

Regression based Machine Learning Model to Predict IPL Score after Power Play

Sreekanth Reddy. B¹, Padma. S¹

¹Department of Computer Science & Technology, Madanapalle institute of technology & science, Madanapalle – 517325, Andhra Pradesh, India.

Corresponding author: sreekanthreddy218@gmail.com

Received: 2022 March 15; **Revised:** 2022 April 20; **Accepted:** 2022 May 10

Abstract: - In Cricket, particularly the 20-20 over format and especially Indian Premier League (IPL) is most watched and loved by the people, even where no one can guess who will win the match until the last ball of the last over of the match. In India, the IPL have started in the year 2008 and now it is the most popular T20 league in the world. So, it is decided to develop a basic Machine Learning (ML) model for predicting the scores at the end of powerplay (at the end of sixth over) in every innings for every match. Predicting a score in a Cricket Match depends on various key factors like a home ground advantage, past performances on that ground, records at the same venue. And also, the overall experience of the players, record with a particular opposition, correlation between ever bowler to the ever batsmen and the overall current form of the team and also the individual player. This paper briefs about the key factors that affect the result of the cricket match and the regression model that best fits this data and gives the best predictions. IPL Score predictor is a ML based prediction approach with the help of Python Programming where the data sets namely IPL Ball-by-Ball 2008-2020 and all matches are used like Previous stats data are cleaned and trained in all dimensions covering all important factors such as: Toss, Home Ground, Captains, Favorite Players, Opposition Battle, Previous Stats etc., with each factor having different strength with the help of Regression Model, Consideration of all these strengthening factors helps in predicting the accurate power play scores.

Key Words: Machine Learning, Regression Model, Data Cleaning, Data Training, Python Programming.

1. INTRODUCTION

After football, cricket is most loved and watched by many individuals in the world but in India cricket is the most loved sport. Cricket is one of the most well-liked sports in the world. Cricket, the mainstream and widely played sport popularly known as IPL in India which has the most noteworthy fan base. Especially the 20-20 format i.e., IPL is very popular as it is a fast-paced form of the game that attracts the spectators at the ground and the viewers at home. IPL follows 20-20 format which is very unpredictable. It is a professional 20-20 cricket league that is governed by the Board of Control for Cricket in India (BCCI). The IPL is conducted every year and participating teams represent a city in India. Various natural factors affect the game, the hype has given by the media, and a huge market like fantasy 11 and betting on sites has provided a lot of importance to the model. The rules of the game, the skill of the players, their form, and various other natural factors are very important in the accuracy score prediction of a cricket match score.

The main objective of IPL score prediction after power play is to improve team performance, to provide a new approach to predict scores only after power play, to enhance the chances of winning the game and to boost the fans confidence. The value of a win takes on different forms like thrilling down to the fans filling the stadium seats, television contracts, fan store merchandise, parking, concessions, sponsorships, enrollment and retention.

As the technology is growing and the applications like fantasy 11 and betting onsite are getting popular, people are tending towards predictions given by the machine learning model. The use of machine learning makes life easier in many aspects. To predict the scores of a match this model is using basics of machine learning model. In machine learning, there are two types of learning supervised learning and unsupervised learning. In Unsupervised learning, the data is not properly labelled so the machine has to sort the data according to patterns, combinations without any training given. But in supervised learning, the data is labelled with the proper classification so the machine can easily analyze it and produce the correct result. For our application, the unsupervised learning models are not of any use because the data of cricket matches are properly labelled. So, the model is going to use the supervised learning model. In this model, Regression is used because the outcome will be the score and it is of type number. The main objective is to find the key factors that affect the score prediction outcome and select the best machine learning model that best fits this data and gives the best results. Some works already have been published in this area of predicting the

outcome of a cricket match. In some papers, they predict the outcome of the match or innings score and that too a few key factors are taken for prediction so the accuracy is less. Whereas in some article the machine learning model is not appropriate. So, it is important to take all the key factors that can affect the match outcome and as well as to select the best model for training and testing the data. This will increase the prediction accuracy drastically. The paper is arranged in the following manner. Section 2 represents the literature survey. Section 3 about the methodology. Section 4 concludes the article.

2. LITERATURE REVIEW

Extensive online research produced very few articles related to players performance prediction in the game of cricket. A very small number of researchers have studied the performance of cricketplayers. Similar to predictions of cricket score for the whole match or innings based on the previous ever batsmen to ever bowler data and also few had done to predict the results of the matches based on the previous data provided in the datasets by the Kaggle.

