the DLS method for 1st innings.

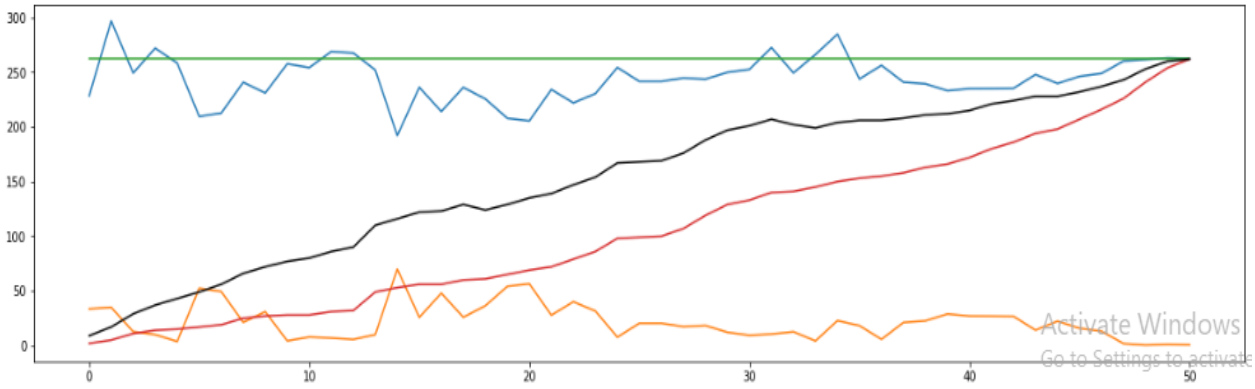



Fig. 17. Lasso regression model test result for 1st innings. Red: Runs scored till that ball, blue: Final Score prediction at that ball, green: Actual final score of the match, yellow: RMSE of predicted and actual final score, black: Target set by DLS method

Basis of the DLS method is that when a match is interrupted at any point in 1st innings, the DLS method calculates the target for 2nd innings, which needs to be scored under the revised conditions. At the same time, our models predict the score at 50th over irrespective of where the match is interrupted.

Fig. 18 shows the predicted score by our method and target set by the DLS method for 2nd innings.

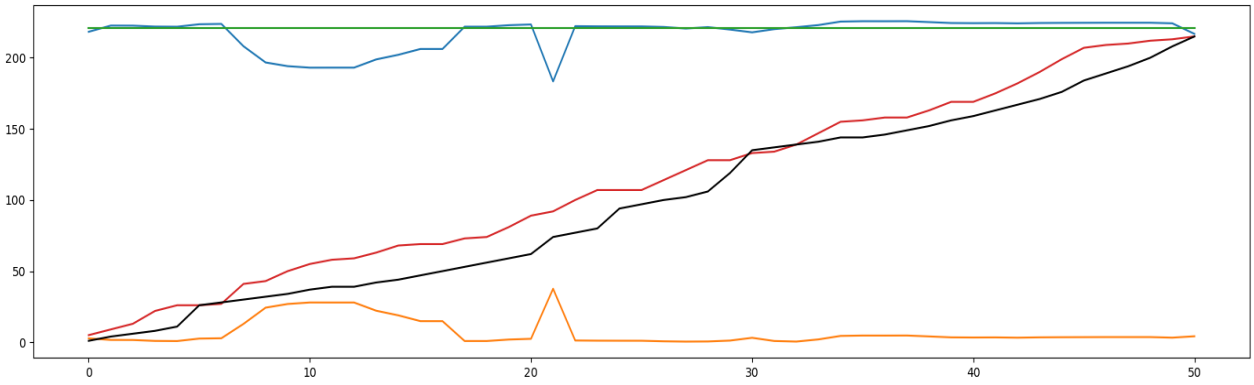
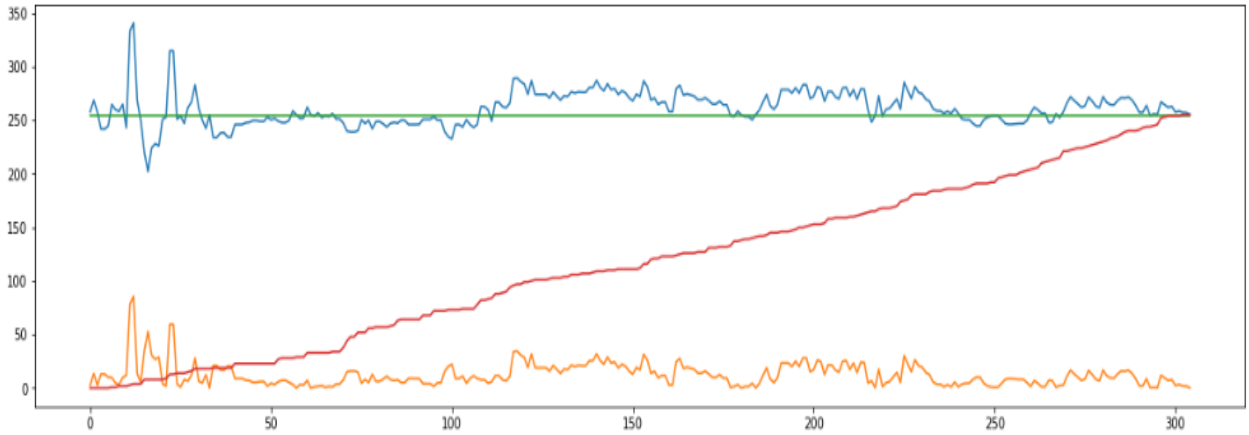



