National College of Ireland

Project Submission Sheet – 2020/2021

Student Name:	Omkar Ratnoji Tawade		
Student ID:	19232136		
Programme:	MSCDAD_A	**Year:**	2020-21
Module:	Research in Computing		
Lecturer:	Noel Cosgrave		
Submission Due Date:	20-04-2021		
Project Title:	A comparative study of cricket par score using Machine Learning and DL method		
Word Count:	9700 words		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Omkar Ratnoji Tawade

Date: 20-04-2021

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A comparative study of cricket par score using Machine Learning and DL method

Research In Computing - Research Proposal

Omkar Ratnoji Tawade
19232136
MSc in Data Analytics

20th April 2021

Abstract

Cricket is one of the popular sport in the world. It was first played in the 16$^{\text{th}}$ century. It has been evolved a lot and now it is played in three formats that are Test Cricket (Unlimited Overs), One-Day Cricket, and T20 Cricket. Technology is now making its entry into the cricket. Whenever a game is interrupted by rain then runs for the over which get wasted due to an interruption is calculated by a method known as Duckworth Lewis. This method is depended on two factors that are overs remaining and wickets lost. Duckworth Lewis method is often criticised by the team because it's bias nature towards team batting in the first innings. This method was originally created for One-Day cricket but the same method is used to T20 cricket by scaling down the values which leads to disputable target sometimes. In this paper, I am proposing a method to calculate the score of overs which get wasted during an interruption using machine learning algorithms like adaptive boosting, gradient boosting and random forest along with feature engineering process. I will be using Indian Premier League (IPL 2020) data for this project.

Keywords Duckworth Lewis, Machine Learning, AdaBoost, Gradient Boosting, Random Forest

Area Data Mining and Machine Learning.

Contents

1 Introduction

Cricket is a sport which cannot be played while it rains. Before 1997 a rain-interrupted match score was calculated with the help of the average run rate score. If a match is interrupted in the second innings after 25 overs, then the runs of the team batting second were calculated with the help of the run rate of the team batting first when they were at the 25th over during their innings. This was very unfair for the team batting second because even if the team batting second was scoring close to the required run rate but if at that instant match is interrupted by the rain and the team batting first had scored less in remaining overs then the run rate of team 2 will decrease. This method was very simple to understand and teams used to take advantage of this method in rain interrupted matches. If rain is predicted in the match then the teams always decide to field first after winning the toss to take the advantage of flaws in this method. To avoid this unfairness, the Duckworth Lewis method was introduced for rain interrupted games. Duckworth Lewis method itself has few drawbacks but it has removed the majority of unfairness that existed in the previous average run rate method. The main difference between the average run rate method and the Duckworth Lewis method was the DL method considers the factor of wickets in hand or wickets lost. Frank Duckworth and Tony Lewis were the two statisticians and mathematicians who worked with England's Cricket Board to implement this method. It is an equation that requires few parameters like team 1's score and resources and team 2 resources. Overs remaining and the number of wickets lost is used to calculate the resources of respective teams. Duckworth Lewis proposed a table to calculate the percentage of resources used. Due to confidentiality reasons, Duckworth Lewis did not discuss how the entries in the table 1 were constructed. They had provided only partial information of the resource table creation and stated that table entries are based on 20 parameters in the equation 2 which are Z_0(w) and b(w) where w stands for number of wickets, w=0 to 9. They did not disclose the function of n(w) and stated lambda as a match factor.

$$Z(u, w|\lambda) = Z_0 F(w) \lambda^{n(w)+1} [1 - exp\{-b\mu/\lambda^{n(w)} F(w)\}] \tag{1}$$

If a match is interrupted by the rain in the first innings after 12 overs and till then if team 1 had managed to score 90 runs with the loss of two wickets then the resource used by team 1 will be 100 - 45.1 = 54.9%. This value is derived from the table 1 by referring to the overs available row and wickets lost column. Team 2 will have 12 overs to play the game but the target will be revised. In this calculation again table 1 is referred to know the resource used by team 2. Since team 2 is starting their innings, so the wicket loss is zero and the resource used by team 2 will be 100 - 66.4 = 33.6% . Thus they have 54.9 - 33.6 = 21.3% greater resource than team 1. So the revised target for team 2 would be 21.3% of 150 (Average T20 International Score) or 32 more runs than team 1 scored. The target for team 2 will be 122 runs.

$$Team\ 2's\ par\ score = Team\ 1's\ score * \frac{(Team\ 2's\ resources)}{(Team\ 1's\ resources)} \tag{2}$$

Duckworth Lewis is a very flexible method as it is very easy to use at any point of the match. If a match is interrupted a couple of times in the same innings then also be used and also if rain is interrupted in both innings then also it can be used. Duckworth Lewis is used in the second innings to decide the result of a T20 match only if 5 overs of the game has been played.

In this paper, researcher Steven (2016) modified the Duckworth Lewis calculation taking into consideration of modern-day cricket and the pace at which it is played. His method is more suitable for modern-day cricket and especially the T20 Cricket format which is the shortest format in cricket. The exponential curves in figure 1 were shifted upwards and because of this score calculation of the death over was more realistic. His method replaced the old Duckworth Lewis method and now it is known as Duckworth Lewis Stern Method(DLS Method).

We can use the reference of all studies mentioned in Section 2 to use different algorithms to predict the score of a match. In section 3 we will discuss our methodology which is not been used earlier. This study intends to predict the score of the match when a game is interrupted by the rain. We will be using the IPL 2020 season data for our analysis. We intend to use algorithms such as Gradient Boosting Regression, Adaptive Boosting Regression, and Random Forest Regression to predict the score of the match. The performance of each algorithm will be evaluated by Root Mean Squared Error(RMSE), Mean Squared Error(MSE), Mean Absolute Error (MAE), and R-square/Adjusted R-square to explain the model.

Table 1: Duckworth Lewis Resource Table (T20 Cricket Edition)

	Wickets Lost									
Overs Available	0	1	2	3	4	5	6	7	8	9
20	100	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	8.3
18	92.2	89.6	85.9	81.1	74.2	65	52.7	36.9	21	8.3
17	88.2	85.7	82.5	77.9	71.7	63.3	51.6	36.6	21	8.3
16	84.1	81.8	79	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	63.4	56.9	47.7	35.2	20.8	8.3
13	71	69.4	67.3	64.5	60.4	54.4	46.1	34.5	20.7	8.3
12	66.4	65	63.3	60.6	57.1	51.9	44.3	33.6	20.5	8.3
11	61.7	60.4	59	56.7	53.7	49.1	42.4	32.7	20.3	8.3
10	56.7	55.8	54.4	52.7	50	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	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	34.3	33.2	31.4	29	24.6	17.8	8.1
5	30.4	30	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
3	18.7	18.6	18.4	18.2	18	17.5	16.8	15.4	12.7	7.4
2	12.7	12.5	12.5	12.4	12.4	12	11.7	11	9.7	6.5
1	6.4	6.4	6.4	6.4	6.4	6.2	6.2	6	5.7	4.4

1.1 Motivation

Duckworth Lewis's method can be unfair sometimes and it is always slightly partial to the team batting second. In this method, team 2 knows the target so they can accelerate the innings accordingly. One of the controversial matches was a One Day International match between New Zealand and England on 12 June 2015. New Zealand scored 398 runs in the first innings, and England in return scored 345 runs with the loss of 7 wickets in 43.5 overs. England required 53 runs in 37 balls to win the match and at this moment rain interrupted the game. By the time rain stopped, only 13 balls were left and the revised target for England was 30 runs to win. When compared to 53 runs in 37 balls and 30 runs in 13 balls, it is clear that the first target is more achievable than the later one. This is due to the monotonic nature of the Duckworth Lewis system. In this scenario, runs are decreased slowly. In figure 1 we can see that line are flattened if the number of wickets lost is more which suggest that this method is static and do not consider the various factors such as players form and other factors which can be important to calculate the revised target.

