VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computing Science and Engineering

VIT Chennai

Vandalur - Kelambakkam Road, Chennai - 600 127

Final Project Report

Programme: B.Tech - CSE

Course: CSE2004 – Database Management Systems

Slot: G1

Faculty: Prof. M. Premalatha

Component: J

TITLE: PREDICTION OF TOTAL RUNS IN AN IPL T20 MATCH USING REGRESSION

TEAM MEMBERS:

• ANTHRA DEVARAJAN - 19BCE1681

• MAKESH SRINIVASAN - 19BCE1717

• SABHARI GIRISH - 19BCE1759

• V RAGHAV ANAND - 19BCE1415

ABSTRACT:

Cricket, usually called the gentleman's game, is already a game of uncertainties, but when external factors outside the game influence it, it becomes even more unpredictable and can end up spoiling the charm of the sport. A lot of research has been done on the area of sports with predictions. In a game like cricket, where weather plays a very important role, the D/L method that is being used in rain affected matches has its own shortcomings.

Teams on the probability of being in the winning side before the game end up being in the losing side at times when any external factor like rain interrupts the match and D/L method comes up with its own predictions of targets to be chased down which may/may not be in the favor of the batting side, or the bowling side.

The calculations involved in setting the target by D/L method do take care of many important factors, however, it still does not give a fair outcome for one of the teams as one of the teams don't get to play the same number of overs the other team did, due to the external factor.

It may never be accurately possible to determine a second-innings target affected by rain, but, by considering few more factors such as the style of how a team goes about in power-play overs, the strength of the team etc., we can add more precision to the target to be predicted for the team to chase.

This project intends to provide analysis on which team has the upper hand at any stage of the innings, and also get the target to be chased down in the second innings when an external factor affects the game. By using the concept of data mining, Python programming language as the front end and excel as the back end where certain necessary datasets are to be stored, the output of the project is to predict the number of runs that can be scored by a team when there is any factor that interrupts the game (such as rain) and to give a fair probability of which side has more probability of winning a game based on certain calculations from the existing dataset which we're about to see.

INTRODUCTION:

Cricket is a bat-and-ball game played between two teams of eleven players on a field at the centre of which is a 20-metre (22-yard) pitch with a wicket at each end, each comprising two bails balanced on three stumps. The batting side scores runs by striking the ball bowled at the wicket with the bat, while the bowling and fielding side tries to prevent this and dismiss each player.. Means of dismissal include being bowled, when the ball hits the stumps and dislodges the bails, and by the fielding side catching the ball after it is hit by the bat, but before it hits the ground. When ten players have been dismissed, the innings ends and the teams swap roles. The game is adjudicated by two umpires, aided by a third umpire and match referee in international matches. They communicate with two off-field scorers who record the match's statistical information.

There are various formats ranging from Twenty20, played over a few hours with each team batting for a single innings of 20 overs, to Test matches, played over five days with unlimited overs and the teams each batting for two innings of unlimited length. Traditionally cricketers play in all-white kit, but in limited overs cricket they wear club or team colors.

Also, during the initial days, since cricket is an outdoor sport, climatic influences such as rain severely affected the outcome of the game and hence to avoid its interference, the Duckworth-Lewis method was introduced which predicted the score to be chased down by the batsmen batting second in case if rain interfered with the game. However, the method has its own shortcomings as it is done by certain calculations that may become unfair to the team batting second, or, at times the team that already batted in the first innings. The DLS method is an attempt to set a statistically fair target for the second team's innings, which is the same difficulty as the original target. The basic principle is that each team in a limited-overs match has two resources available with which to, and the target is adjusted proportionally to the change in the combination of these two resources.

Various methods had been used previously to resolve rain-affected cricket games, with the most common being the Average Run Rate method and the Most Productive Overs method. These earlier methods, while simple in nature, had intrinsic flaws that meant they produced unfair revised targets that altered the balance of the match, and were easily exploitable:

• The Average Run Rate method took no account of how many wickets were lost by the team batting second, but simply reflected how quickly they were scoring when the match was interrupted. So, if a team felt a rain stoppage was likely, they could attempt to force the scoring rate without regard for the corresponding highly likely loss of wickets, skewing the comparison with the first team.