Fig. 18. Lasso regression model test result for 2nd innings. Red: Runs scored till that ball, blue: Final Score prediction at that ball, green: Actual final score of the match, yellow: RMSE of predicted and actual final score, black: Target set by DLS method

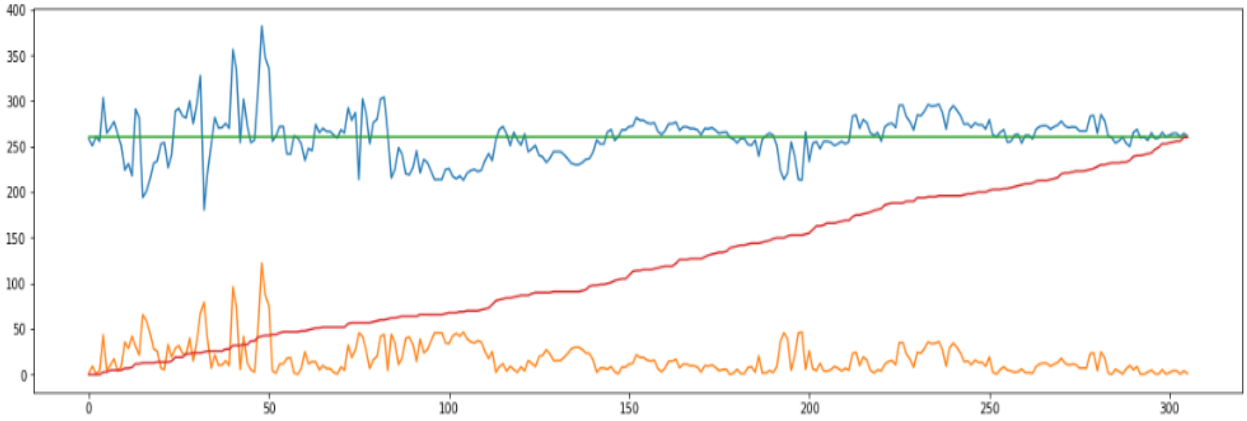
# CHAPTER 5. RESULT ANALYSIS

To conclude, we decided to use the models which show consistently decreasing RMSE in both the innings. For both the innings Random Forest Regression model is selected as it shows a consistent performance over extensive data set, unlike the other models.

Fig. 19 and fig. 20 shows an example of the predicted final scores of matches for both the innings taken from outside the data set.

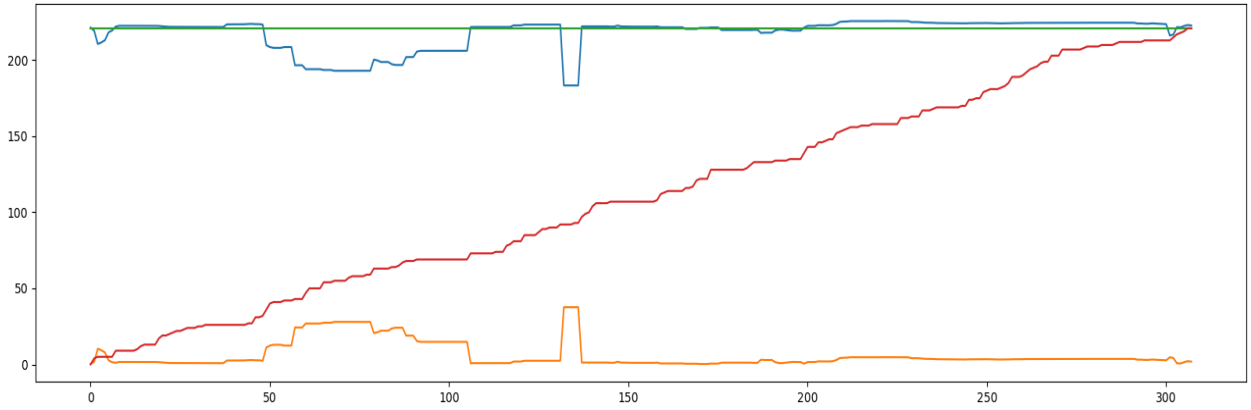



a.