[1] Some studies in machine learning using the game of checkers, shows where two machine learning procedures are used to analyze the results with respect to the situation. In thus model Generalization learning procedure is used to re-evaluate the coefficients for the linear polynomial of program.

[2] Predicting the winner in one day international cricket, coming to this model the prediction is focused on Duckworth-Lewis method where Logistic regression is involved and the evaluation is done through ROC curve.

[3] An analysis of Bayesian classifiers, this model speaks about analysis of behavioral implications, which is representing predicted learning curves and to give experimental resultson artificial domains.

[4] Machine learning to predict the outcome of English county twenty over cricket matches, this is preferably only to the English twenty over county cricket cup which is an optimal model with hierarchical features to outperform gambling benchmark.

[5] Can sports analytics provide wisdom from your data? Definitely, yes rigorous analysis of large data will produce the knowledge in the sports and allows us to find new innovations, researches and findings. But prospective studies should be preferred over retrospective analysis of data to allow new innovative and research.

[6] What businesses can learn from sports analytics? Business can learn way of attaining leadership skills at multiple levels and developing human dimension. It is a theoretical description on how the businesses can learn from sports.

[7] A classification-based tool to predict the outcome in ODI cricket, many parameters which are subset of every match such as Day/Night effect, batting first are involved in Bayesian classifiers in Machine learning which predicts the outcome as well the scores of the match.

[8] A Data Mining Approach to ODI Cricket Simulation and Prediction in which modelling the game is done using subset of lot of match parameters. These were done using the combination of linear regression and nearest neighbor algorithm in which clustering algorithm gives the more complexity which effects the code and similarly increases the false outcomes.

[9] AutoPlay, Cricket Score Predictor is a prediction model that predicts the score as well as the outcome of the match which is completely based on the polynomial regression to value of the determine value of dependent variable which predicts the score and performances.

Hence in all of the above-mentioned papers prediction is done based on the previous outcomes, results and various match factors. But, most important factor correlation between the players is missing. So, this model has taken a dataset of all the previous matches happened previously from 2008-2020 which is taken from Kaggle Website and data is used to train the model. Here the proposed model predicted the performance of ever bowlers against ever batsmen (correlation between the players) which the Indian premier league players played most frequently. Here proposed model outcome is completely different from the already published papers. As this model predicts the score at the end of every six overs in each and every innings during the matches that will held in IPL. This Prediction is completely based on the previous performance between the particular ever batsmen to ever bowler. So, the outcome of this proposed model is at most accurate.

S.NO	AUTHOR	YEAR	FINDINGS	LIMITATIONS
1.	A. L. Samuel [1]	1988	Two ML procedures are used. Rote learning procedure to machine analyse the results with respect to situation.	The two ML programs rote and generalization procedure were not

			Generalization learning procedure to re-evaluate the coefficients for the linear polynomial of program.	sufficient to outplay the best human checker players.
2.	A. Bandula Siri [2]	2008	The Prediction is focused on Duckworth-Lewis method. Logistic regression is involved and the evaluation is done through ROC curve.	Duckworth-Lewis method doesn't have the full amount of information to decide the true winner.
3	P. Langley,		Analysis of behavioural	Chi-squared statistic
	W. Iba,	1992	implications, representing	is used to estimate
	K. Thompson [3]		predicted learning curves and	the dependency
			to give experimental results	coefficients among
			on	attributes but
			artificial domains.	difficulty of
				interpretation when
				there is large amount
				of data.
4.			Preferably only to the English	A prediction method
	S. Kampakis,		twenty over county cricket	combining simple
	W. Thomas [4]	2015	cup. Optimal model with	hierarchical features
			hierarchical features to	are used which
			outperform gambling	results in less
			benchmark.	accuracy.
5.			Rigorous analysis of large	Prospective studies
	L. Pass field,		data	should be preferred
			will produce the knowledge in	
	J. G. Hopker [5]	2017	the sports and allows us to	find
			find	over retrospective