• The Most Productive Overs method also took no account of wickets lost by the team batting second, and effectively penalised the team batting second for good bowling by ignoring their best overs in setting the revised target. The strength of the team is one of an important factor. If a team has more depth in batting than the other, then the probability of that team scoring in the back end overs is more than the other team. Hence the team with more depth in batting should have a slight upper hand while determining the winning percentage of a team in a match.

RELATED WORK:

JOURNAL 1:

METHODOLOGY:

Average run rate (ARR):
The winning team is decided by the higher average number of runs per over that each team has had the opportunity to receive. It is a simple calculation but the method's major problem is that it very frequently alters the balance of the match, usually in favor of the team batting second

Most productive overs (MPO).
The target is determined for the overs the team batting second (Team 2) are to receive by totaling the same number of the highest scoring overs of Team 1. The process of determining the target involves substantial bookwork for match officials and the scoring pattern for Team 1 is a criterion in deciding the winner. We believe that it is only Team l's total that should be used in setting the target and not the way by which it was obtained. The method strongly tends to favor Team 1.

Discounted most productive overs (DMPO) .
The total from the most productive overs is discounted by 0.5% for each over lost.

This reduces slightly the advantage MPO gives to Team 1 but it still has the same intrinsic weaknesses of that method.

Parabola (PARAB) D.

This method, by a young South African , calculates a table of norms y,
for overs of an innings, x, using the parabola y = 7.46x - O.059x2 to model, rather inappropriately since it has a turning point (at about 63 overs, the diminishing returns nature of the relation- ship between average total runs scored and total number of overs available. The method is an improvement upon ARR but takes no account of the stage of the innings at which the overs are lost or of the number of wickets that have fallen.

DUCKWORTH LEWIS METHOD:

The average total score Z (u) which is obtained in u overs may be described by the exponential equation

$$Z(u) = Zo [1 - \exp(-bu)] \quad (1)$$

Where Zo is the asymptotic average total score in unlimited overs (but under one-day rules) and b is the exponential decay constant.

The next stage of development of a suitable two-factor relationship is to revise (1) for when w wickets have already been lost but u overs are still left to be received. The asymptote will be lower and the decay constant will be higher and both will be functions of w. The revised relationship is of the form

$$Z(u, w) = Zo(w) [1 - \exp\{- b(w) u\}] \quad (2)$$

where Zo(w) is the asymptotic average total score from the last 10-w wickets in unlimited overs and b(w) is the exponential decay constant, both of which depend on the number of wickets already lost hence w= 0..9.

They have been obtained following extensive research and experimentation so that Z (u, w) and its first partial derivative with respect to u behave as expected under various practical situations and give sensible results at the boundaries.

In this paper they have explained the mechanisms of other methods used for resetting target scores in interrupted one- day cricket matches. Each of these methods yields a fair target in some situations. None has proved satisfactory in deriving a fair target under all circumstances. We have explained a DL method which gives a fair revised target score under all circumstances. This is based on the recognition that teams have two resources, overs to be faced and wickets in hand, to enable them to make as many runs as they can or need. Although this method is little bit complex for fans who are not person of mathematical field. But overall this method is quite satisfactory by ignoring some odd cases.

JOURNAL-2:

PITCH: Unlike other sports, cricket stadium's size and shape is not fixed except the dimensions of the inner circle and pitch which are 30 yards and 22 yards respectively. Outfield variations and pitch can have a substantiate effect on bowling and batting. The spin of the ball, seam movement and the bounce depends on the nature of the pitch. It depends on how wet is the pitch. The more wet the pitch, the slower it will play. On the off chance that it is drying out, those balls will change significantly, yet all it will get less difficult those drier it gets

TOSS: According to cricket analysts, there is sure measure of advantage for a team if it wins the toss. This might not be deciding factor in a match but it would give the team the opportunity of choosing "what they want".

Home Ground Advantage: This is another attribute which determines the winner in the match. If you are playing in the home ground conditions everything would be in your hands like climatic factors, pitch nature and major role is played by the home crowd.

PROBLEM STATEMENT

To design a system to predict the score of a team after 1 st and 2 nd innings. This can be done by considering multiple parameters like winning toss, batting side, DL approach, Home ground advantages, player wise performance etc. To solve this problem we have collected the historical data of all some team like (India , South Africa, Australia , New Zealand etc.), and using prediction algorithm like Naive Bayesian algorithm we are predicting the best starting players for both the teams that can be used in fantasy league for winning the maximum points

METHODOLOGY

They have followed 5 methodology in the course of my project. The methodology consists of 5 different phases i.e. Data Set Generation, Data Cleaning, Attribute Selection, Data Mining and Analysis of Results.

