Predicting Winner of Indian Premier League (IPL) Matches Using Machine Learning

Kalki Bhavsar AU1841029, Harsh Mange AU1841130,
Vardhan Shah AU1841138, Khushi Shah AU1841139

Faculty : Mehul Raval
Teaching Assistant: Jay Patel , Arpit Patel

***Abstract*—Prediction of the outcome (winner) of the match using features like home-ground advantage, team-points, power-play performance, and team-points based upon the past performance in addition to the basic features before the match begin showed performance improvement $4.55 \sim 12.12\%$ using logistic regression as our classification model to classify which team wins. Further, optimal values of learning rate and number of iterations was estimated that minimized the cost function.**

I. Introduction

DATA acquisition process has become relatively easy with the advancing of technology over the last few years. The popular field of sports analytics (SA) has become a trend because of the abundant amount of live as well as past data collection. SA is the process of collecting data of the past matches, analyzing them, and extracting useful inference out of it. The inference should help in the effective decision-making process. For example, whether to select batting or bowling after winning the toss, what should be the order of batting, who will win the match, designing strategies for the forthcoming matches based on the players' past performances, etc.

A. Background

Machine learning (ML) can be applied effectively in many areas of sports; both on-the-field and off-the-field. In the case of on-field, ML can be used to predict the performance of the player or the team, outcome of a match, etc. While in the case of off-field business, ML may turn out to be helpful while understanding the sales patterns of tickets, merchandise and assign the prices accordingly.

Generally, the on-field analytics make use of supervised ML algorithms. Regression analysis can be used for calculating the fitness of a player while the classification problem can be used for predicting the outcome (winner) of the match. In off-field analytics, sentiment analysis (SA) can be used to understand the crowd's opinion about a player, team, or league. Nowadays, Twitter is one of the widely used sources of data for SA.

The game of cricket also uses sports analytics to predict the outcome of a match even before the match begins as well as while the game is in progress. ML has also been used to predict the runs (target) or wickets, etc. Hawk-Eye technology is used to track the trajectory of a ball and hence visually display the most statistically accurate path. Indian Premier League (IPL) is a professional cricket league based on T-20 format and is governed by the Board of Control for Cricket (BCCI) in India. Hawk-Eye technology has been officially used in the umpire decision review system in IPL matches since 2009. In 2012, New Zealand introduced a tool named WASP (Winning and Score Predictor) tool to predict the score and winner of a fixed-over cricket match; for example ODI or T-20 format which is used widely even today.

B. Literature Review

To predict the outcome of a match, first, we need to extract the essential features that impact the output variable (label) of a match. According to literature review, the majority of work predicted the outcome of the test or ODI format prior to the match. Many authors has analyzed the features like toss-winning, game-plan (first batting or fielding), and the venue for IPL format [1]. Analysis of various features of an IPL match was done in [2]. The authors of [3] has shown the way of including team strength in the overall prediction. Effect of power-play performance was incorporated in prediction by [4].

C. Motivation

IPL happens every year with participating teams name representing various cities of India. While most of the foreign leagues are not so popular with the franchise team losing money each time, IPL has outperformed exceptionally well. According to an ESPN report, Star Sports sends 2.5 billion dollars for exclusive broadcasting rights. The latest season (13th) of IPL has shown an 21% increment in the number of viewers including both digital streaming media like Hotstar and television [7]. Since the league is so popular worldwide, various brands are now offering discounts and prizes to the customers who correctly predict the questions like winner, player of the match, etc.

Due to this, predicting the winner of the match well in advance may turn out to be useful to the people as well as the companies. If the match is being played on the home-ground and the home team is likely to win, the ticket price may be charges high; also watching this match will be interesting in the stadium itself rather than choosing to watch a losing match. Predicting the winner of IPL seems a interesting and useful problem to solve.

The further work is organized as follows: Section II I-C contains the algorithm to solve the problem followed by numerical results and conclusion.