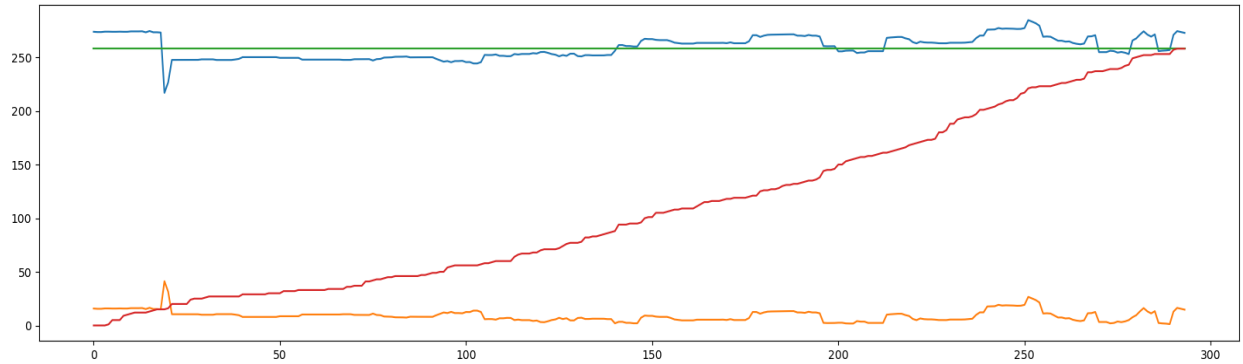



b.

Fig. 19. a, b: Random Forest regression model test results for 1st innings. Red:Runs scored till that ball, blue:Final Score prediction at that ball, green:Actual final score of the match, yellow:RMSE of predicted and actual final score




a.




b.

Fig. 20. a, b: Random Forest regression model test result for 2nd innings. Red: Runs scored till that ball, blue: Final Score prediction at that ball, green: Actual final score of the match, yellow: RMSE of predicted and actual final score

It is clear from the above illustrations that our algorithms give consistent output with very low RMSE for both the innings.

# CHAPTER 6. CONCLUSIONS AND SCOPE FOR FUTURE WORK

## 6.1. CONCLUSION

- Our model is fundamentally different from the DLS method as the DLS method resets the target for revised conditions. In contrast, our model predicts the scaled-up score at $50^{th}$ over with respect to the changed conditions.
- The DLS method does not take into account the variability of other features like ground where the match is being played, scoring pattern in power-play overs, player performance over the years, etc. Our model takes into account the history of ground averages and scoring patterns in power-play overs.
- The DLS model is a constant formula based on the number of overs left and the number of wickets fallen. Our algorithm keeps training the models after every match is played, which makes it a continually updating model whose performance will improve with time, unlike the DLS method.

## 6.2. FUTURE SCOPE

Our model has incorporated two features, which are ground average and scoring pattern in power-play overs. This model can further be improved by adding more such features, which will help give better performance and more accurate prediction.

The model can be modified by incorporating other features such as:

- Players' performance over the years(player's statistics),
- Depth of batting in the lineup,
- Head to head records of the two teams.
- Result of the toss
- Ground's history (ratio number of matches won by the team batting first to the team batting second)
- Teams' record of successfully defending a target and chasing a target

The model can also be modified to predict the score at any given point of the match to set a target for the team batting second in a limited number of overs.

# REFERENCES


[1] Predictive Analytics, www.Wikipedia.org (3/1/20)

[2] Geddam Jaishankar Harshit, Rajkumar S., "A Review Paper on Cricket Predictions Using Various Machine Learning Algorithms and Comparisons Among Them", International Journal for Research in Applied Science & Engineering Technology, ISSN: 2321-9653; IC Value: 45.98

[3] T. Prabhakar Reddy et al., "A Method for Resetting the Target in Interrupted Twenty20 Cricket Match" pp.226-234, Journal of Physical Education and Sport Science ISSN 2229-7049, Vol. 2, 2014.

[4] Kalpdrum Passi and Niravkumar Pandey," INCREASED PREDICTION ACCURACY IN THE GAME OF CRICKET USING MACHINE LEARNING", Vol.8, No.2, March 2018.

[5] Madan Gopal Jhanwar and Vikram Pudi, "Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach", European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, September 2016.

[6] Ayush Kalla, "AutoPlay - Cricket Score Predictor", IJESC, Volume-8 Issue-4, April 2018

[7] FC Duckworth and AJ Lewis, "A fair method for resetting the target in interrupted

one-day cricket matches", Stinchcombe, Glos. and University of the West of England, November 1997.

[8] Aminul Islam Anik," Player's Performance Prediction in ODI Cricket Using Machine Learning Algorithms", IEEE, 2018.

[9] A do it yourself tutorial on the D/L method, www.sportskeeda.co

[10] V. Jayadevan, "A new method for the computation of target scores in interrupted, limited—ver cricket matches", Current Science, Vol. 83 No. 5, September 2002

[11] Records, http://stats.espncricinfo.com,(3/2/20)

[12] A do it yourself guide on the DL method, www.sportskeeda.com(25/2/20)