			new innovations, researches	analysis of data.
			and findings.	
6.			Can learn way of attaining	Only a theoretical
	T. H. Davenport	2014	leadership skills at multiple	description on how
	[6]		levels and developing human	the businesses can
			dimension.	learn from sports.
7.	Kaluarachchi,		Factors such as Day/Night	Bayesian classifiers
	Amal,		effect, batting first are	is not at all great for
	S. Varde Aparna	2010	involved in Bayesian	imbalance data at any
	[7]		classifiers in Machine	circumstance so the
			learning.	prediction rate will
				go down.
8.	Sankar Narayanan,		Modelling the game using	Model is based on
	Vignesh Veppur,		subset of match parameters.	subset of match
	Junaed Sattar,	2014	Combination of linear	parameters using
	VS Lakshmanan		regression and nearest	clustering algorithm
	[8]		neighbor algorithm are	which produces more
			involved.	complexity which
				makes higher level of
				code defects which
				results in false
				outcomes.
9.			To Build a prediction model	All the subset
	Aayush Kalla [9]	2018	that predicts team score as	parameters are
			well	included whereas
			as outcome of the match.	correlation between
			Polynomial regression to	an individual is not
			determine value of dependent	

			variable.	involved which
				produces the best
				outcomes.

3. METHODOLOGY

In this system it is decided to predict the score after the end of six overs in every innings. Also, to predict the scores of the teams who are playing the game based on the squad selected. This system will not be only beneficial for the audience but also for the team selector as they can predict the scores on a particular team.

The following are the steps involved in the methodology:

- Data Gathering
- Data Cleaning
- Data Preparation
- Model Development

3.1 Data Gathering:

ML algorithms requires huge amount of data to function and produce outcome properly. When compiling the data, it is not only enough to gather but also have to feed it to the model to expect good outcome. So, this model used the IPL Score datasets which has more than two lakhs’ observations and fifteen features provided by the Kaggle website. The fifteen features are described table1:

Table 1: Description of Features 3.2 Data Cleaning:

S. No	Feature	Description
1.	Id	Player id
2.	Over	which over is the bowler bowling
3.	Inning	which innings either 1 st or 2nd
4.	Ball	no. of bowled in a over
5.	Bowl team	Bowling team
6.	Bat team	Batting team
7.	Batsmen	Batsmen who is batting

8.	Batsmen runs	runs scored by batsmen up to then
9.	Extra runs	runs scored by wides, leg byes and
		so on
10.	Bowler	bowler
11.	Runs	runs scored
12.	Striker	batsman playing as main 1
13.	Non-striker	batsman playing as runner up – not
		main
14.	Fielder	fielder who is fielding
15.	is wicket	whether that ball is wicket or not

Data cleaning is preparing data for analysis by removing or modifying data that is incorrect, irrelevant and duplicate. It is never as easy as organizing some rows and erasing duplicate values, these includes standardizing data sets, correcting the empty fields and to identify the duplicate values. If there is data that doesn't relate to your data set it reflects in accurate results. After understanding our data, we can now proceed to clean it for our use case. Since there are many variables which may not be useful for our problem. And we have the data of every ball that ever played in IPL since beginning, we just need the balls that played in powerplay because the mindset of the players is different in powerplay due to different arrangement of fielders in the field so we are going to keep the relevant data and discard all that we won't need. After cleaning was done in csv file, using "finder" function we get required variables. Here the finder function iterates over all the rows of the data set and for each correlation between the batsmen and bowler it gets the no. of runs scored through four runs and six runs. Also, it returns the runs scored through non boundary runs and runs due to extras.

```

for row in csv_reader: #iterating over the rows of data
    if row[8]==batsmen:
        if row[10]==bowler:
            balls=balls+1
            if int(row[11])<4: #getting no boundary runs
                rr=rr+int(row[11])
                ball=ball+1
            if int(row[11])==4: #getting no.of 4's
                fours=fours+1
            if int(row[11])==6: #getting no.of 6's
                six=six+1
            extras=extras+int(row[12]) #runs due to extras

```

This is a part of finder function which helps to get relevant data of the relevant data variables.

Data cleaning involves in four steps:

- Screening
- Diagnosis
- Treat
- Documenting

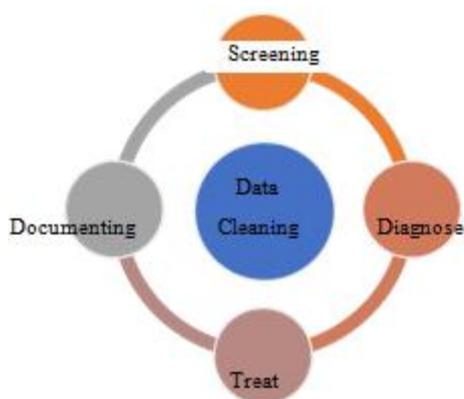


Figure 1: Data Cleaning Process

Data Screening goes through the dataset by checking data for errors and fixing or removing the errors found while screening the data, this helps in good preparation of data for the model. After that data diagnosis is done to determine the correlation between the variables. Once the data is diagnosed now the data can be treated to summarize it as a graph and finally documenting is done by providing sufficient describing information about the data for clear understanding.