The data was collected from the http://cricsheet.org [1] website. The website has data about all 8 seasons (from 2008 to 2016) of domestic tournaments held in India i.e. the Indian Premier League.

They used the Java classes File and the FileWriter to read the .csv files and write the contents to a new file

ALGORITHM:

1: Q-leaning approach

Algorithm 1: The function implementing the tabular TD(0) algorithm. This function must be called after each transition. Function TD0 (X;R; Y; P; V)

2: Naïve Bayes Classification

Input: User input file data record which contains {C Score, Overs, WF}, segment of average score from train database of k to n over's.

Output: Projected score

The project thus, aims not only to attract more users to this game that is Fantasy Cricket, but also aims at improving the general attraction to the Premier League. And thus predicting the total score at the end of the 1st and 2nd innings.

JOURNAL-3:

The major contribution of their paper was:

1) To predict the winner of ODI cricket matches, we propose a novel dynamic approach to reflect the changes in player combinations.

2) Using machine learning supervised learning algorithms to predict the outcome of the matches.

3) various models in order to check which model has better efficiency and which algorithm gives the accurate or best results.

METHODOLOGY:

They used four supervised machine learning algorithms. And they have downloaded Datasets from Espncricinfo which has 5000 records of Data. They used supervised learning algorithm because Supervised learning is where you have input factors (x) and a output factor (Y) and you utilize an algorithm to learn the mapping function from the input to the output.

ALGORITHMS

They used Logistic regression, support vector machine, Decision tree, Bayes point machine binary classification model.

DECISION TREE: A decision tree is a decision help device that uses a tree-like chart or model of decision and their conceivable results, including possible outcomes, asset expenses, and utility.

SUPPORT VECTOR SYSTEM: A support vector machine is a supervised machine learning algorithm which is used in classification or regression environment . The support vector clustering applies statistics of support vectors, created in the support vector machines calculation, to arrange un labeled information, and is a standout among the most generally utilized clustering calculations in mechanical applications.

LOGISTIC REGRESSION: It's a classification supervised learning algorithm. It is utilized to foresee a binary result (1/0, Yes/No, True/False) given an arrangement of autonomous factors. It enables one to state that presence of a risk factor expands the chances of a given result by a specific factor. The model is a probability model and not a classifier.

BAYES POINT CLASSIFICATION MODEL: This algorithm productively approximates the theoretically ideal Bayesian average of linear classifiers (regarding speculation execution) by picking one "normal" classifier, the Bayes Point. Since the Bayes Point Machine is a

Bayesian grouping model, it is inclined to over fitting to the training information.

Microsoft Azure ML cloud: This I not the algorithm used by them. This is a software/tool used by them to train the models and to obtain the accuracy of the mentioned models

REQUIREMENT ANALYSIS:

Non Functional Requirements

Efficiency: Using data mining models are very fast as data mining and machine learning are the new technologies which are used to predict some kind of output with the given data sets as inputs.

Reliability: It is reliable until and unless the data is so huge and complicated and is difficult to process. For the large raw data we have to implement big data to convert that raw data in information.

Usability: It can be used in predictions of ODI cricket matches as they will help the coaches

and the players to analyze the area of improvement.