The motivation for this project is to implement such a method that would be more realistic than the Duckworth Lewis method. A huge amount of data is generated every year and it is not being used to improve the experience of cricket when it is interrupted by weather conditions. Many matches are already impacted due to the Duckworth Lewis method and we don't wish to repeat the famous case of the South Africa vs England world cup match in 1992 which was interrupted by the rain and after the game was resumed South Africa needed 22 runs in 1 ball using most productive over method.

1.2 Research question

Can a machine learning model which use both match details as well as external features to determine the par score surpass the traditional Duckworth Lewis method?

1.2.1 Research Objective

The primary focus for this study is to create a machine learning model which will help to estimate the par score for rain interrupted T20 games using the IPL 2020 data. Along with this, research also tries to address to find below objectives.

- To find out the significance of winning the toss in a result of the match
- To find out the significance of playing in home matches in a result of the match.

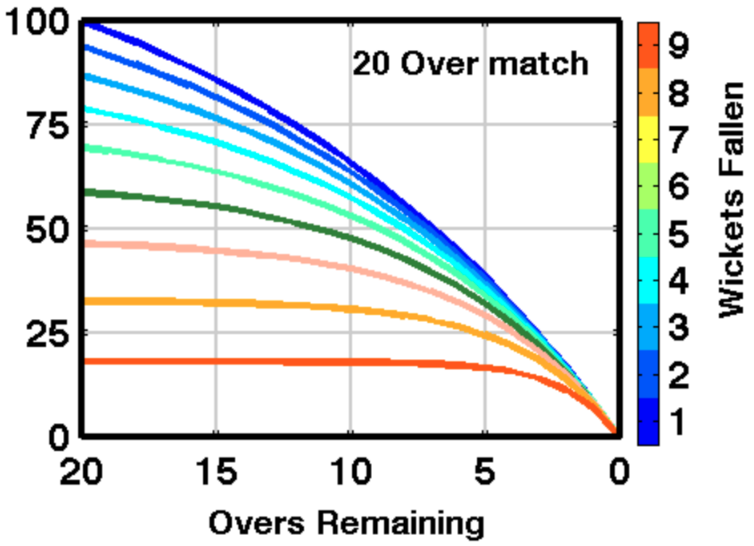


Figure 1: Duckworth Lewis Resource Graph - T20 Edition

1.3 Plan of Paper

In this paper, we are going to discuss some understanding and various results. In section 2 we will find different papers and will discuss the methodology, results of those papers. This section will help to find different techniques used earlier. Section 3 will introduce the proposed methodology in detail. Section 4 This section will have few subsections which explain the process of implementation in detail. Section 5 will discuss the implication of the paper, limitation of methodology.

2 Related Work

In this section, we will discuss the different methods or research used earlier to replace the DLS method. We discussed the DL method in detail and how it is implemented in a real match scenario. But first, we will need to know the kind of interruptions on which it can be used. If rain interrupted the first innings is the first case. The second case will be the repeated interruption in the first innings. The third case will the interruption between the innings break which results in the delay of the start of second innings and shorten of over in second innings. The fourth case will be the repeated interruption in the second innings. Fifth and the last case will be no resumption of the game after an interruption in the second innings. Duckworth Lewis is a statistical method and we will discuss how machine learning will be used to determine the par score or an outcome of the match using the past data. Data consists of the ball by ball information of every match of IPL season 2020. Regression methods will be more suitable for this type of data. Regression methods will help to allow the addition of independent variables such as toss, venue, etc. which can be significant in our analysis to predict the par score.

In this paper, Sambasivarao et al. (2014) reviewed all rain rules which were implemented in the cricket game. They discussed the average run rate(ARR) method which was first implemented in the cricket matches which were interrupted by the rain. In this method, a team that has a high average number of runs per over is declared as a winner. Then they discussed the most productive over(MPO) method in which the target of team 2 for particular overs is calculated by summing up the runs scored by team 1 in the same number of highest scoring overs. After this method, they discussed the discounted most productive overs (DMPO) which is the same as the most productive over method but the only difference is that runs from the most productive overs were decreased by 0.5% for each over lost. Researchers also discussed the Parabola (PARAB) method in which they discussed how a South African statistician do Rego proposed a parabolic equation to calculate the target. It performed better than ARR but it did not consider the number of wickets lost. After these methods, they discussed the Duckworth Lewis method. V. Jayadevan(VJD) was the last method implement for rain interrupted games. It is similar to the Duckworth Lewis method but the resource table is calculated such that team 2 who is chasing the target will always get a more achievable target unlike in some cases of Duckworth Lewis. They explained this difference by applying and comparing the DL method and VJD method in different situations of the match. They discussed how a T20 edition of the DL table is produced from the 50 over DL table by just dividing each value in the table by 0.566. Since T20 cricket is the shortest format of cricket so the scoring rate and rate at which wickets are lost is different from the one-day cricket and using the table derived from the one-day cricket DL resource table will not estimate a good target in T20 format. Researchers proposed a method in which they used the DL resource table of one-day cricket but they multiplied the target by 2.5 because the T20 format is 2.5 times shorter than ODI cricket. They provided an example of how their target was more reasonable compared to the target derived from the DL method or VJD method. They also used their method of multiplying the target by 2.5 on the VJD method and found that the target obtained was more like the T20 format target.

In this paper, Preston and Thomas (2002) developed a simulation technique for rain interrupted games. They proposed a method of preserving probability of winning the game of both team. They divided the rain rule into two scenario. In first scenario they discussed about the method of preserving probability in second innings. They formulated an equation with help of parameters like overs remaining, runs accumulated after number of overs, wickets lost. They preserved the run rate at which the team was scoring before the interruption. In second scenario they proposed a method to preserve the winning probability by formulation equation depending upon the parameter like number of balls got wasted in the interruption. Their equation predicted the runs at optimal rate for the number of overs which got waster during the interruption. Since these techniques is based on preserving the probability of winning of both the team, researchers explained how their proposed technique will fail to declare a winner in case of premature termination of the match. They suggested that in such case team which was ahead before the interruption should declare as a winner.

Bhattacharya et al. (2011) in their research first discussed how Duckworth Lewis constructed the table for one day cricket. They questioned about the possibility of other parametric curves that could be better fit, is there any advantage of using parametric fit? They also questioned about the resource value which are constant in last tow columns of the table 1. They proposed a non parametric method to replace the Duckworth Lewis table in T20 cricket. They used the ball by ball data from the cricinfo website and constructed a matrix where they average the scores of each over from all innings. They observed that their table does not exhibited the monotonicity. If a resource table is constructed then it should always increase from right to left but due to less sample size there were abnormal value generated in the table. They imposed the monotonicity using isotonic regression since resource table is always non decreasing from left to right. They found that after applying the isotonic regression, adjacent value of their non parametric resource table was same which suggested that this approach is not good. They used the Gibbs sampling method to estimate the values of the resource table. Due to use of sampling technique, values in adjacent values are not same in Gibbs sampling version of table. They compared their non parametric resource table using Gibbs sampling method with Duckworth Lewis resource table and found that the target computed by using their resource is always slightly higher compared to Duckworth Lewis table. They concluded that target computed by using their resource table is more T20 like target.

