2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) | 978-1-6654-2503-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMARTGENCON51891.2021.9645855

Live Cricket Score Prediction Web Application using Machine Learning

Eeshan Mundhe
Information Technology
K. J. Somaiya College of Engineering
Mumbai, India
eeshan.m@somaiya.edu

Ishan Jain
Information Technology
K. J. Somaiya College of Engineering
Mumbai, India
ishan.jain@somaiya.edu

Sanskar Shah
Information Technology
K. J. Somaiya College of Engineering
Mumbai, India
sanskar.shah@somaiya.edu

***Abstract*- Cricket is an uncertain sport in which two teams compete against each other, and no Machine Learning model can predict the outcome with 100% accuracy. However, with some inputs such as conditions, required factors, circumstances, and past data, it is possible to predict an approximate score for either team or the match's victor. As a result, analysts from the respective teams use all the available data and statistics to improve their team's performance. Therefore, the authors have developed a web application for performing predictive analysis of a live T-20 match to predict the final score and to also predict the winner of the contest before the match begins. A live score scraper is also included in the app, which extracts accurate match data and loads it into the model for prediction. For a deeper understanding, the application allows to visualize the predictions using graphs. Algorithms such as Multivariate Polynomial Regression and Random Forest Classifier are used to make these predictions, while the web application is developed using Flask.**

Keywords: Cricket, IPL, Predictive Analytics, Machine Learning, Web Application, Flask.

I. INTRODUCTION

There are numerous T20 leagues taking place around the world and almost every cricketing nation has its own league. The Indian Premier League takes place every year and has a lot of stakes involved. Every team strives to win their respective league. Analysis and prediction have become an essential part of this circuit, and the support staff of these teams includes not only the doctors, trainers, and coaches, but also the analysts. Predictions can be made for the number of runs scored by either team, the final score after 20 overs, the number of wickets lost, and so on. Many prerequisites must be met for these predictions to be accurate and useful. These prerequisites take the form of historical data, current conditions, player information, and so on.

There are two main modules in this web application, the first module is the Victory Predictor, and the second module is the Run Predictor. Different graphs are used for visualization of prediction. Data scraped from a live match is used to estimate the final score at the end of innings. This is achieved by considering factors such as which team is batting and bowling, the number of overs bowled, the present score of the batting team, the total wickets taken by the bowling team, the runs scored, and the wickets lost in the 5 overs before scraping. The batting team, bowling team, winner of the toss, the venue, are the input features for the victory predictor module. The visualization module works in tandem with the Live Runs Predictor module.

The structure of the paper is as follows: Section I provides an outline of the research, Section II contains a literature review of the existing work in the same field, and Section III contains the proposed methodology. Section IV includes the implementation details for each module. Section V includes the results of the different Machine Learning algorithms used. Section VI describes the conclusion and VII addresses potential future work.

II. LITERATURE REVIEW

The study of the current match prediction work in the Cricket domain explains why predictions are not made on Test cricket matches, but T-20 is chosen as there are many matches played each year, despite being the most difficult format to model due to its highly unpredictable nature. The authors investigated a few missing links, such as dealing with some players' lack of data and considering the playing conditions and hope to develop a better model once these missing links are discovered, as a larger data set improves prediction accuracy [6].

Prediction of player results, including the runs scored by each batsman and the total wicket of each bowler is performed and the results of four multiclass classification algorithms were compared. For both datasets, Random Forest proved to be the most reliable classifier for predicting these outcomes [2].

For predicting every cricketer's performance based on their previous records, the cricketers are split to form three categories: performer, moderate, and failure. The neural network models were systematically trained and tested, and reliable predictions were thus obtained for the future matches [8].

Investigation using an ordered response model to predict test cricket outcomes indicate that the results

of test matches can be determined by simple measures of batting and bowling inputs. The circumstances where wrong predictions are produced by the model were examined, and it was discovered that the most frequent occurrences were failed fourth innings chases, successful fourth innings runs chases and rain delayed matches [7].

III. METHODOLOGY

HTML and Bootstrap. JavaScript is used for validation within the HTML. On the front end, the user can choose between two modules, the Runs predictor, and the Victory predictor. The back end includes developing the web scraper, feeding the data to the modules, deployment of the prediction models for both modules and displaying the visualization of the match predictions.