Functional Requirements

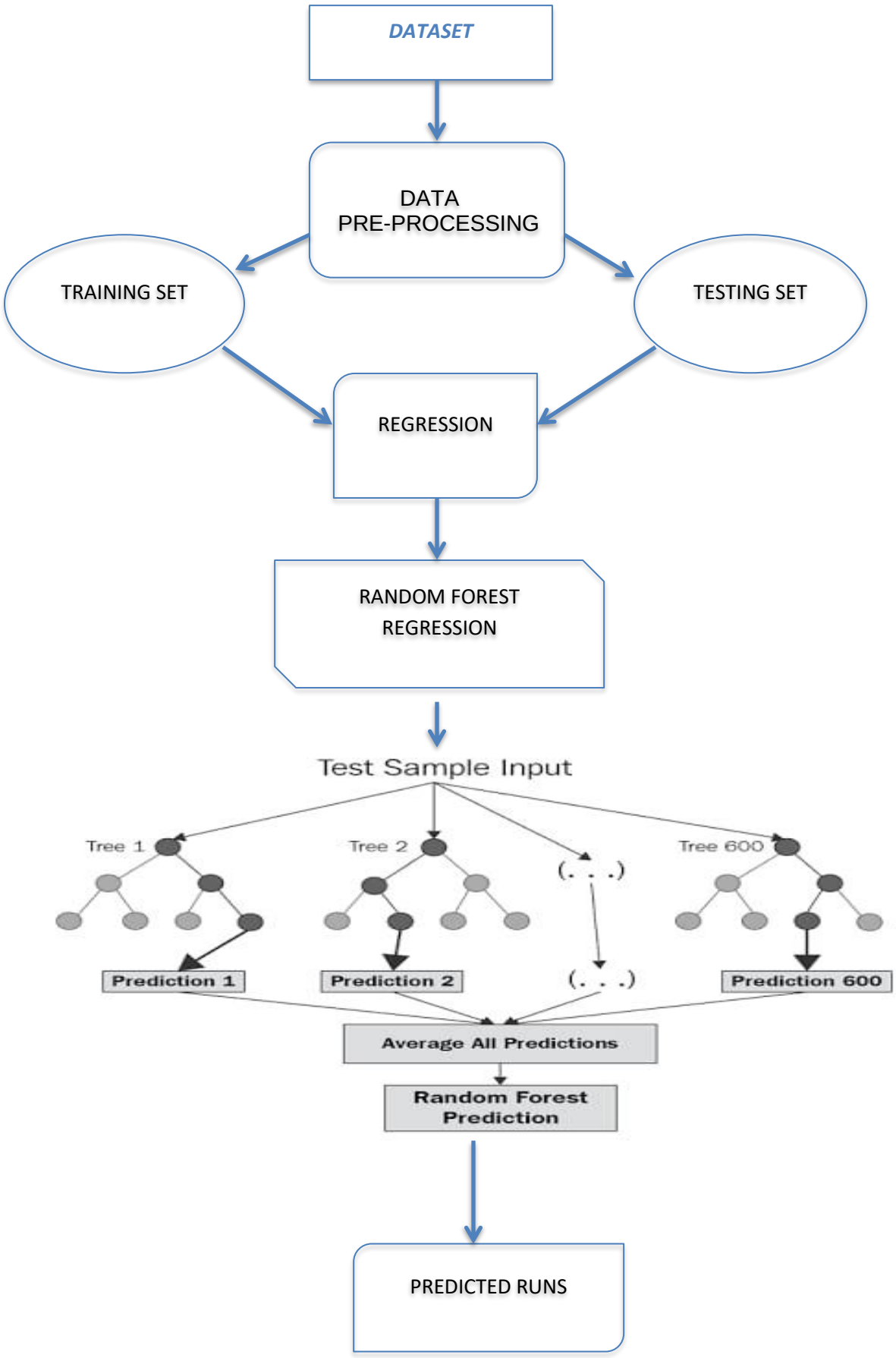
User characteristics: The user is just need to select the two teams to predict the winning team.

Assumptions & Dependencies: This mainly depends on the size of the data that is to be analyzed and the type of the model been implemented.

The paper addresses the problem of anticipating the result of an ODI cricket match utilizing the insights of 5000 matches. The curiosity of our approach lies in addressing to the issue as a dynamic one, and utilizing the results of the previous matches as the key element in predicting the winner of the match. We watch that simple highlights can yield exceptionally encouraging results. Predicting the winner of the matches using different supervised algorithms is been achieved and now we can predict the upcoming matches. There might be some more algorithms coming in future which give better results then used in this paper. But the best part is we are now predicting the future events just by using the technology and the computers.