In this paper, Mchale and Asif (2013) proposed a two improvements in the Duckworth Lewis method. In first step they proposed a methodology to adopt an alternative estimation for F(w) and in second step to use a different model for Z(u,w) in the equation 1. They explained F(w) was dependent nine parameters as w ranges from 0 to 9. They found that main problem is to smooth the F(w) which they found by observing the plot of DL method in which second wicket partnership is weighted few runs than first and third wicket runs partnership. To remove this non linearity they decided to smooth the F(w), so they had used various function forms such as Cauchy, Gamma, Wei- bull and negative binomial distributions. They stated that due to this smoothing of F(w) the Z_0 value changed from 288.6 to 340. After smoothing their model was super-fit to the data. Researchers suggested that their model performs better for T20 cricket as the curves are not flattened quickly at end of innings,

In this paper, Sankaranarayanan et al. (2014) proposed a model for predicting game progression and outcome in one-day cricket. They worked on predicting the outcome of a match from any point of the match depending upon the runs scored and wickets lost at that instant of the match. Using the runs available and wickets remaining data they stated an equation to calculate the runs at the end of the innings by using the number of wickets as weight. Duckworth Lewis's method also works on the same principle. To increase more accuracy they divided their problem into predicting runs for home matches and away matches. Their experiment concluded that ridge regression for home matches degraded the model accuracy and the margin of error of predicting runs was too high. They implemented different models for home and away matches. For home matches, they used the attribute bagging ensemble method with nearest neighbour clustering. The performance of attribute bagging was better than the plain nearest neighbour algorithm. Their model predicted the runs at end of the innings with an accuracy of 70%.

In this paper, Phanse and Deorah (2011) researcher explained the shortcomings of the Duckworth Lewis method using random forest and C4.5 data mining algorithm. They used a data set consisting of 50 matches in which Duckworth Lewis method was used. Dataset consisted a total of 32 features They used WEKA tool to identify patterns in the data. These pattern identification helped them to conclude that team wining the toss win the DL matches, DL is more biased to team batting first, wickets fell more often when DL is used. They used the parameters such as overall probability of winning of a team, toss, weather and venue to build a heuristic model to predict the winner. They trained the data on WEKA tool using C4.5 an Random Forest algorithm to build a classifier which predicts the winner of DL matches. They concluded that their model predicted with 70% accuracy which suggest that Duckworth Lewis method is biased. They proposed a method to improve the DL method by adding a factor which represented a impact of home or away matches.

In this paper, Ali and Kusro (2018) researcher proposed a method to improve the Duckworth Lewis method. They used the player rankings to find the solution for DL method. They formulate a mathematical equation in which they used the International Cricket Committee(ICC) career rating points of the batsmen, current rank of the batsmen, remaining overs or overs lost during the interruption and wicket lost. They used the equation to scale up or down the score depending upon the batsmen ranking as the the batsmen rankings are static. They discussed how their method can be incorporate in different situation of the DL. In first case where rain interrupted the first innings of the game, score remains unchanged after game resumes in normal case researchers proposed that a small change should be incorporate before the game resume and this additional number of runs are calculated using the researchers equation. In second case if rain interrupted the first inning till innings break then researcher suggested to used the same approach applied in the first case. Researchers also explained the same approach for second innings interruption. They compared their results and found that their method improved the Duckworth Lewis method to some extent.

In this paper, Jaipal (2017) proposed a machine learning method to improve the Duckworth Lewis method. He compared the original version of Duckworth Lewis method with two methods which are Duckworth Lewis method with a run rate variable and improved version of Duckworth Lewis method. Researcher used the data from the cricsheet website. He discussed about the class imbalance problem in his data set. He calculated the specificity and sensitivity of the model to detect the class imbalance problem. Since sensitivity value was less than of specificity he concluded that there was a class imbalance problem in his data-set. He used the ROSE library in R to to under-sample his data-set to remove this problem. He implemented a models using random forest, decision tree, support vector machine, logistic regression, binomial logistic regression and neural network algorithm with run rate as a variable. He used the same algorithms to produce a models by removing the class imbalance problem and found an increase in accuracy of the models. He concluded his research by comparing the traditional Duckworth Lewis method models with Duckworth Lewis method as run rate variable and his improved version of Duckworth Lewis method. He concluded that his improved version of Duckworth Lewis method preforms better than tradition Duckworth Lewis method. He also found that logistic regression preformed very bad compared to the other models.

In this paper, Abbas and Haider (2019) proposed a method to optimized the Duckworth Lewis resource table. They also compared the accuracy of Duckworth Lewis method against supervised learning algorithm. They used the data-set from cricinfo website. Dataset consisted the summary of the matches. They calculated the Duckworth Lewis par score for each over and append a Duckworth Lewis prediction column in the dataset depending upon the par score. They applied the Duckworth Lewis to the matches which were not rain affected to check the accuracy of Duckworth Lewis. They used various classification algorithms such as Neural network and Naive Bayes on different stages of the match and compared the result of these classifiers with Duckworth Lewis prediction. They concluded that their accuracy is always better than the Duckworth Lewis prediction rate. They found that the Duckworth Lewis resource table is less accurate for first three wickets, so they optimised the values in resource for these columns by applying the Particle Swarm Optimization (PSO), which they developed on .NET. They compared the result of the modified Duckworth Lewis table with traditional Duckworth Lewis and found that accuracy was better for modified Duckworth Lewis Table. They also proposed a method to improve the Duckworth Lewis method in which they calculated the unpredictability index to find the patterns. They applied the Duckworth Lewis method at 40^{th} over for all teams and found four patters which were team winning while chasing, team lost while chasing, team won while defending and team lost while defending. They ranked each team depending on this patters and suggested it can be used along with Duckworth Lewis method to increase the accuracy.

In this paper, Koul et al. (2020) proposed a method to predict the score a cricket match using machine learning. They divided their method into two phase. In first phase they predicted the score of a match from the current situation. In second phase they built a model which will predict the win percentage of both team based on player selection. They used the IPL 2009 dataset to predict the runs that will score by a batsman using K means clustering algorithm. They implemented this method foe better strategy making and to analyze the performance of the players.

In this paper, Passi and Pandey (2018) proposed a method to increase prediction accuracy in the game of cricket using machine learning. In this method, they predicted the performance of the batsman and bowlers. They used the classification algorithms such as Naive Bayes, Random Forest, Multi-class SVM and Decision Tree to predict the performance of batsmen and bowlers. They used the data from cricinfo website and for predictive analytic they used tools like WEKA and Dataiku. They considered the attributes such as Number of innings, Batting Average, Strike Rate, number of Centuries, Number of Zeros, Highest scores for batsmen data. For bowler data they used the attributes such as Number of innings, Number of Overs, Bowling Average, Bowling Strike Rate, and Number of five wicket haul. They assigned weights to these parameters using analytic hierarchy process. Researchers used these attributes and theirs weights to know the parameters of the player such as consistency, form, performance against opposition team, performance on a particular venue using a mathematical equation. They also added other attributes in the data such as Batting Hand, Bowling Hand, Batting Position, Type of Match, Strength of Opposition, Toss. They found the class imbalance problem in their model so they used Supervised Minority Oversampling Technique to make all class equally distributed. They built different different models using the machine learning algorithms and concluded that Random Forest model predicted with an accuracy of 90% and as they increased the training data the accuracy of other models was increased.

In this paper, Dakhani and Maginmani (2020) proposed a system to predict the accuracy of players in the cricket using machine learning. They proposed this system for team selection which help to predict the player performance and will help the selectors to select best eleven players for the game. They proposed a system which store the performance of all players and this system will show the prediction of performance of every player to the captain and coach. Researchers used the WEKA tool to produce a predicting model using Random Forest and Naive Bayes Algorithm. They concluded that the Random Forest algorithm accuracy was higher compared to Naive Bayes.

In this paper, Wickramasinghe (2014) proposed a system to predict the performance of player in the Test cricket. Researcher developed a model to predict the performance using the characteristics of players, team strength and weight of match series. He used a linear hierarchical method to develop the model. He used the data of 5 years matches. In first level he used the individual performance data for a series. He formulated an equation for level model by using these factors and added a random term. In second level, researcher used the height and batting hand of the player in the equation. In third level, researcher used the rank of the team to formulate an equation. He combined all equations to predict the runs so scored. He compared the runs scored by player actually against the runs predicted by researcher's model. He concluded that height of the batsmen was very insignificant variable and his model preformed reasonably well with respect to the fact that it not easy to predict the performance of player in Test Cricket. He also discussed how runs were influenced by the home/away matches. He further discussed that due to emergence of T20 Cricket and One-Day Cricket players strike rate have been increased significantly.