The flow of the application is as given in Fig. 1., the level 1 of the data flow diagram.

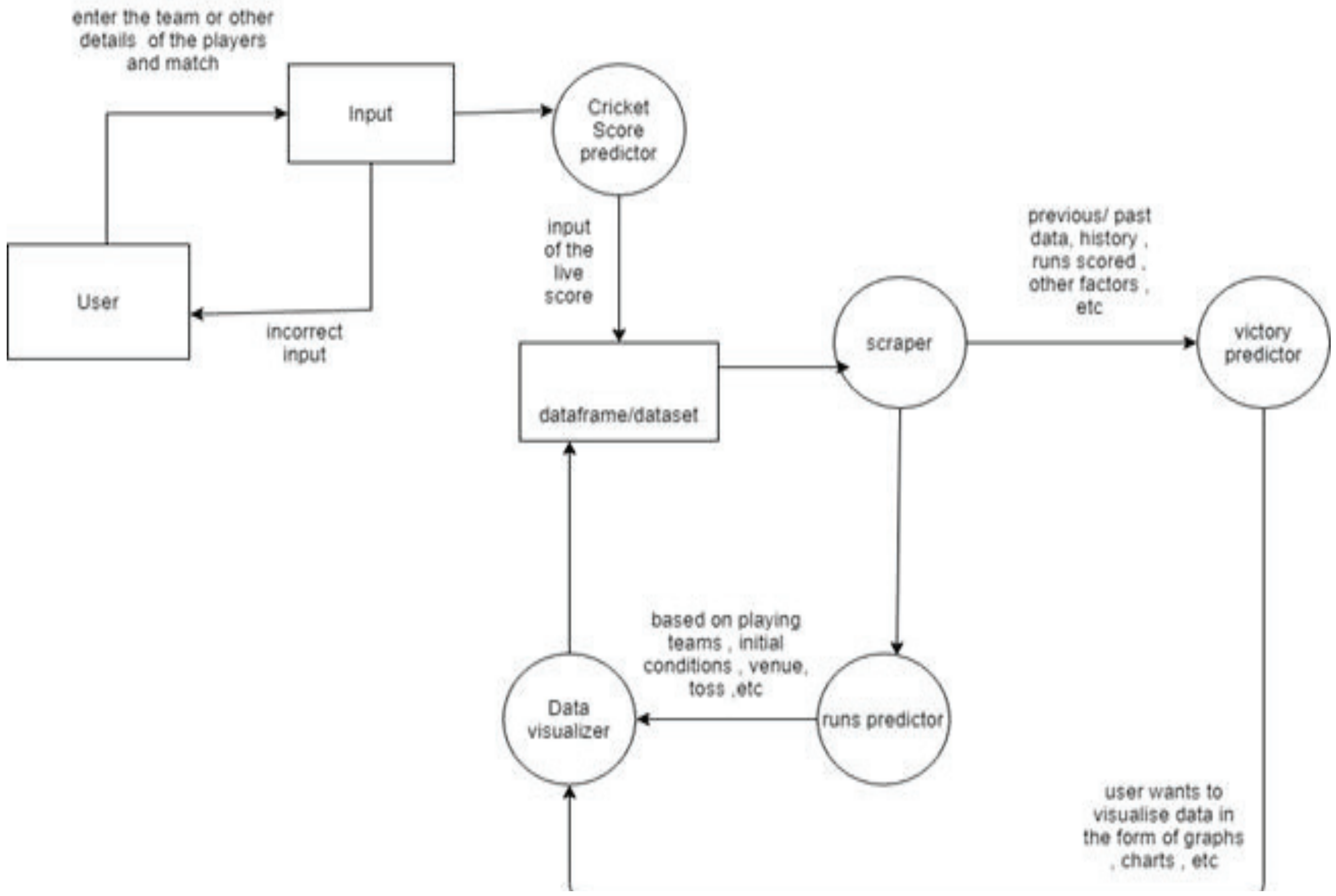
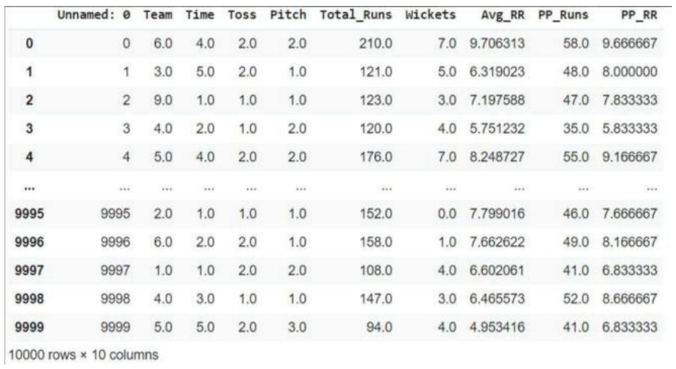