3.3 Data Preparation:

Good data preparation produces clean and well-curated results which leads to more clean and practical accurate models. So, the data takes a variability of data types including strings and numbers. But the model requires them all to be in a numeric format, so this needs to make model friendly to perform some calculations. The model wants the data in such a way that fits to this training model. This can be achieved by using finder function.

```

with open('all_matches.csv', mode='r') as csv_file:
    csv_reader = csv.reader(csv_file)
    balls = 0
    ball=1
    rr=0
    fours=0
    six=0
    extras=0
    avg=0
    for row in csv_reader: #iterating over the rows of data
        if row[8]==batsmen:
            if row[10]==bowler:
                balls=balls+1
                if int(row[11])<4: #getting no boundary runs
                    rr=rr+int(row[11])
                    ball=ball+1
                if int(row[11])==4: #getting no.of 4's
                    fours=fours+1
                if int(row[11])==6: #getting no.of 6's
                    six=six+1
                extras=extras+int(row[12]) #runs due to extras
    balls=balls/6 #total no.of overs played
    bats = ['RD Gaikwad', 'F du Plessis', 'MM Ali', 'SK Raina', 'Ms Dhoni',
    batsman=bats.index(batsmen)+1 #giving a numerical value to batsmen.

```

These lines are the heart of the finder function. It executes, by iterating over the rows of the csv file which contains the data of all matches and takes input of two strings “batsmen”, “bowler” and keeps a count of no. of balls bowled by the bowler to that specific batsman and keep track of non-boundary runs, fours, sixes and extras that occurred between these players. It returns a list of encoded batsman value (converting string to number) and economy of the bowler and the rest of the attributes that we found from the data (no boundary runs, extras, no. of 4’s per over, no. of 6’s per over).

3.4 Model Development:

The proposed model uses a **Regression** model for predicting the runs. The main idea behind development of this model is when the model gets the input of the players played in the first 6 overs of the innings, it has to convert them into numbers (for batsmen custom number, for bowler economy of the bowler) then based on this as input it finds the average runs by the batsmen with that bowler by calculating no boundary runs, fours,sixes, extras and these values will be treated as output and will make a regression model. The values of these variables are based on the players played in powerplay of the innings. Ultimately it is trying to know the relation between batsmen and the bowler instead of looking at every aspect in general like this

batsman plays well in spin bowling etc., by know the relation between players indirectly it is taking care of all the aspects that are required. So, the model will start with making a csv file which all these variables later using which it can create a pandas data frame and use for training.

```
filename = "IPL/"+"final_data"+"csv"
with open(filename, 'w') as csvfile:
    csvwriter = csv.writer(csvfile)
    csvwriter.writerow(["batsmen", "bowler", "runs", "fours", "six", "extras", "balls"])
    for i in batsmen:
        print(i)
        for j in bowlers:
            try:
                csvwriter.writerow(finder(i,j))
            except ZeroDivisionError:
                pass
```

These lines of code will do the job of creating a csv file with all data variables we need. The first few lines create the filename “final data” in IPL folder and gives names to the columns and it starts iterating over all batsmen with all bowlers and get the required values using finder function, Here an exception has been raised because there may exist few players who does not have played against together (batsmen, bowler of same team) to deal with it raised an exception.

3.5 Data Visualization:

```
fig = plt.figure(figsize = (8,8))
ax = plt.axes(projection='3d')
ax.grid()
x = dfinput["bowler"]
y = dfinput["batsmen"]
tr = dfout["runs"]
tf = dfout["fours"]
ts = dfout["six"]

ax.plot3D(x, y, tr, ".")
#ax.plot3D(x, y, tf, ".")
#ax.plot3D(x, y, ts, ".")
ax.set_title('3D Plot')
```

This will help to plot the data for visualization with x and y as inputs i.e., batsmen and bowlers and z axis will be output (runs, fours, sixes, extras) since it is a 3D plotting all the four variables in a single plot will be difficult, we need more dimensions so I am going to show one plot at a time.

```
model=tf.keras.Sequential()  
model.add(tf.keras.layers.Dense(2, input_dim=2))  
model.add(tf.keras.layers.Dense(4))  
model.compile(optimizer=tf.optimizers.Adam(learning_rate=0.1),loss='mean_absolute_error')  
model.summary()
```

These lines describe how the model creates datasets for training, “dfinput” contains input, “dfout” contains output.