II. System Model

A. Proposed Work

We need to predict the outcome of the match before the match begins. Among all the formats of cricket, T-20 format sees a lot of turnarounds in the momentum of the game. An

over can completely change a game. Hence, predicting the outcome for a T-20 format game is quite a challenging task. Besides, developing a prediction model for a league which is completely based on auction is another hurdle. IPL-matches cannot be predicted simply by making use of statistics over historical data. Since the players going under auctions, most of the players are bound to change their teams while only few of them are retained in the same team; which is why the ongoing performance of every player must be taken into consideration while developing a prediction model.

In this SL problem, the prediction model $y = f(x)$ is learned by the learning algorithm from a set of data-set: $D = ((X_1, y_1), (X_2, y_2), ..., (X_N, y_N))$. Based on the type of output (y) SL algorithm is divided further into two categories: Regression and Classification. In Regression, the output is a continuous value while the classification deals with discrete kind of output. Here we have to classify the winning team as playing team1 or playing team2. Hence this is a classification problem.

Figure 1, shows the flow of our work. The data is the key to solve this problem. The data of following features id, season, city, date, team1, team2, toss_winner, toss_decision, result (normal or tie), DL_applied, winner, win_by_runs, win_by_wickets player_of_match, venue, home_ground_advantage, umpire1, umpire2, umpire3, powerplay_runs_team1, powerplay_wickets_team1, powerplay_runs_team2, powerplay_wickets_team2, pitch_type, form_of_player, etc. was scraped from the official *ESPN* and *IPL* website. The first step of data collection ends here.

The following teams and their abbreviations were considered: Chennai Super Kings (CSK), Deccan Chargers (SRH), Delhi Capitals (DC), Delhi Daredevils (DC*), Gujarat Lions(GL*), Punjab Kings (PBKS), Kochi Tuskers Kerela (KTK*), Kolkata Knight Riders (KKR), Mumbai Indians (MI), Pune Warriors (PWI*), Rajasthan Royals (RR), Rising Pune Supergiants (RPS*), Royal Challengers Banglore (RCB), and Sunrisers Hyderabad (SRH). Note that * denotes the dead teams. But these teams are imporant to include since they also show the performance of the other active teams.

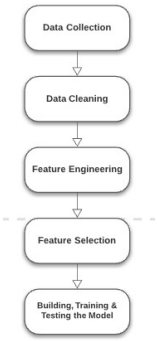

Fig. 1: Flow-chart

In the next step, the data was cleaned. The data contained null values for the features like city, umpire1, umpire2, and umpire3. City could had null values where venue was "Dubai International Cricket Stadium". The null values were replaced by "Dubai" for the column city. The features umpire1, umpire2, and umpire3 were dropped since the number of empty columns 62 for umpire1 and umpire2 while 694 for umpire3. The team names were renamed as mentioned in table I. Deccan Chargers (DC) was renamed to Sunrisers Hyderabad (SRH) in 2012 while Delhi Daredevils (DD) was renamed to Delhi Capitals (DC). Keeping this into account the renaming of teams was done. The next step was to remove redundancy in values. For example, "Feroz Shah Kotla" and "Feroz Shah Kotla Ground" are same venues in the same city Delhi. Also, "M Chinnaswamy Stadium" at cities Bengaluru and Bangalore are the same stadiums. These kind of redundant information was removed in this step.

Comparison of various models accuracy w.r.t feature inclusion				
Included features	Logistic Regression	Decision Tree	SVM	Random Forest
None	50%	56.06%	52.27%	56.82%
Team points	61.36%	52.27%	59.85	56.82%
Home ground	60.61%	50.76%	59.85%	60.61%
Difference of team points	62.12%	60.61%	61.36%	61.36%