In this paper, Shah et al. (2017) proposed a system to predict the player performance using factor analysis approach. They performed the Principal Component Analysis on IPL 9th edition, World Cup 2015 and IPL 2016 data. Research explained how PCA clustered the identical variables together. They used the Kaiser Criterion to select the groupings. They found that their model is influenced only by two factors after PCA. 62.50% variance of the model was explained by the batting factor and only 19.96% variance of the model was explained by the bowling factor. Researchers based on analysis results stated that batting performance dominate over bowling performance significantly. They also performed the factor analysis individually on World Cup 2015 data and found that the variance of model explained by the bowling factor increased which helped researchers to conclude that as the format becomes short, it will be difficult to explain the performance of player using bowling factor.

In this paper, Swetha and KN (2017) discussed the factors which decide the win. They discussed about the pitch how weather changes the nature of pitch. The pace of ball becomes slow on wet wicket whereas if grass in introduced on the pitch the wicket become fast. Hard pitches helps the batsmen to score more runs whereas the dry and dusty wickets help the bowler to claim more wickets. Further researchers discussed the toss factor while predicting the winner of the match. Third factor discussed by researchers was the team strength which is decided by balance of the squad and captain. Fourth factor was the past records, researchers discussed that past records can be important to predict the outcome of the match using couple of examples. Fifth factor was the home ground advantage and concluded that team always perform better in the home conditions. They also discussed about current performance which helps to predict the result of a match, which was the sixth factor. Last factor was weather and they also discussed about the impact of Duckworth Lewis method in result of the match. they concluded that these six factors which are external factors can be important in determining the result of the match.

In this paper, Anik et al. (2018) presented a method to predict players performance using machine learning algorithms such as linear regression, support vector machine with linear and polynomial kernel. They collected the data from espncricinfo.com and howstat.com. They included the features such as runs score, balls faced, number of boundaries, position, opposition and ground information. They discussed the over fitting problem and to avoid that problem they performed the feature selection process. They performed recursive feature selection and found that among eight features only five features are most important. Further, they performed feature selection using uni variate selection and got same result or same features selected as important in the dataset. Researchers removed variables which ere not important based on the feature selection methods. They implemented the linear regression model and support vector machine using linear and polynomial kernel. They concluded based on their analysis that support vector machine model with polynomial kernel performed bad because the data is linear. They also compared the score of batsman in actual match against the score predicted by their model and found that their accuracy was almost or above 80% for most of the players. Among three models they implemented, they concluded that support vector machine model with linear kernel preformed good for both batsmen as well as bowlers data.

In this paper, Somaskandhan et al. (2017) proposed a method to identify the optimal set of attributes that impose high impact on the end results of a cricket match using machine learning. They proposed this system for owners of the teams in IPL to enhance their winning ratio in the tournament. Their proposed solution depends on statistical analysis and machine learning. They collected the ball by ball data of IPL games from 2008 to 2019 and shrink the data as match summary. They calculated 23 features from ball by ball data such as average, strike rate, highest individual score, runs in power-play etc. They prepared the data innings wise and appended a win/loss column which suggested that whether a particular contribution by player in an inning leads to win or loss for the team. They applied classification algorithm on this data such as Navies Bayes and Support Vector machine. They used these classification algorithm to find the patterns and output of support vector machine presented a relative importance of attribute table. Researchers suggested that higher the value of importance attribute higher is its role in winning the matches. They concluded that High individual wickets, Number of bowled deliveries, number of thirties, total wickets, wickets in power-play, runs in death overs, dots in middle overs, number of fours and singles in middle overs are the most important features which impact the match result significantly.

In this paper, Manivannan and Kausik (2019) proposed a two method to predict the outcome of a match based on modeling of team performance and players performance. In first method they implemented the feature encoding. For team performance they used the support vector machine and fed the data related to players in team, match played on home ground, which helped them to get feature representation of a team without using classification. Further, researchers used the K Means clustering on the data to help model to classify the batsmen and bowler. They used the summation of distances derived from players metrics and obtained the team category relationship. They performed the feature encoding based on this relationship. Further researchers used the convolutional network and found that their proposed feature encoding learnt the features as well as it classified. They also implemented the ensemble approach where the prediction from 10 convolutional neural network models was averaged to get final prediction.

In this paper, Sachi et al. (2020) proposed a method to predict winner of Indian Premier League 2020 using data mining algorithm such as Decision Tree, Naive Bayes, Support vector machine and Random forest. They used Rapid Miner platform to produce and evaluate the models. They used the ball by ball data of IPL (2008 - 2019) and these algorithms to produce models. They used the produced models to predict which bowler will take most number of wickets? which team is most favourite? which team will win the IPL 2020 edition? They verified their model result and actual result and found that both the result were almost similar.

In this paper, Abedin et al. (2019) implemented a method to forecast the outcome of One-Day International Cricket. They produced model using Random forest, K-Nearest Neighbours, Support vector machine and Decision tree. Their data contained features such as previous match results, score of individual team, total wickets loss, number of score above 300 runs. They collected these data from ESPN and cricinfo website. They performed the feature selection task as number of features in the dataset was more and related to each other in term of batting and bowling. They used the highest scoring feature to produce the respective models. They compared the results of models produced using different machine learning algorithms on different type of distribution of train and test data. In their result they found that random forest model predicted the outcome of One-Day International matches with an accuracy of 92%. They also analyzed the trend of increase in accuracy as more train data is used.

In this paper, Agrawal et al. (2018) proposed a method to predict the results of Indian Premier League matches using machine learning. They collected the dataset of 500 IPL matches from techgig.com. The data consisted of ball by ball details of every IPL match along with 21 attributes. They divided their dataset into tow sub dataset. In first first sub dataset, they collected ball by ball details of every innings and in second subset of dataset they collected the results of ever match along with 14 other features. They also computed and appended the features such as average run rate, average strike rate and average play strike rate in first subset of dataset. They converted the categorical data into numerical format expect the wining attribute to avoid the problems in classification. Further researcher used machine learning algorithms such as Support vector machine, Naive Bayes and Ctree to produce classification models. They concluded that among three models, Naive Bayes model classified more matches correctly based on the result of confusion matrix.

In this paper, Barot et al. (2020) proposed a method to predict the winner of Indian Premier League matches and also proposed a novel analysis on batting and bowling. Their analysis showed that 55% percent of times teams winning the toss wins the matches. 57% of times team chasing a target becomes the winner. Target above 200 runs are rarely chased and their result analysed that 15.6% of times teams chased the target of 200 runs. They formulated batting index and bowling index for the players based on the batting average, batting strike rate, bowling average and bowling strike rate. They used the dataset and appended these feature on machine learning algorithms like Naive Bayes, Decision Tree and Logistic Regression to predict the outcome of IPL matches. Further, based on the confusion matrix, they concluded logistic regression predicted the outcome of IPL matches with an accuracy of 95%.

In this paper, Aburas et al. (2018) proposed a method for ICC Cricket World Cup prediction with help of machine learning algorithm and business intelligence techniques. They collected dataset from different sources such as ESPN cricinfo and Kaggle dataset. They designed the database and uploaded on the SQL database. They used R language and its libraries to retrieve the data sot red in the sql database. They used the label dataset for batsmen in which batsmen was labeled as Bad Batsmen, Good Batsmen and Elite Batsmen. They used the KNN classification to predict the batsmen type using this data. Further, they repeated the same process for bowlers dataset and all rounder dataset. Based on this classification they distributed points to all teams and the teams who had more elite players will have high chances of winning the world cup. Researcher concluded that England or India had same points after points distribution so both teams had fair chance of winning the world cup. Their model predicted the winner of world cup 2019 accurately.