METHODOLOGY:

Let us first have a look at the architecture diagram of the project:


DATASET
DATA
PRE-PROCESSING
TRAINING SET
TESTING SET
REGRESSION
RANDOM FOREST
REGRESSION
Test Sample Input
Tree 1
Tree 2
(. . .)
Tree 600
Prediction 1
Prediction 2
(. . .)
Prediction 600
Average All Predictions
Random Forest
Prediction
PREDICTED RUNS

DATASET:

- Match_id- Every match is given a unique match_id
- Batting team- The team that is batting
- Bowling Team- The team that is bowling
- Balls remaining- The number of balls left
- Current score- The score at each point in time
- Total score- The final runs scored by the batting team
- Wickets- The number of wickets lost at a point
- Home- check for if it's the home-ground of the team
- Toss- Check for whether the toss was won
- Diff- Difference between the ranks of the teams
- Runrate- The runrate for each ball
- Partnership- The runs scored by the striker and the non-striker before a wicket falls
- Runs per wicket- The runs the team has scored with each wicket that they have lost

ALGORITHM AND METHODOLOGY:

```
import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.externals import joblib
```

All these library files mentioned above are imported and we pre-process the data for the training set and the test set from the dataset files 'the end.csv' and 'testcasei1_4.csv'. Adter that, we consider the attributes that we want the training set and test set to have and then replace all the team names with a respective team id(for easier manipulation).

```
# Data frame is created

X = training_set.iloc[:, [0, 1, 2, 4, 5, 6, 7, 8]].values

y = training_set.iloc[:, 3].values


# Splitting the data set into the Training set and Test set
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05, random_state=0)
```

The data frame for X, Y and their respective training and test sets are created by splitting the data randomly and feature scaling is done.
For training the model, we use RandomforestRegressor and store the output obtained in a '.pkl' file to save runtime and occupy lesser space.

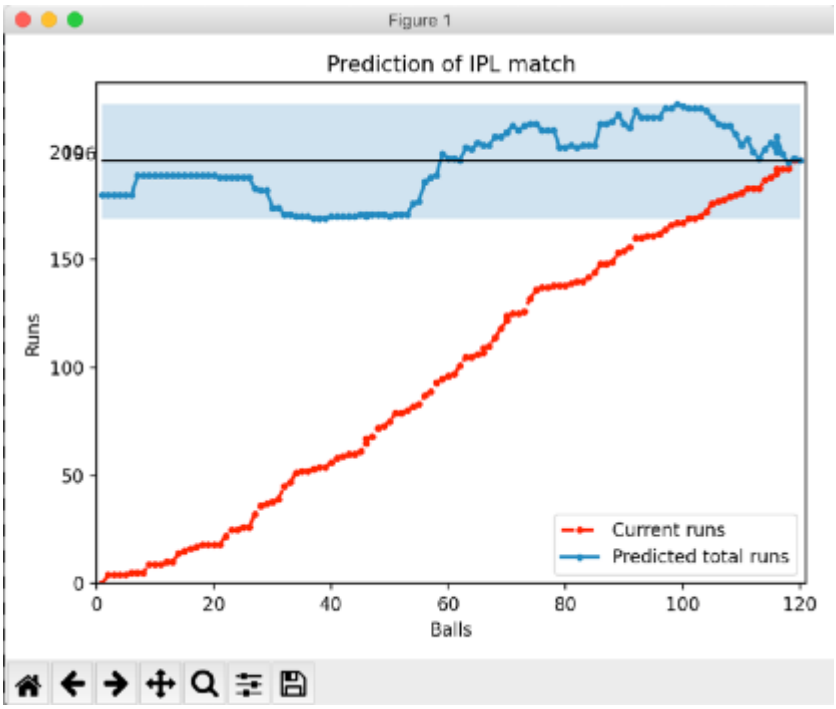
```
x_values = []

y_values_prediction = []

y_values_runs = []
```

x_values has the values of the remaining balls and y_values_prediction is basically the predicted outcome by the machine. Y_values_runs gives us the actual runs scored in the match. The total runs scored is stored in a variable and in the output graph, it is displayed separately.
The x-axis has overs and y-axis has total runs. Separate commands are given for labelling the plot and highlighting the graph with blue and red for predicted score and actual score respectively and a key is created for that. The actual runs scored by the team is highlighted with a black line and the runs is displayed by the side. A blue region is obtained to show the maximum and minimum deviation in the predicted score. A final compact output graph which shows the working of the RandomforestRegressor by displaying the predicted runs, actual runs(with every ball that was played) along with the total runs scored and maximum and minimum deviation is displayed.

RESULTS AND DISCUSSION:

As we've seen how our algorithms work and the methodologies used, let's see how well our results come have come out.