TABLE I: Comparison of various models accuracy w.r.t feature inclusion

The next step is Feature Engineering. The column values should make some sense to the computers. Since the computer don't have the ability to understand and draw inference from the text, we need to encode the strings to numeric categorical values. There are two encoding ways. First is manually and second is by using tools *LabelEncoder()* from the Scikit-learn library in PYTHON. The columns "team1", "team2", "winner", "toss_winner", "city", and "venue" were encoded using *LabelEncoder()* while the new columns team1_win, team1_toss_win, and team1_bat were encoded using 1's and 0's manually. In addition to this, home ground advantage was encoded as +1 if the venue is the home-ground of team1. Similarly -1 was used to encode if the venue is the home-ground of team2. If both teams or none of them have home-ground advantage, 0 was used to encode the same. The columns team1_points and team2_points did not make much impact to the accuracy. But when the difference was used i/e team_point_difference that is equal to team1_points - team2_points, the accuracy was improved of $4.55 \sim 12.12\%$.

Adding the feature of the team_strength was also worked upon. We used of multivariate regression to calculate points of each player in the league and compute the overall strength of each team based on the past performance of the players who have appeared most for the team. There are various ways a player can be awarded points for their performance in the field. The official website of IPL has a Player Points section where every player is awarded points based on these 6 features: (i) number of wickets taken (ii) number of dot balls given (iii) number of fours (iv) number of sixes (v) number of catches, and (vi) number of stumpings. To find out how IPL management was assigning points to each player based on these 6 features, a multivariate regression was used on the players' points data. For this problem with six independent variables, the multivariate regression model takes the following form: $Y = B_0 + B_1X_1 + B_2X_2 + B_3X_3 + B_4X_4 + B_5X_5 + B_6X_6$

where, y: points awarded to a player B_0: the bias term, B_1: per wicket weight, B_2: per dot ball weight, B_3: per four weight, B_4: per six weight, B_5: per catch weight, and B_6: per stumping weight.[5] On applying multivariate linear regression, the parameters obtained were $B_0 \approx 0, B_1 = 3.5, B_2 = 1, B_3 = 2.5, B_4 = 3.5, B_5 = 2.5, B_6 = 2.5$.

In the feature selection step, few unnecessary and redundant columns like season, date, id, play_of_match, etc were removed. We found the correlation between two variables using PEARSON R CORRELATION which is mostly used to measure the degree of the relationship between linearly related variables. When two variables are redundant, we decided a threshold of 0.8 while removing the redundant features.

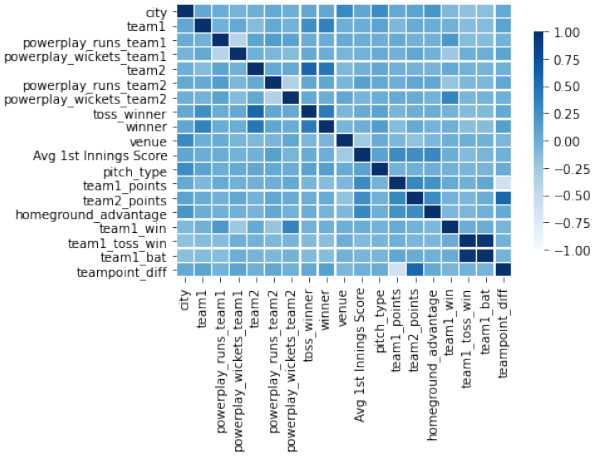

Fig. 2: Correlation Matrix

B. Logistic Regression

Linear Regression is used to determine the value of a continuous input dependent variable. While, logistic Regression is used to solve classification problems. In Here, the target variable has only a limited number of values. Hence, the target variable or label is categorical [6]. In our case, the categories are is whether team 1 wins or team 2 wins. Therefore, our problem comes under **Binary Logistic Regression** since the number of possible outcomes is only two.