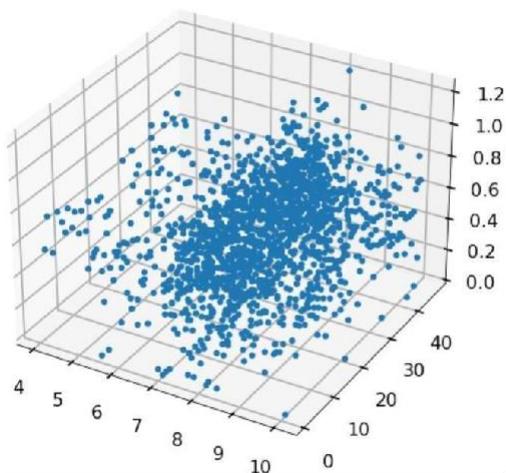


Figure 2: Z axis – runs, fours, extras, sixes.

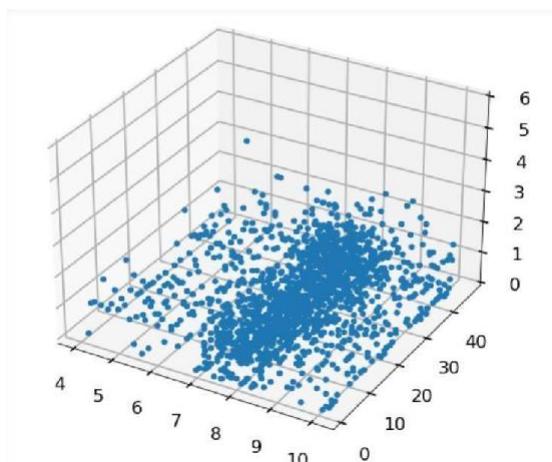


Figure 3: Z axis – Non boundary runs

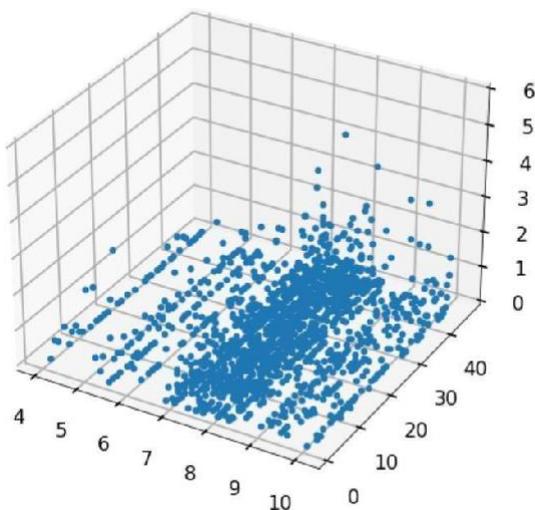


Figure 4: Z axis- fours

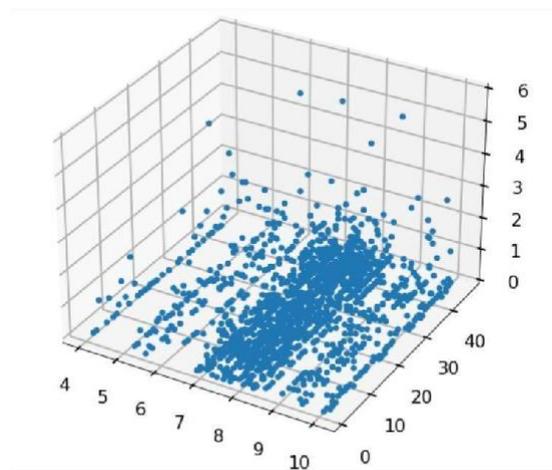


Figure 5: Z axis- Sixes

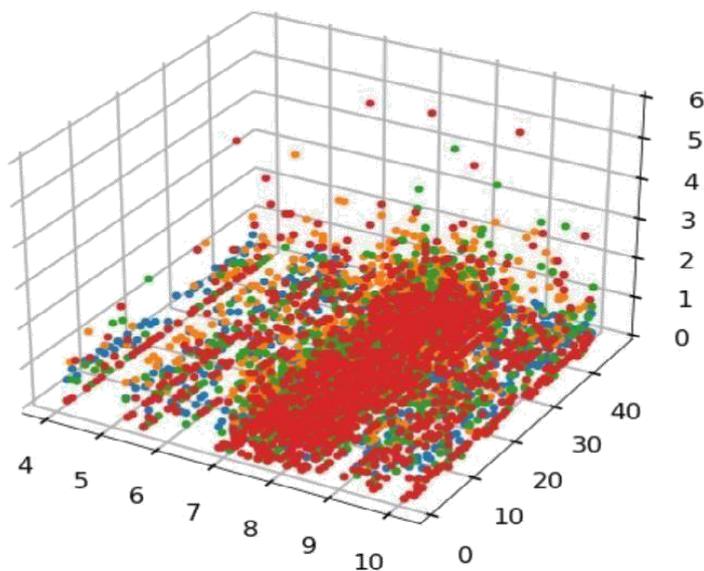


Figure 6: Z-axis extras

The Figure 2 plotting describe about the runs came from the bat of batsmen incase of all form of runs. Figure 3 plotting describe about the runs came from the bat of batsmen incase of non-boundary runs which means in the form of singles, doubles and three runs. Figure 4 plotting shows only the boundary runs scored by the batsmen in the form of four runs. Figure 5 plotting shows only the runs scored by the batsmen in the form of Six(boundary)runs. Figure 6 plotting shows only the runs scored by the team in the form of extras such as byes, wide's, no-ball, penalty runs.

The next big step is to create a proper model for training, here used a simple ML algorithm for finding the parameters for regression, and used Adam optimizer with Mean Squared Error (MSE) as loss function.

Model: "sequential_19"

Layer (type)	Output Shape	Param #
dense_46 (Dense)	(None, 2)	6
dense_47 (Dense)	(None, 4)	12

Total params: 18
 Trainable params: 18
 Non-trainable params: 0

A summary of the model there are 18 parameters to be trained.

```
model.fit(dfinput,dfout,epochs=500)
Epoch 492/500
54/54 [=====] - 0s 1ms/step - loss: 0.3470
Epoch 493/500
54/54 [=====] - 0s 2ms/step - loss: 0.3427
Epoch 494/500
54/54 [=====] - 0s 1ms/step - loss: 0.3480