Now, let us look at some statistical analysis of the graph that we've obtained by running the results of the trained model.



.

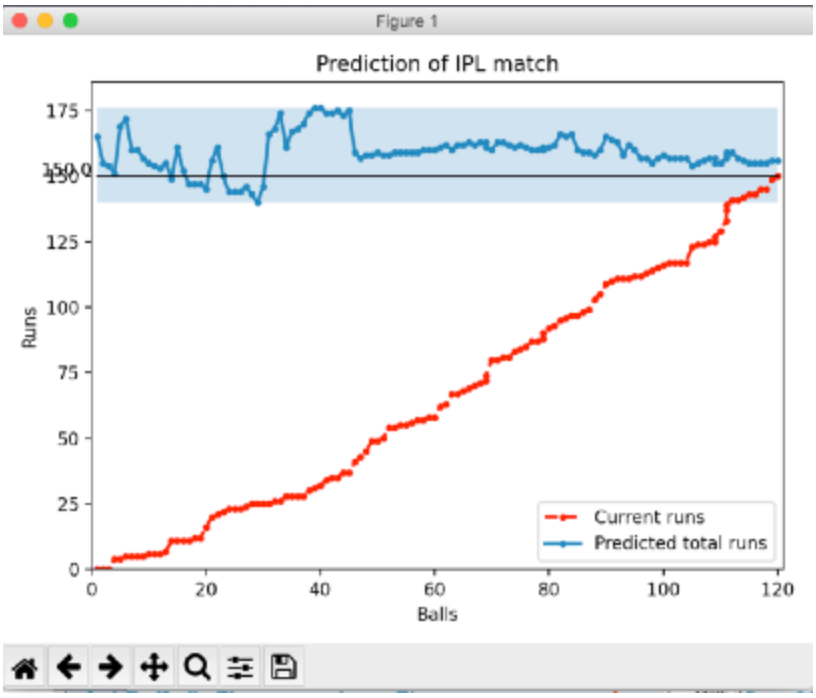
<u>The Components of the Graph:</u>

The x-axis shows the number of balls played by the team in the innings.
The y-axis shows the number of runs scored by the team over time.

The red line in the graph shows the number of runs scored by the batting team with respect to the number of balls played, while the straight black line indicates the number of runs finally achieved by the batting team after the innings ended.

The blue line is our prediction line. It predicts the runs that the batting team may finally be able to achieve by the end of the innings.

Here is another example of the model:



What the graph gives us:

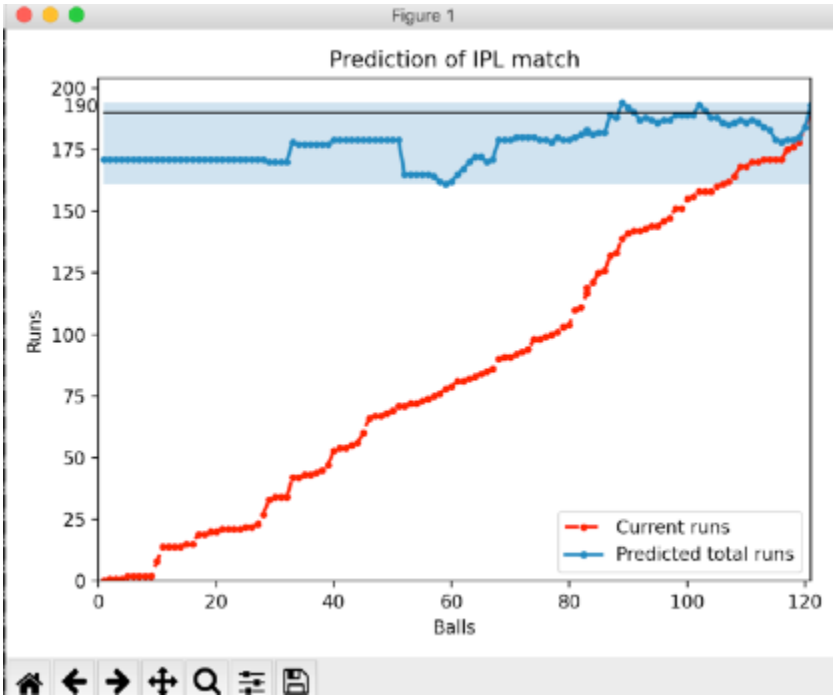
Given an instant of time in the match, the machine learns from our trained model and is able to predict the final runs that the batting team will be able to achieve based on various factors such as the rank of the team, and the various other attributes that we have used in the code to plot the regression-trees.