In this paper, Teja et al. (2020) proposed a method for selecting cricket players using machine learning. They produced two separate models for batsmen and bowler by evaluating his statistics and using comparative approach to select the players. They scraped the data from espncricinfo.com and their data consisted the features such as names, matches, average, strike rate, not out, high score, innings, runs, 50's , 100's. Due to class imbalance problem, they used libraries like ROSE and SMOTE to perform oversampling on the dataset. Further researchers implemented various models for batsmen using machine algorithms like Random Forest, Ada Boost, Support vector machine, Light GBM, Linear Discriminant Analysis, Voting Classifier and Naive Bayes. Among these models, Light GBM, Random Forest and Ada Boost performed exceedingly well with an accuracy above 98%. They implemented the bowlers model using Catboost, Logistic Regression, Support vector machine and Naive Bayes algorithms. For bowlers dataset, they found that accuracy of support vector machine model was 82% which was highest among all. They integrated these models with web application for selecting players based on classification based on their previous statistics.

In this paper, Baboota and Kaur (2018) implemented the predictive model to forecast the football result using machine learning on English Premier League data. They used the data of 11 seasons and performed the feature engineering process in which they appended the features such as home/away factor, attack, midfield, and defence strength factor. They included the strength of attack, midfield and defence based on the ratings data from the FIFA database. After including these features researcher faced the problem of non Gaussian distribution of features over the dataset which would have affected the results of the model. Later they found that differential approach on feature will help to overcome this issue. They incorporated 33 other features in the dataset. Further researchers implemented the feature selection process since there were large number of features in the dataset. They divided their features int two class, first class contained the individual feature of home/away matches, second class contained the rest of differential features. Researcher did not performed the feature selection process for random forest and gradient boosting algorithm because these algorithm are designed to perform features selection on their own. Researchers addressed the problem of ternary classification while modeling. They used the machine learning algorithms such as Gaussian Naive Bayes, Support vector machine, Random Forest and Gradient Boosting. In their results they found that Gradient Boosting algorithm performed exceedingly well compared to other models. Researchers found that their test data contained majority of draw matches which is least likely outcome of the match, so they produced a model using radial basis function kernel over support vector machine and found that it predicted well compared to other models.

In this paper, Cornman et al. (2017) implemented the prediction of tennis match and betting using machine learning. They used the dataset available on GitHub which included the match statistics and betting data. They merged the data and performed feature engineering and selection process. In feature engineering they computed the player's form based on the performance in their last 5, 10, 20 matches. They calculated the head to head record against each player. After feature engineering they performed the feature selection process. They tried machine learning algorithm such as logistic regression, support vector machine, neural network and random forest. Based on the comparison of the results of these respective models they concluded that neural network model performed better with an accuracy of 73.5%. They developed betting model based on the output of random forest model. In their analysis of the results, they found that their model is always predicting the high ranked player and favoured player which suggested that their model is not learning the features of low ranked player and disfavoured player.

3 Research Methodology

From the above literature, we can clearly see that majority of the studies emphasised on the win/loss probability of the cricket matches using machine learning techniques. Their model is built on classification techniques which identified the batting and bowling strength of a team and depending on this strength they were able to classify the winning and losing team. We also referred the approach of increasing an accuracy of resource table of Duckworth Lewis table but still there is considerable margin of error is left in those methods. While referring to the previous studies in the section 2, we got to know that the external factors such as Toss, Pitch, Home/Away Ground, Batting average, Bowling Average, number of thirties, number of five wicket hauls can be important features to predict the score of the interrupted over. We have listed below the techniques which were used in the previous studies as it will help us to compare the techniques used along with the model accuracy's.

Table 2: A list of Methodology used in Section 2

References	Features Added	Method	Performance
Dakhani and Maginmani (2020)	Highest score by batsmen, Highest Wicket by bowlers	Decision Trees, Naive Bayes,Random Forest,Mutli-class Support Vector machine	Best model: Random Forest
Sachi et al. (2020)	Co-players, team and opposite teams are presented with their mathematical formulation	Decision Tree, Naïve Bayes, SVM – Support Vector Machine, Random Forest	Decision Tree(72.112%) Naïve Bayes(35%) Random Forest(82.73%)

Continued on next page

Table 2 – Continued from previous page

References	Features Added	Method	Performance
Barot et al. (2020)	Batting Index, Bowling Index	Naive Bayes, Logistic Regression,Support Vector Machine, Random Forest, Decision Tree	Naive Bayes(81.63%) SVM(83.67%) Decision Tree(87.75%) Random Forest(83.67%) Logistic Regression(95.91%)
Teja et al. (2020)	High-Score, Innings, 50's, 100's	Support Vector Machines, Naïve Bayes, Random forest, Linear Discriminant Analysis, Voting Classifier, CatBoost, Logistic Regression	**Batting:** Random Forest(98.68%) SVM(88.15%) Linear Discriminant Analysis(82.89%) Voting Classifier,(97.64%) Naive Bayes(84.21%) **Bowling:** Catboost(59.15%) Logistic Regression(57.34%) SVM(82.07%) Naive Bayes(80.76%)
Abbas and Haider (2019)	NA	Bagging with Naïve Bayes, Neural Network, Naïve Bayes	Bagging with naïve bayes(80.97%) Neural Networks(82.80%) Naïve Bayes(87.45%)
Manivannan and Kausik (2019)	Classification of batting bowling and all rounders	Convolutional Neural Networks	CNN(70%)
Abedin et al. (2019)	Toss, Won by how many runs, Venue, Match schedule	Random Forest, K-Nearest Neighbour, SVM, Decision Tree	Random Forest(91.83%) KNN(71.4%) SVM(88.4%) Decision Tree(88.4%)
Passi and Pandey (2018)	Batting-Average, Strike Rate, Century, Zeros, Highest Score, Bowling-Average, Bowling Strike Rate, Five Wicket Haul	Naïve Bayes, Random Forest, Multi-class SVM, Decision Trees	**Batting:** Naive Bayes(42.47%) Decision Tree(79.38%) Random Forest(90.67%) SVM(50.88%) **Bowling:** Naive Bayes(57.48%) Decision Tree(85.99%) Random Forest(91.80%) SVM(68.35%)
Anik et al. (2018)	Pitch Condition	Linear Regression, SVM Linear Kernel	Linear Regression(89.6%) SVM Linear Kernel(75%)
Agrawal et al. (2018)	Batting Average, Bowling Average, Power-play run-rate	Support Vector Machine, CTree and Naive Bayes	Naïve Bayes(98.98%) SVM(95.96%) Ctree(97.97%)
Aburas et al. (2018)	Bad, Good, Elite Players	KNN	Classified Players

Continued on next page

Table 2 – Continued from previous page

References	Features Added	Method	Performance
Baboota and Kaur (2018)	Attack, Midfield, Defence Strength	Naïve Bayes, RBF SVM, Random Forest, Gradient Boosting	**Ranked Probability Scores:** Naive Bayes(0.23) RBF SVM(0.180) Random Forest(0.175) Gradient Boosting(0.173)
Jaipal (2017)	NA	Random Forest, SVM, Decision Tree, Logistic Regression, Multinomial logistic regression, Neural Network	SVM(88%) Logistic Regression(47%) Multinomial Logistic Regression(88%) Decision Tree(87%) Random Forest(95%) Neural Network(49%)
Somaskandhan et al. (2017)	Total runs scored in an innings, Total number of wickets in an innings, Highest individual score in an innings, Runs in the power-play of the innings	Extra Tree, Naïve Bayes, Support Vector Machine	SVM(80%)
Cornman et al. (2017)	Average of each match statistic over the most recent 5, 10, and 20 matches. Players head to head record against their opponent	Logistic Regression, SVM, Neural Network, Random Forest	Most of the time, the model is predicting the higher ranked player and the favored player
Sankaranarayanan et al. (2014)	Batting Average, Strike Rate, Home-run hitting ability, Milestone reaching ability	Ridge regression, Attribute bagging, Nearest neighbors	Overall(70%)

By referring the table 2 and section 2, we can clearly see that majority of research used the random forest function, naive bayes and support vector machine to determine the outcome of the matches. Referring this studies helped us to verify the class imbalance problem at early and knowledge of suitable method that can be used to remove this problem. Also, we got to know various features which is added in the dataset and have significant impact on the model and these factor might be useful for us to obtain more accurate result for our model.