enter the team or other details of the players and match
Input
Cricket Score predictor
User
incorrect input
input of the live score
dataframe/dataset
scraper
previous/ past data, history , runs scored , other factors , etc
victory predictor
based on playing teams , initial conditions , venue, toss ,etc
Data visualizer
runs predictor
user wants to visualise data in the form of graphs , charts , etc

Fig. 1. . Data Flow Diagram Level 1

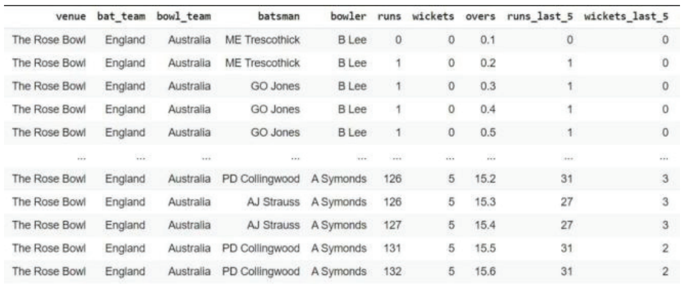
The live runs predictor dataset had 98923 rows and 15 columns, while the Victory Predictor dataset had even fewer rows, only 15762 rows and 28 columns. These datasets are available for free on Kaggle and Data World. In both cases, the dataset was not entirely clean and needed extensive cleaning, which was completed as previously stated. The dataset for the Live Runs Predictor contains data for most matches played since the start of the IPL in 2008, while the dataset for the victory predictor contains data for just a few matches due to the large number of columns. Fig. 2 and Fig. 3 show the head and tail of our datasets.



	Unnamed: 0	Team	Time	Toss	Pitch	Total_Runs	Wickets	Avg_RR	PP_Runs	PP_RR
0	0	6.0	4.0	2.0	2.0	210.0	7.0	9.706313	58.0	9.666667
1	1	3.0	5.0	2.0	1.0	121.0	5.0	6.319023	48.0	8.000000
2	2	9.0	1.0	1.0	1.0	123.0	3.0	7.197588	47.0	7.833333
3	3	4.0	2.0	1.0	2.0	120.0	4.0	5.751232	35.0	5.833333
4	4	5.0	4.0	2.0	2.0	176.0	7.0	8.248727	55.0	9.166667
...	...	...	...	...	...	...	...	...	...	...
9995	9995	2.0	1.0	1.0	1.0	152.0	0.0	7.799016	46.0	7.666667
9996	9996	6.0	2.0	2.0	1.0	158.0	1.0	7.662622	49.0	8.166667
9997	9997	1.0	1.0	2.0	2.0	108.0	4.0	6.602061	41.0	6.833333
9998	9998	4.0	3.0	1.0	1.0	147.0	3.0	6.465573	52.0	8.666667
9999	9999	5.0	5.0	2.0	3.0	94.0	4.0	4.953416	41.0	6.833333

10000 rows × 10 columns

Fig. 2. The Victory Predictor dataset after cleaning.



venue	bat_team	bowl_team	batsman	bowler	runs	wickets	overs	runs_last_5	wickets_last_5
The Rose Bowl	England	Australia	ME Trescothick	B Lee	0	0	0.1	0	0
The Rose Bowl	England	Australia	ME Trescothick	B Lee	1	0	0.2	1	0
The Rose Bowl	England	Australia	GO Jones	B Lee	1	0	0.3	1	0
The Rose Bowl	England	Australia	GO Jones	B Lee	1	0	0.4	1	0
The Rose Bowl	England	Australia	GO Jones	B Lee	1	0	0.5	1	0
...	...	...	...	...	...	...	...	...	...
The Rose Bowl	England	Australia	PD Collingwood	A Symonds	126	5	15.2	31	3
The Rose Bowl	England	Australia	AJ Strauss	A Symonds	126	5	15.3	27	3
The Rose Bowl	England	Australia	AJ Strauss	A Symonds	127	5	15.4	27	3
The Rose Bowl	England	Australia	PD Collingwood	A Symonds	131	5	15.5	31	2
The Rose Bowl	England	Australia	PD Collingwood	A Symonds	132	5	15.6	31	2

Fig. 3. The Runs Predictor dataset after cleaning.