Prediction and accuracy:

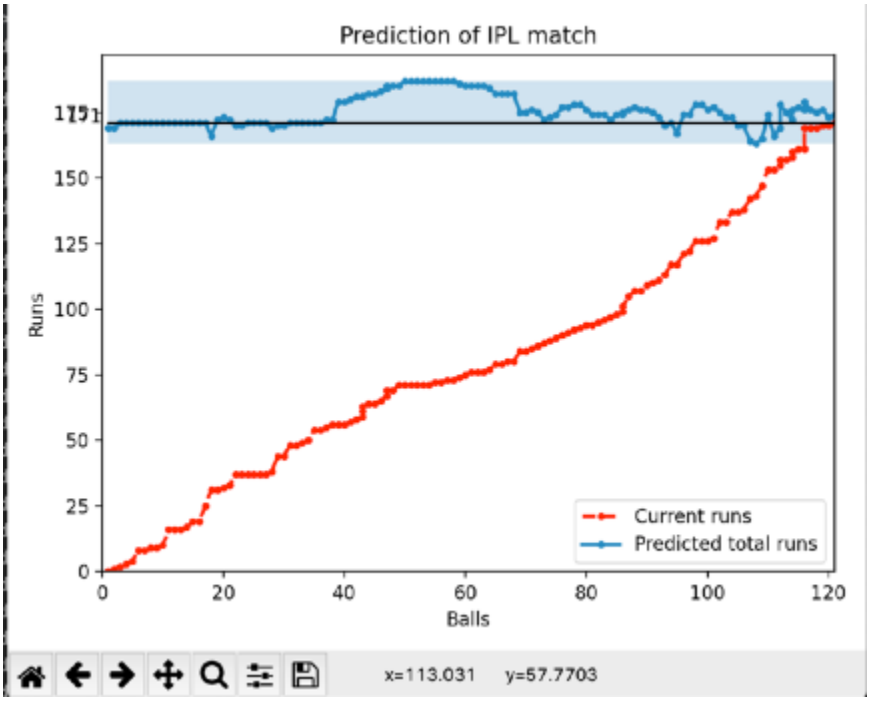
It is practically impossible to predict the accurate number because cricket is an unpredictable game and we cannot accurately judge the gameplay.

Hence the model thrives as good as possible from the trained dataset to make the machine predict the target as close as close as possible, practically.

Let’s have a look at a couple of more models:



(Each model shows different matches played by different teams)



We see that each model's prediction curve keeps changing.

It is due to the fact that we consider so many attributes while the random forest regression tree is being plotted in the training model and for each match the prediction curve varies with respect to the rank of the batting team and the way each team plays, and hence we get different curves in different graph-models.

CONCLUSION AND FINAL WORK:

Model developed in this project is a relatively good predictor compared to some of the methods commonly used, however, there is still scope for improvement. Additionally, it is important to remember that cricket is a game of uncertainty and a game of determination, persistence and confidence, and to measure these is possibly one of the most challenging task in programming today. The model is a good estimator of the total runs, but it is not advisable to only rely on this. It is also important to consider the spirit of the game, and remember that game can be swayed to anyone's side. All it takes are some confidence, courage, determination and skills to take home the victory and glory. With the speed of innovation in the field of AI, computer science and engineering today, making predictions by incorporating these parameters with greater accuracy would be very much possible in the near future!

Scope of the project content:

The model could be made more accurate by incorporating more attributes such as weather and even providing the ability to dynamically learn from the current game. Data on the individual players and their gameplay may also be also used in training the prediction model, however the players gameplay will not always be the same, which is why the dynamic learning feature is important to ensure the model predicts accurately almost all the time.

Implementation of neural networks can also improve the way the predictions are made, moreover, Tensorflow package on machine learning can provide better models to predict with. However, complex algorithms such as these can require time and memory more than what a regular household pc or a laptop can provide. Building a database powered by SQL could also make the analysis a lot smoother than using CSV or Excel files. The prediction model could be mounted on a user-friendly and interactive user interface such as a website or a mobile application in order to simplify the task for users to implement the program. Django web framework is a possible area of focus in achieving this objective, the DASH module in python could also be used in association with javascript and html to make the website more aesthetically pleasing, and as a result encourage more people to use the program.