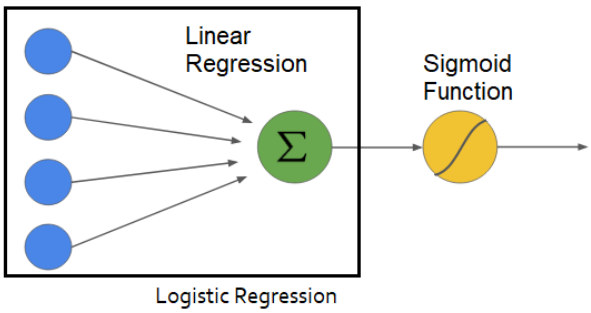

Fig. 3: Logistic Regression

We have used the **Sigmoid** function was our activation function. This function provides hypothesis for our model. The plot looks as shown in 3. It can be seen that the value of the Sigmoid function always lies between 0 and 1. and the value is 0.5 at $X = 0$. Hence, we can use 0.5 value as the probability threshold to determine the classes. If the probability is greater than 0.5, we classify it as Class-1 ($y = 1$, team-1 wins) or else as Class-0 ($y = 0$, team-2 wins).

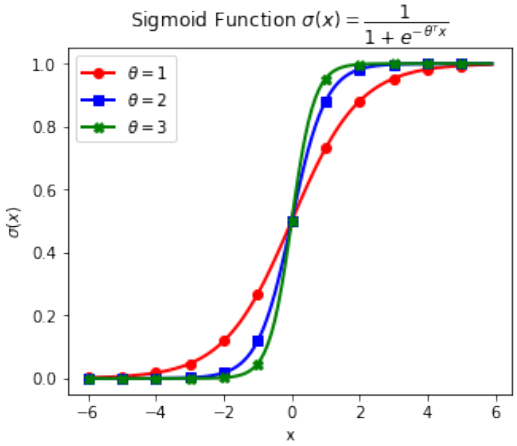

Fig. 4: Sigmoid Function

C. Assumptions

The following assumptions are considered and incorporated in our data-set that is important to be considered while implementing Logistic regression:

1) The dependent variable (label) must be categorical.
2) The variables or features must be independent..

D. Hypothesis

A Linear Regression model can be represented by the hypothesis whose equation is given in ((1)).

$$h_{\text{linear}}(x) = \theta^T x \tag{1}$$

On applying logistic regression model, our hypothesis can be modelled as in (2).

$$h_{\text{logistic}}(x) = \sigma(\theta^T x) \tag{2}$$

where Sigmoid function is given by

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The hypothesis for logistic regression finally becomes,

$$h(x) = \frac{1}{1+e^{-\theta^T x}} \tag{4}$$

$$h(x) = \begin{cases} > 0.5 & \text{if } \theta^T x > 0 \\ < 0.5 & \text{if } \theta^T x < 0 \end{cases} \tag{5}$$

If the weighted sum of inputs $\theta^T x$ is greater than zero, the team 1 wins and vice-versa. So the decision boundary separating both the classes can be found by setting the weighted sum of inputs to 0.

E. Cost Function

The cost function for a single training example used in our model is given in (6). Our objective is to minimize the cost.

$$C(x) = \begin{cases} -\log h(x) & \text{if } y = 1 \\ -\log\left(1 - h(x)\right) & \text{if } y = 0 \end{cases} \tag{6}$$

If team1 wins i.e y is 1 and the model predicts 0, we should highly penalize it and vice-versa. From figure 5, in case of the plot of $-\log(h(x))$, as $h(x)$ approaches 1, the cost becomes 0 and as h(x) approaches 0, the cost approaches ∞; penalizing the model heavily. Similarly for the plot $-\log(1-h(x))$ when the actual value is 0 and the model predicts 0, the cost is 0 and the cost approaches ∞ as $h(x)$ approaches 1.

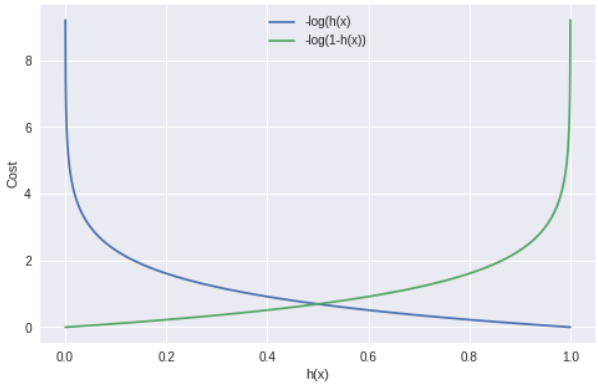

Fig. 5: PLot of $-\log(h(x))$ and $-\log(1-h(x))$

Combine both of the equations, we get (7).