After pre-processing, we must work on model training, so we started by selecting which features to use. Some features were categorical, we used Label Encoder and One Hot Encoder for their conversion. We decided to randomize the rows of our dataset so that it was not chronologically ordered. Following that, we divided our data into training and testing sets using 80% of our data for training and 20% for testing. Since we are using a simpler algorithm, hyperparameter tuning is not as important, so we used an exhaustive method to find the best hyperparameters and used them.

For model training, we tested several algorithms like SVC, Logistic Regression, Random Forest Classifier, and Decision Trees before deciding on which one to use for the victory predictor. For the Live Runs Predictor module, we decided to use advanced regression algorithms like XGBoost, Gradient Boosted trees, etc. Finally, Multivariate Polynomial Regression is chosen to predict the final runs, while Random Forest Classifier is used to predict the winner. We used the R2 score and precision-recall as our metric to verify accuracy for the Live Runs Predictor. For the Victory Predictor, we used model confidence as our metric since confidence of the model can also act as the percentage chance of winning based on the model. We used an exhaustive method for hyperparameter tuning Then we made a pickle to go with it. Fig. 3. shows the cleaned dataset used for the victory prediction model, and the Fig. 4. shows the cleaned dataset for runs prediction model.

For developing the Victory Predictor module, the three different algorithms tested along with their respective accuracy are shown in Table 1. Random Forest Classifier being the most accurate model, we implement it for the Victory Predictor module.

TABLE I. COMPARISON OF ALGORITHMS AND THEIR ACCURACIES FOR VICTORY PREDICTION MODULE.

Model	Accuracy
Logistic Regression	42.10%
SVC	51.21%
Random Forest Classifier	55%

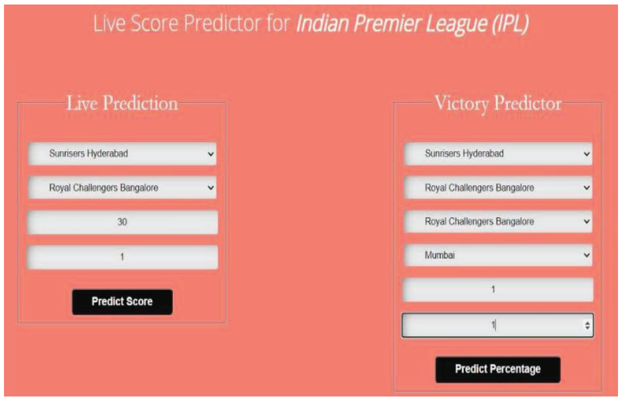
Similarly, the four different algorithms tested for developing the Score Prediction module along with their respective accuracy are shown in Table 2. As Multivariate Polynomial Regression gives the highest accuracy among them, we chose it for our Live Runs Predictor module.

TABLE II. COMPARISON OF ALGORITHMS AND THEIR ACCURACIES FOR RUNS PREDICTION MODULE.

Model	Accuracy
XGBoost	55.67%
SVM	47.6%
Random Forest Classifier	66%
Multivariate polynomial Regression	67.3%

Matplotlib, NumPy, and Seaborn are used in the data visualizer module. For the data visualizer we used the current run rate and standard deviation of the runs scored per over to generate a vector of runs that might be scored per over to reach the predicted final score. Fig. 6. shows the representation of the Graphical outputs, where X-Axis of the graphs represents the overs bowled while the Y-Axis represents the total runs scored, and Fig. 7. depicts the actual score and the predicted score by our Machine Learning model. Fig. 5. shows the UI for home page.

We have also performed testing on our application like load testing, compatibility testing, etc. To do this software like Selenium, JMeter, Lambda testing, Google Page Speed Insights, etc., were used.