REFERENCES

JOURNAL-1

[1] Armstrong J and Willis RJ (1993). Scheduling the cricket World Cup-A case study. J Opl Res Soc 44: 1067-1072.

[2] Willis RJ and Terrill BJ (1994). Scheduling the Australian state cricket season using simulated annealing. J Opl Res Soc 45: 276- 280.

[3] Wright MB (1991). Scheduling English cricket umpires. J Opl Res Soc 42: 447-452.

[4] "A Decade of Duckworth-Lewis" (http://news.bbc.co.uk/sport1/hi/cricket/6222943.stm). BBC. 1 January 2007. Retrieved 2009-03-21.

[5] Scorecard (http://uk.cricinfo.com/db/ARCHIVE/1996-97/ENG_IN_ZIM/ENG_ZIM_ODI2_01JAN1997.html) of the 2nd ODI between England and Zimbabwe, 1 January 1997, from Cricinfo.

[6] Duckworth, F.C. and Lewis, A.J. (1998), a fair method of resetting the target in interrupted one-day cricket matches. Journal of the Operational Research Society, Vol. 49 No. 3, pp. 220-227.

[7] Data Analysis Australia's detailed mathematical analysis (http://www.daa.com.au/analytical-ideas/cricket/) of the Duckworth-Lewis Method daa.com.au.

JOURNAL-2

[1] Auto-play: A Data Mining Approach to ODI Cricket Simulation and Prediction Vignesh Veppur Sankaranarayanan, Junaed Sattar and Laks V. S. Lakshmanan Department of Computer Science University of British Columbia Vancouver, B.C. Canada V6T 1Z4.

[2] Analysis and Prediction of Cricket Statistics using Data Mining Techniques Anurag Gangal VESIT, Mumbai Abhishek Talnikar VESIT, Mumbai Aneesh Dalvi VESIT, Mumbai Vidya Zope VESIT, Mumbai Aadesh Kulkarni VESIT, Mumbai.

[3] Predicting the Winner in One Day International Cricket Ananda Bandulasiri, Ph.D.

[4] Ananda Bandulasiri, “Predicting the Winner in One Day International Cricket” Journal of Mathematical Sciences & Mathematics Education.

[5] Tejinder Singh, Vishal Singla and Parteek Bhatia, “Score and Winning Prediction in Cricket through Data Mining” 8 October 2015.

[6] Amal Kaluarachchi and Aparna S Varde, ”CricAI: A classification based tool to predict the outcome in ODI cricket”.

[7] Viraj Phanse, Sourabh Deorah,”Evaluation & extension to the Duckworth Lewis method:

A dual application of data mining techniques”.

[8] K. Raj and P. Padma. Application of association rule mining: A case study on team India. In International Conference on Computer Communication and Informatics (ICCCI), pages 1–6, 2013.

JOURNAL-3

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

[2] Predicting the Match Outcome in One Day International Cricket Matches, while the Game is in Progress. -
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3861745/

[3] Kaluarachchi, Amal, and S. Varde Aparna. ”CricAI: A classification based tool to predict the outcome in ODI cricket.” 2010 Fifth International Conference on
Information and Automation for Sustainability. IEEE, 2010.

[4] ESPN Cricinfo, http://www.espncricinfo.com

[5] Swartz, Tim B., Paramjit S. Gill, and David Beaudoin. ”Optimal batting orders in one-day cricket.” Computers and operations research 33.7 (2006): 1939-1950.

[6] Barr, G. D. I., C. G. Holdsworth, and B. S. Kantor. ”Evaluating performances at the 2007 cricket world cup.” South African Statistical Journal 42.2 (2008): 125

[7] Pedregosa, Fabian, et al. ”Scikit-learn: Machine learning in Python.” Journal of Machine Learning Research 12.Oct (2011): 2825-2830.

[8] Croucher, J. S. ”Player ratings in one-day cricket.” Proceedings of the fifth Australian conference on mathematics and computer sport. Sydney, NSW: Sydney
University of Technology, 2000

-----X---- X ---- X -----