Epoch 495/500
54/54 [=====] - 0s 2ms/step - loss: 0.3466
Epoch 496/500
54/54 [=====] - 0s 2ms/step - loss: 0.3526
Epoch 497/500
54/54 [=====] - 0s 2ms/step - loss: 0.3577
Epoch 498/500
54/54 [=====] - 0s 1ms/step - loss: 0.3517
Epoch 499/500
54/54 [=====] - 0s 1ms/step - loss: 0.3533
Epoch 500/500
54/54 [=====] - 0s 1ms/step - loss: 0.3407
```

Model. Fit() function will fit the data based on the optimizer and loss function assigned to the model and I have given 500 epochs for better training.

```
def predictor(batsmen,bowlers,model):
    l=6/len(bowlers)
    s=0
    for i in bowlers:
        for j in batsmen:
            x=model.predict([[j,i]])[0]
            s=s+x[1]*4+x[2]*6+x[3]+x[0]*(6-(x[1]+x[2]))-((x[2]*6*(0.3)+x[1]*4*0.3+x[0]*0.3)*(len(batsmen)-2))
    print(s)
```

This the final function which gives the value of the runs that scored in the powerplay of the innings, after saving the model we can use it again by loading after passing it as input to the

“predictor” we can get no. of fours, sixes, extras and ultimately total runs between bowlers and batsmen.

3.6 Step by Step Procedure to Predict Score:

Step 1:

Select the batsmen who are going to play during the first six overs from their particular team from the defined list of batsmen.

```
batsmen = ['RD Gaikwad', 'F du Plessis', 'MM Ali', 'SK Raina', 'MS Dhoni', 'RV Uthappa', 'PP Shaw', 'S Dhawan', 'AM Rahane', 'RR Pant', 'MF
```

Step 2:

For instance, 3 batsmen from Rajasthan royals.

```
l=[batsmen.index("JC Buttler"),batsmen.index("D Padikkal"),batsmen.index("N Rana")]
```

Step 3:

Select the bowlers who are going to bowl during the first six overs from their particular team from the defined list of bowlers.

```
bowlers=["DL Chahar", "SN Thakur",
"DJ Bravo", "SM Curran", "RA Jadeja", "MM Ali",
"L Ngidi", "K Gowtham",
"Imran Tahir", "JP Behrendorff", "MP Stoinis",
"CR Woakes", "R Ashwin", "TK Curran", "A Mishra", "Avesh Khan", "I Sharma", "AR Patel", "K Rabada", "A Nortje", "JJ Bumrah", "TA
DS Kulkarni", "Sandeep Sharma", "T Nataranjan", "B Kumar", "Rashid Khan", "JO Holder",
"S Nadeem", "K Ahmed",
"Mustafizur Rahman", "CH Morris",
"R Tewatia", "JD Unadkat", "Kartik Tyagi", "JC Archer", "AJ Tye",
"Mohammed Shami", "AnshDeep Singh", "CJ Jordan",
"HV Patel", "Mohammed Siraj", "KA Jamieson", "NA Saini",
"Washington Sundar", "YS Chahal",
"KW Richardson", "A Zampa",
"DR Sams",
"DT Christian", "Shahbaz Ahmed",
"M Prasidh Krishna", "KL Nagarakoti", "Harbhajam Singh", "PJ Cummins",
"CV Varun",
"Kuldeep Yadav", "LH Ferguson",
"S Sandeep Warrier", "Shivam Mavi", "Shakib Al Hasan", "BCJ Cutting",
"AD Russel",
]
```

For instance, 3 bowlers from Royal challengers Bangalore.

```
bowl=[dic[bowlers.index("Mohammed Siraj")],dic[bowlers.index("HV Patel")],dic[bowlers.index("NA Saini")]]
```

Step 4:

Run the Predict function.

```
model.predict([[dfinput["batsmen"],dfinput["bowler"]])[0])
```

4. Result: Predicted Score: 48 Actual Score: 46

Match	CSK vs		RR vs		DC vs		SRH vs		RR vs	
	MI		RCB		KXIP		RCB		CSK	
Innings	1	2	1	2	1	2	1	2	1	2
Actual	54	37	38	50	23	35	48	53	54	53
Predicted	48	53	37	39	43	40	36	46	48	56

Accuracy (%)	89	70	97	78	53	88	75	87	89	95
Time taken(μs)	1.01	0.98	1.04	0.96	1.02	0.94	0.98	0.97	0.95	0.92

Table 2: Few Predictions of IPL-2020

The above table 2 describes the comparison of actual score and predicted score for few matches during the IPL-2020 in both the innings of different matches scores after powerplay. The percentage of accuracy is calculated to the nearest integer. Time taken for execution is measured in micro seconds simplified to us where Unicode is not available.

Actual vs Predicted Score graph:

The Prediction of Scores will depend on various parameters such as ground, venue, dew factor, innings, target and many such factors, hence prediction cannot be one hundred percent accurate and there exists a difference between actual and predicted score.