Sunrisers Hyderabad
Royal Challengers Bangalore
30
1

Sunrisers Hyderabad
Royal Challengers Bangalore
Royal Challengers Bangalore
Mumbai
1
1

Fig. 4. The Home Page

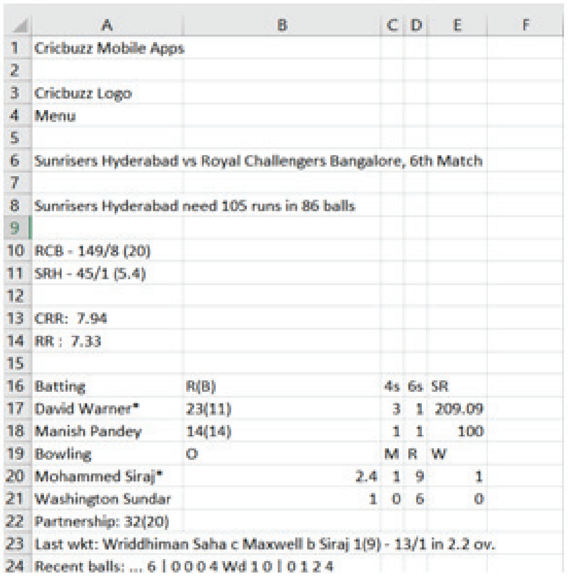
IV. IMPLEMENTATION DETAILS

To implement the proposed methodology, we divided our project in four different modules - web scraper, live runs predictor, victory predictor and data visualizer. All these modules will be encapsulated into our project using the Flask framework. To show the working of our system, we take the example of a match going on between Royal Challengers Bangalore and Sunrisers Hyderabad.

1. *Pre-processing:* For pre-processing the data for the victory predictor module, we began by removing unnecessary features from the dataset, such as "match id," "batsman name," "bowler name," and so on, because they

are irrelevant to our model. Then we made a list of all the IPL teams that have not been consistent over multiple seasons,such as "Pune Supergiants," and we removedall its tuples from the dataset. The we removed the data from overs 0 to 5 because the prediction made during these overs will change drastically.The runs scored in this duration is often uncertain and volatile, the result of a match does not rely much on the first five overs because they are just the start, and with less data for run rate, runs scored would serve as an outlier for the dataset rather than a true datapoint. We only searched for empty values, null values, and invalid data types because the dataset used for the victory predictor had already been cleaned. The competing teams, toss winner, toss decision, and so on were the features used from this dataset; if there was any empty or invalid value, we couldn't fill that datapoint, so we had to drop the entire row for any missing value. We used a similar method for our Live Runs Predictor module. The main difference being that in this dataset, we divided the training and testing sets based on the season of IPL because that is our motive for the project.

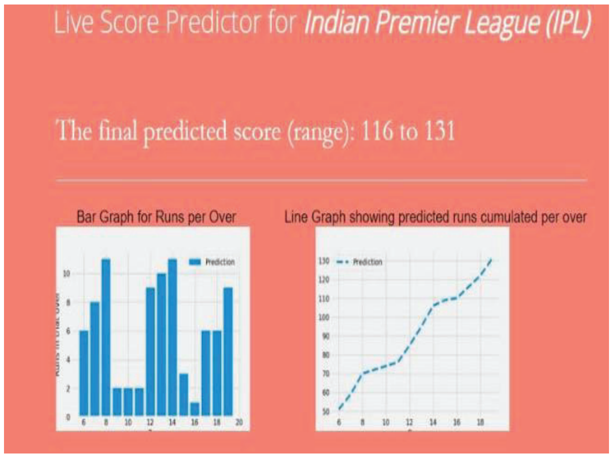
2. *Web Scraping*: Live score of a match is scraped from Cricbuzz website. Using the match link in the import data section, the live match feed is imported into excel. The rows and columns containing the runs scored, overs played, current run rate, and wickets lost are extracted and fed into the Machine Learning models. We can see from Fig. 5 the scraped data for the match from the Cricbuzz website.