$$C(x) = -y \log h(x) - (1-y) \log\left(1 - h(x)\right) \tag{7}$$

The cost for all the training examples denoted by $J(\theta)$ can be computed by taking the average over the cost of all the training samples.

$$J(\theta) = -\frac{\sum_{i=1}^{i=m} \left[y^i \log h(x^i) + (1-y^i) \log\left(1 - h(x^i)\right)\right]}{m} \tag{8}$$

where m is the number of training examples.

F. Minimizing the Cost Function

Now, our main task was to optimize the logistic regression model. We have used gradient descent function to minimize the cost function of our model. The gradient with respect to any parameter is given in (9).

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{\sum_{i=1}^{i=m} \left(h(x^i) - y^i\right) x_j^i}{m} \tag{9}$$

Using Sigmoid function on our test data we get a matrix of floating values which we have to classify as 0 or 1 based on the threshold value of 0.5. After this final operation, we can get our final predicted classes of which team wins.

For a very low learning rate, the higher accuracy was seen. We considered learning rate $\alpha = 0.001$ and number of iterations $n = 5000$ for which we got the accuracy of 67.52%.

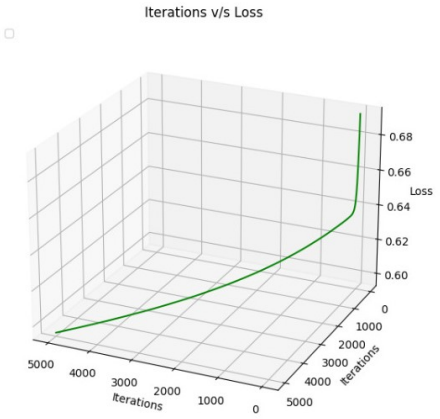

Fig. 6: Number of iterations v/s Loss

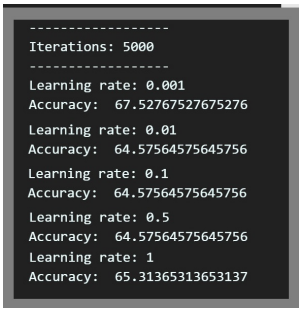

Fig. 7: Accuracy for various learning rates, n=5000

III. Conclusion

Logistic Regression outperformed other models like Decision Tree Classifier, SVM and Random Forest Regression when the new features like home ground advantage, teams' point difference were added, improving the accuracy by 12.12% . Here, considering features like team-strength on the basis of players' past performance, weather, etc. can further improve the accuracy whose data for is difficult to get. The problem of winner prediction was difficult to solve considering IPL matches that known for breathtaking matches whose outcome can be instantly changed in a ball or so with a small-sized data-set of only 812 examples.

References

[1] Pithadia,Geet ."Predictive Analysis of an IPL Match", TowardataScience. Accessed:Mar.17,2021.[Online].

[2] Satya."Predicting outcome of IPL match based on variables",Kaggle. Accessed: Mar.17,2021.[Online]. Available:. https://www.kaggle.com/sathyannarayan/predicting-outcome-of-ipl-match-based-on-variables

[3] Lamsal, Rabindra Choudhary, Ayesha. Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning. (2018)

[4] Sharma, Siddhartha. "IPL Match Prediction Based on Powerplay Using Machine Learning by Siddhartha Sharma" Medium, Artificial Intelligence in Plain English, 17 Dec. 2020.

[5] "IPL 13 viewership up by 28 percent compared to last season: What led to this spike?"Hindustan Times.

[6] Animesh Agarwal. "Building a Logistic Regression In Python", Available:. https://towardsdatascience.com/building-a-logistic-regression-in-python-301d27367c24

[7] Predicting Winner of Indian Premier League (IPL) Matches Using Machine Learning, (2021), GitHub repository,