3.1 Feature Engineering

As we saw earlier in the Section 2, few studies have used the feature engineering technique in their work. Feature engineering is a process that uses the domain knowledge of data to create features that help machine learning algorithm to work better. In our study, we will need to compute the run rate after every over which will be our key feature in order to achieve our target. Further we can add features like batting average, bowling average, number of times team surpassing 200 runs score, number of thirties, number of five wicket hauls. These features will help to predict the scoring rate and wickets lost rate.

4 Design Specification

Cross-industry standard process for data mining (CRISP-DM) approach will help to do feature engineering and feature selection process as well as to understand the data well. Shearer (2000) in his article described the CRISP-DM process. It is now accepted as open standard for developing data mining and knowledge discovery project. This framework has six steps with data at the center. It starts with business understating, this step help to identify the problem which is to be solved. In our case, our first problem is to solve the bias nature of Duckworth Lewis method. Further after evaluating the model, we might face a few problems that can be related to accuracy of the model. Once we got the business understanding, we will know the type of data which will be required. After data understanding, we can start processing the data in which we will clean the data, add features into the data, feature selection process which will keep important variables in the data. Almost more than half percent of the time is consumed in first three steps of this framework because it is important to have right data ready for modeling step. In the modeling step, we will prepare a analytical model and it has three types of model that supervised models, unsupervised models. After feeding the data and generating the model, we will need to evaluate the model. This will help to identify that whether our model performed as expected or under-performed. It helps to identify whether there is any redundancy in the data or helps to identify whether data preparation was done correctly or not. Once we got the evaluation that we were expecting then model is ready for the deployment.

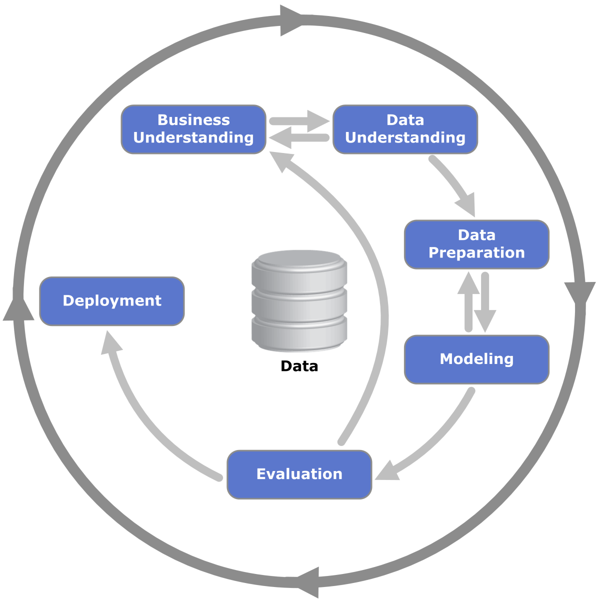


Figure 2: CRISP-DM Framework

4.1 Data Gathering

Our project will based on the IPL 2020 data. Indian Premier League is a franchise based T20 league which is conducted every year in the month of April and May. Data will be collected from the Kaggle which is a crowd source platform and the original source of data is cricsheet.org. This dataset contains two files, first file contains match by match data and second file contains the ball by ball data of every match. Match by match dataset contains the match summary details like Ground information, player of the match, city, team details, toss winner, toss decision, result information, umpires information. Ball by ball data contains the information like inning's number, over number, number of ball, batsmen and bowler information, batsmen runs, total runs, extra's runs, wicket, type of dismissal, dismissed player and fielder information. Ball by ball dataset consists of total 14,475 records and match by match dataset contains 60 records since 60 matches are played in a single season.

4.2 Data Cleaning

There are variables like dismissal type, extra's information, city, ground, toss winner, toss decision and match winner information which are in string format. We will need to convert this variables into categorical variables. Further will need to clean out the matches which got terminated due to rain or any interruption since we will need to train the dataset which contains whole 20 overs information in order to predict runs from any stage of match.

Dataset Link: `https://www.kaggle.com/patrickb1912/ipl-complete-dataset-20082020`

4.3 Data Mining

Data mining refers to extracting knowledge from large amount of the data to predict the future trend or result. To construct a model, we must arrange the data in such a way that certain parts of the data are special for prediction. This arrangement of the data is executed using hold out cross validation or k-fold cross validation techniques Reitermanov´ (2010). Hold out cross validation splits data into three parts that are training, testing and validation. Validation data is used to evaluate the model performance during training. It helps to avoid the over fitting problem. Test data is used to evaluate the model performance after training. Hold out cross validation is majorly used for large dataset. After splitting the data, it is ready to build a model. After reviewing all past studies in section 2, we decided to use the regression techniques for this problem as it is more suitable for this problem.

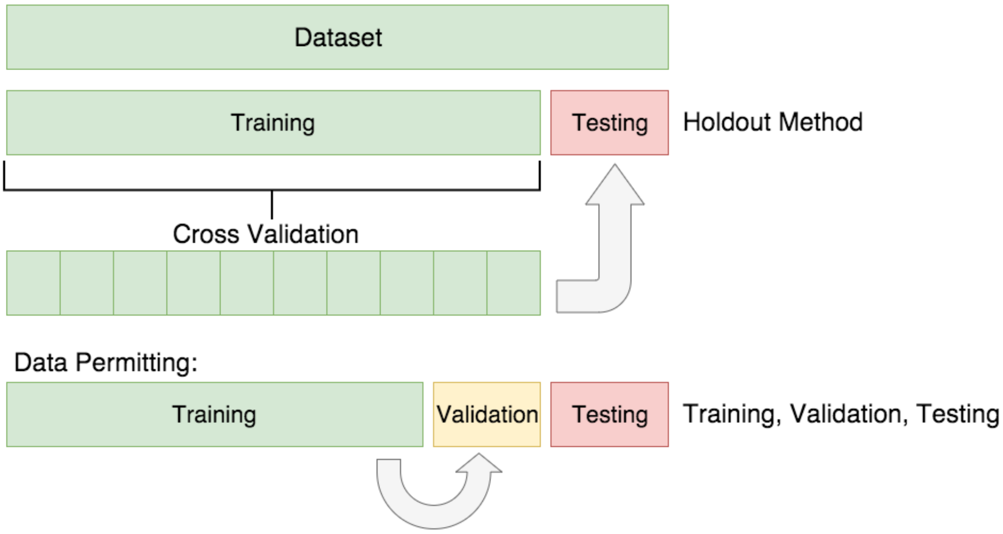


Figure 3: Cross Validation

4.4 Regression Models

4.4.1 Ada Boost

In this research, we will be using Ada boost algorithm and will be our novel technique. As we see in the Section 2, most of the researchers used svm, naive bayes and random forest but ada-boost will help us to produced enhanced method because it is an ensemble method. Ada-boost initially assign weights to all records equally. After the first iteration or after the first learner is generated, if any miss-classified record is detected then the weights of that record is increased and weights of rest of the records is decreased. By increasing the weight of this record, algorithm will try to classify or next weak learner will try to classify these record correctly Schapire (2013). Since multiple learners are generated in this algorithm and current model is adapted from the the previous model so this technique is called Adaptive Boosting Algorithm.