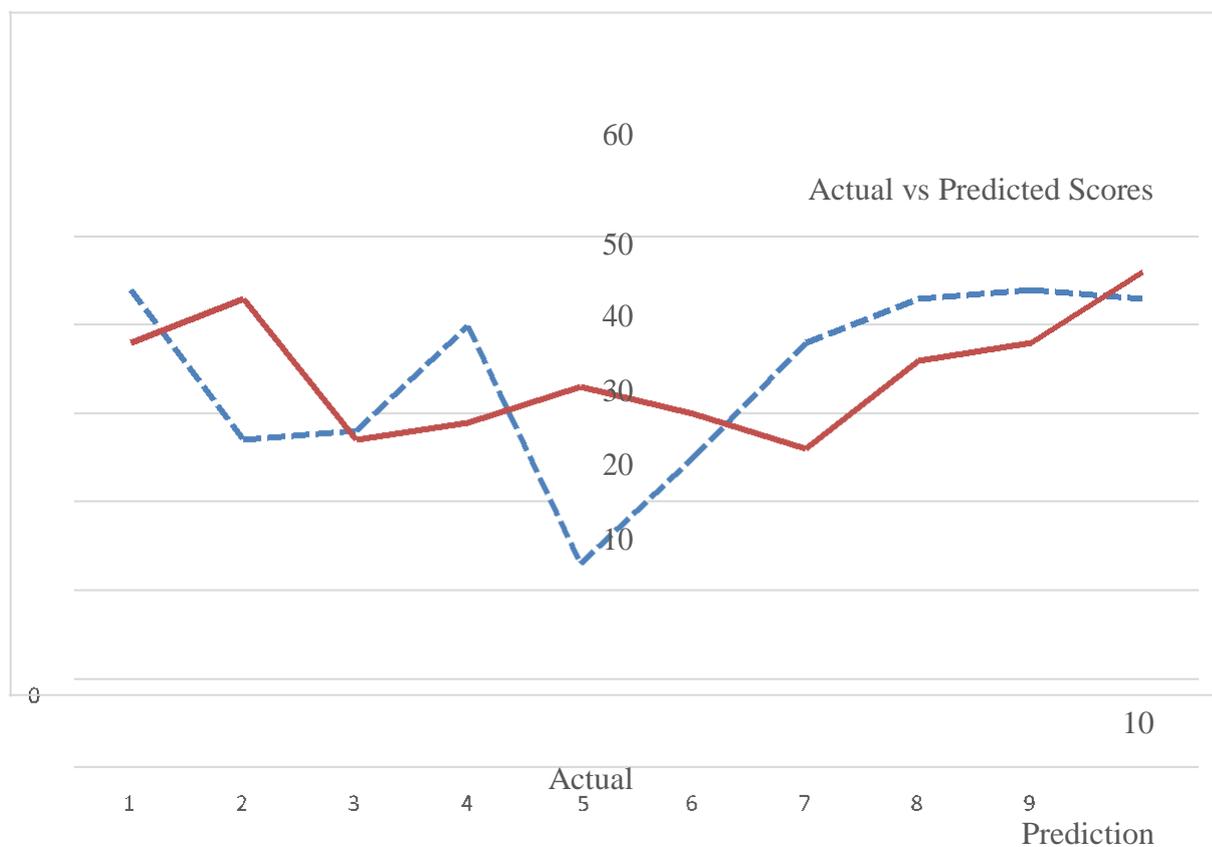


Figure 6: Actual score vs predicted score graph

5. Discussions & Conclusion:

The main moto is to use Machine learning techniques to find the score at the end of six overs and benefits of the model can be describes as to predict the score, for effective prediction technique and to predict the powerplay score. There are few issues with the data that don't have a proper weather or pitch report of the day on which match is played and these can play a crucial part in the innings and as the model don't have any information regarding form/fitness of the players all that have is past data and there is a good chance that their performance form might be changed. There can be few improvements in the model which has used to train instead of just a single deep layer we can have multiple deep layers. With multiple layers the model noticed that there was no any reduction loss, so stuck with the simplest model these happened may be due insufficient data of pitch, weather and form of the player in that season. Finally, a lot of work can be done in the final function here just distributed the balls played to batsmen in a simple way of runs scored per over by the batsmen (just like taking average) and reduced a little bit to account for the wickets taken instead of doing this in a simple way since we can have the data of no. of fours, no. of sixes, extras we can try to simulate the actual ball by ball scenario by assuming some initial conditions and thus can get a better expected value of the final runs scored.

References:

- [1] A. L. Samuel, some studies in machine learning using the game of checkers. in recent progress, in Computer Games I, pp. 366–400, Springer.
- [2] A. Bandula Siri, Predicting the winner in one day international cricket, Journal of Mathematical Sciences & Mathematics Education.
- [3] P. Langley, W. Iba, K. Thompson, et al., “An analysis of Bayesian classifiers,” in Aaai, vol. 90, pp. 223–228, 1992.
- [4] S. Kampakis and W. Thomas, “Using machine learning to predict the outcome of English county twenty over cricket matches,” arXiv preprint arXiv:1511.05837, 2015.
- [5] L. Passfield and J. G. Hopker, “A mine of information: can sports analytics provide wisdom from your data?,” International journal of sports physiology and performance, vol. 12, no. 7, pp. 851–855, 2017.
- [6] T. H. Davenport, “What businesses can learn from sports analytics,” MIT Sloan Man-agement Review, vol. 55, no. 4, p. 10, 2014.

- [7] 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.
- [8] Sankar Narayanan, Vignesh Veppur, Junaed Sattar, and Laks VS Lakshmanan. Auto-play: A Data Mining Approach to ODI Cricket Simulation and Prediction. SDM, 2014.
- [9] Aayush Kalla, AutoPlay - Cricket Score Predictor, International Journal of Engineering Science and Computing, 2018.
- [10] Adichie, J.N. [1967], "Estimates of regression parameters" Ann.Math.Stat.
- [11] Data Cleaning by acaps, [Acaps](#).
- [12] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of Machine Learning Research 12.Oct (2011): 2825-2830.
- [13] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, 3rd Edition ed., Waltham: Elsevier, 2012.