4.4.2 Gradient Boosting

In this research, gradient boosting algorithm will be the second novel technique. In first step this algorithm produce an average model and output of the model will be compared to the actual output and error will computed. This error will be passed to a decision tree model and these residual models will be fitted on the residual error to optimise the model. Residual models are created till error become zero or minimal Helmbold (2002). In gradient boosting a loss is used to input for next model and it goes on.

4.4.3 Random Forest

In this research random forest will be used to enhance the result of the model. It is an ensemble method in which decision tree are built using the bagging techniques. In random forest, subsets of the data is created which are called as bags. After creating bag, decision tree will be fitted on each bag and in this way training is implemented Fawagreh et al. (2014). While testing, each bag's decision tree will give a output and mean of this output is considered as the final output for random forest regression algorithm. It helps to reduce the variance of decision tree.

4.5 Proposed Evaluation Metrics to Measure Performance of The Models

We will be using the following metrics to evaluate the performance of the model. Model performance is determined by difference between the actual output and predict output. Variance is one of the most important factor which helps to find whether generated model will perform good or not.

4.5.1 Mean-Squared Error

Mean squared error calculates the average squared error between actual and predicted data. The advantage of this metrics is that it considers positive and negative variance. It's drawback is that it gives more weight to higher values and other drawback is it's evaluation with different units.

$$MSE = \frac{1}{\mathrm{n}} \sum\nolimits_{i=1}^{n} (observed - predicted)^2 \tag{3}$$

4.5.2 Root-Mean-Squared Error

Root Mean squared error calculates the square root of average squared error between actual and predicted data. The advantage of this metric is that unit of error is same as target which was drawback of Mean Squared Error. The disadvantage of Root Mean Squared Error is that it increases than Mean Absolute Error as the sample size increases. This metric is suitable when outliers are less.

$$RMSE = \sqrt{\sum\nolimits_{i=1}^{n} \frac{\left(x_{predicted,i} - x_{measured,i}\right)^2}{n}} \tag{4}$$

4.5.3 Mean Absolute Error

Mean Absolute Error is average of absolute or positive error of all the values. The disadvantage of this metric is that it fails to punish large errors and also it does not indicate direction of the errors. The metrics is useful when overall impact is proportionate to actual increase in error.

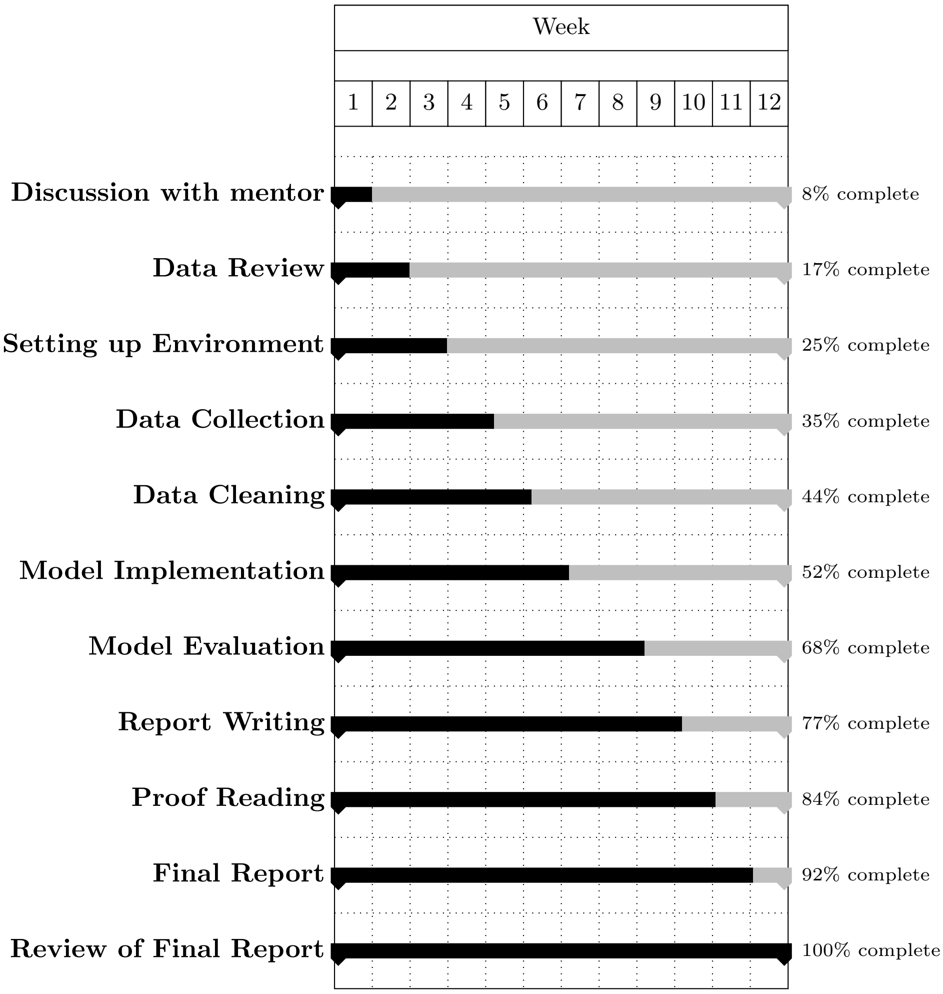
$$MAE = \frac{1}{n} \sum_{t=1}^{n} |e_t| \tag{5}$$

4.5.4 Adjusted $\mathbf{R^2}$

It is a metric that represent proportion of variance for a dependent variable that is explained by independent variables. Adjusted R-square metric is improved version of R-square metric. The problem with the R-square metrics is that it keep on increases or remain unchanged whenever a new variable is added in the the model irrespective of its impact. Adjusted R square impose the penalty if new variable introduced does not have an impact.

4.6 Project Plan

Below Gantt chart shows the overview of a plan of the project which will be going to be carried out in next semester.



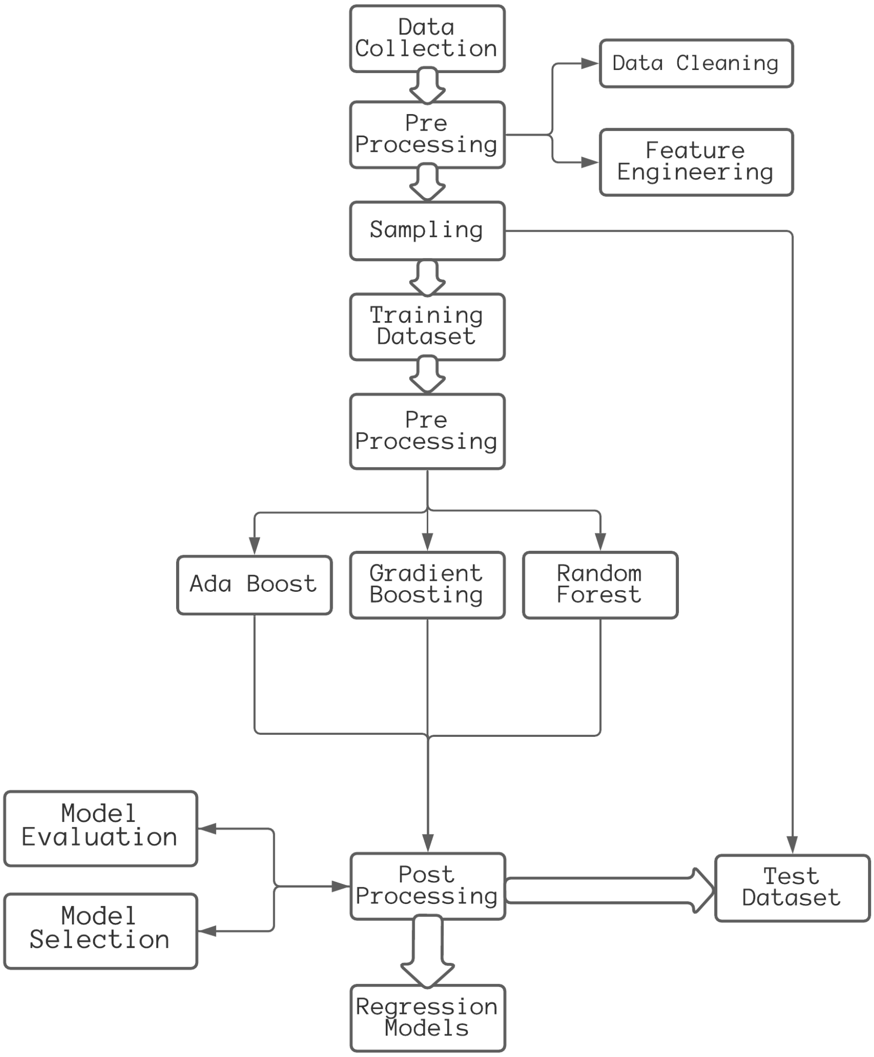


Figure 4: Flowchart of proposed research

5 Conclusion

After reviewing all papers from literature review we can say that few machine learning models are used to predict the score of match after an interruption but still some of the machine learning models are not used yet. By using our proposed machine learning models we can try to improve the performance and we can achieve the score which will more suitable for T20 format of cricket. We will be performing feature engineering process and planning to use machine learning algorithms like Adaptive Boosting, Gradient Boosting and Random Forest. This algorithms will lead us to predict the score of interrupted overs which we will compare with the score that is derived from the Duckworth Lewis Rule.

References

Abbas, K. and Haider, S. (2019). Duckworth-lewis-stern method comparison with machine learning approach, *International Conference on Frontiers of Information Technology (FIT)* pp. 197–1975.

Abedin, M. M., Urmi, S. R., Mozumder, M. T. I., Rahman, M. S. and Firoze, A. (2019). Forecasting the outcome of the next odi cricket matches to be played, *International Journal of Recent Technology and Engineering (IJRTE)* **8**(4): 10269–10273.

Aburas, A. A., Mehtab, M. and Mehtab, Y. (2018). Icc world cup prediction based data analytics and business intelligent (bi) techniques, *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* pp. 273–276.

Agrawal, S., Singh, S. P. and Sharma, J. K. (2018). Predicting results of indian premier league t-20 matches using machine learning, *8th International Conference on Communication Systems and Network Technologies (CSNT)* pp. 67–71.

Ali, F. and Kusro, S. (2018). Player ranking: A solution to the duckworth/ lewis method problems, *International Conference on Emerging Technologies (ICET)* pp. 1–4.

Anik, A. I., Yeaser, S., Hossain, A. G. M. I. and Chakrabarty, A. (2018). Player's performance prediction in odi cricket using machine learning algorithms, *4th International Conference on Electrical Engineering and Information Communication Technology (iCEEiCT)* pp. 500–505.

Baboota, R. and Kaur, H. (2018). Predictive analysis and modelling football results using machine learning approach for english premier league, *International Journal of Forecasting* **35**(2): 741–755.

Barot, H., Bide, A. K. P., Ahir, B. and Kankaria, R. (2020). Analysis and prediction for the indian premier league, *International Conference for Emerging Technology (INCET)* pp. 1–7.

Bhattacharya, R., Gill, P. S. and Swartz, T. B. (2011). Duckworth–lewis and twenty20 cricket, *Journal of the Operational Research Society* **62**(11): 1951–1957.

Cornman, A., Spellman, G. and Wright, D. (2017). Machine learning for professional tennis match prediction and betting.

Dakhani, M. and Maginmani, U. H. (2020). Predicting accuracy of players in the cricket using machine learning, *International Research Journal of Engineering and Technology (IRJET)* **7**(5).

Fawagreh, K., Gaber, M. M. and Elyan, E. (2014). Random forests: from early developments to recent advancements, *Systems Science & Control Engineering* **2**(1): 602–609.

Helmbold, N. D. . D. (2002). Boosting methods for regression, *Machine Learning* **47**: 153–200.

Jaipal, M. S. (2017). Improving duckworth lewis method by using machine learning.

Koul, N., Adhav, K., Dixit, A. and Pakhare, R. (2020). Predicting cricket score by using machine learning concepts, *International Journal of Grid and Distributed Computing* **13**(1): 2396– 2401.

Manivannan, S. and Kausik, M. (2019). Convolutional neural network and feature encoding for predicting the outcome of cricket matches, *14th Conference on Industrial and Information Systems (ICIIS)* pp. 344–349.

Mchale, I. and Asif, M. (2013). A modified duckworth–lewis method for adjusting targets in interrupted limited overs cricket, *European Journal of Operational Research* **225**(2): 353–362.

Passi, K. and Pandey, N. (2018). Increased prediction accuracy in the game of cricket using machine learning, *International Journal of Data Mining and Knowledge Management Process (IJDKP)* **8**(2).

Phanse, V. and Deorah, S. (2011). Evaluation and extension to the duckworth lewis method: A dual application of data mining techniques, *IEEE 11th International Conference on Data Mining Workshops* pp. 763–770.

Preston, I. and Thomas, J. (2002). Rain rules for limited overs cricket and probabilities of victory, *Journal of the Royal Statistical Society* **51**(2): 189–202.

Reitermanov´, Z. (2010). Data splitting, *WDS'10 Proceedings of Contributed Papers* **1**: 31–36.

Sachi, P., K, V. and Iyer, K. B. P. (2020). Prediction of indian premier league-ipl 2020 using data mining algorithms, *International Journal for Research in Applied Science and Engineering Technology (IJRASET)* **8**(2): 790–795.

Sambasivarao, S., Reddy, T. and Ramu, P. (2014). A method for resetting the target in interrupted twenty20 cricket match, *Journal of Physical Education and Sport Science* **2**: 226–234.

Sankaranarayanan, V. V., Sattar, J. and Lakshmanan, L. V. S. (2014). Auto-play: A data mining approach to odi cricket simulation and prediction, *Proceedings of the 2014 SIAM International Conference on Data Mining* pp. 1064–1072.

Schapire, R. E. (2013). Explaining adaboost, *Springer* **1**: 37–52.

Shah, S., Hazarika, P. J. and Hazarika, J. (2017). A study on performance of cricket players using factor analysis approach, *International Journal of Advanced Research in Computer Science* **8**(3).

Shearer, C. (2000). The crisp-dm model: The new blueprint for data mining, *Journal of Data Warehousing* **5**(4): 13–22.

Somaskandhan, P., Wijesinghe, N., Bashitha, L., Wijegunawardana, Bandaranayake, A. and Deegalla, S. (2017). Identifying the optimal set of attributes that impose high impact on the end results of a cricket match using machine learning, *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)* pp. 1–6.

Steven, S. (2016). The duckworth-lewis-stern method: extending the duckworth-lewis methodology to deal with modern scoring rates, *Journal of the Operational Research Society* **67**(12): 1469–1480.

Swetha and KN, S. (2017). Analysis on attributes deciding cricket winning, *International Research Journal of Engineering and Technology (IRJET)* **4**(3).

Teja, I. S. R., Kalyan, T. P., Reddy, V. A. K. and Sagar, P. V. (2020). Cricket player selection using machine learning, *International Journal of Engineering and Advanced Technology (IJEAT)* **9**(5): 68–71.

Wickramasinghe, I. (2014). Predicting the performance of batsmen in test cricket, *Journal of Human Sport and Exercise* **9**: 744–751.