	A	B	C	D	E	F
1	Cricbuzz Mobile Apps					
2						
3	Cricbuzz Logo					
4	Menu					
5						
6	Sunrisers Hyderabad vs Royal Challengers Bangalore, 6th Match					
7						
8	Sunrisers Hyderabad need 105 runs in 86 balls					
9						
10	RCB - 149/8 (20)					
11	SRH - 45/1 (5.4)					
12						
13	CRR: 7.94					
14	RR : 7.33					
15						
16	Batting	R(B)	4s	6s	SR	
17	David Warner*	23(11)	3	1	209.09	
18	Manish Pandey	14(14)	1	1	100	
19	Bowling	O	M	R	W	
20	Mohammed Siraj*	2.4	1	9	1	
21	Washington Sundar	1	0	6	0	
22	Partnership: 32(20)					
23	Last wkt: Wriddhiman Saha c Maxwell b Siraj 1(9) - 13/1 in 2.2 ov.					
24	Recent balls: ... 6 | 0 0 0 4 Wd 1 0 | 0 1 2 4					

Fig. 5. Scraped data in MS-Excel.

3. *Runs Predictor:* This module requires the selection of Batting and Bowling teams, as well as the runs scored and wickets lost in the previous 5 overs as well as the current score, which is taken from the live score scraper. Using the selected input, the prediction generates a prediction score for the batting team using the Multivariate Polynomial Regression algorithm which we made using the Polynomial Regression class of scikit-learn library using degree = 3.



Live Score Predictor for *Indian Premier League (IPL)*

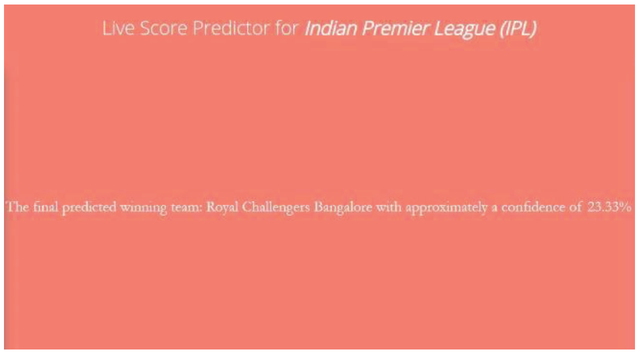
The final predicted score (range): 116 to 131

Bar Graph for Runs per Over

Line Graph showing predicted runs cumulated per over

Fig. 6. The Score Prediction and it's visualization.

4. *Victory Predictor:* This module uses the data of the toss winner and the names of both the teams along with the venue and toss decision to determine the winner using Random Forest Classifier Algorithm. The RFC algorithm is a part of the sklearn.ensemble library. The hyperparameters used are max depth = 3, number of trees = 150.



Live Score Predictor for *Indian Premier League (IPL)*

The final predicted winning team: Royal Challengers Bangalore with approximately a confidence of 23.33%

Fig. 7. The Victory Prediction.

5. *Data Visualization:* This section is used with the live run's prediction module, and it displays a graph that will show the predicted score. This is the section that is used to forecast the final score. There are two graphs that the user can see in our model. The bar graph is the first, and the line graph is the second. As the innings advance, a line graph depicts the progression of runs scored. It will plot the cumulative sum of the scores after every over.

V. RESULTS

The proposed web application enables the users to view the predicted outcomes of a live cricket match. The prediction system correctly predicts both the game's outcome and the final score. It is observed that while predicting the runs for a live match at the end of 20th over the actual score was in the range of the predicted scores for 6 out of 10 times since the accuracy of the algorithm used i.e. multivariate polynomial regression was 67.3%. Similarly, while considering the victory prediction before the start of the match, the algorithm random forest classifier gives an accuracy of 55%. The final predicted winning team would be correct for more than 1 out of 2 times compared with the actual winning team. The model also predicts the approximate confidence by which the team wins, which gives an idea of how the team will win based on the factors and toss as selected by the user on the home page. The data visualizer also allows the user to see how the match will progress and how runs will be scored in each over and the total score after the end of each over cumulatively.

Some previous work in this field includes predictions of past matches that were already played and the results were known beforehand or some would consider live matches with only a few factors taken into consideration and hence the result would not be much accurate. Some would not consider the visualization part. This project is better than these previous works, since it gives the predictions for live matches, and predicts the winner before the match is even played. It also considers various factors and past data to increase the model's accuracy while the user can visualize the data for a better understanding of the predictions.

VI. CONCLUSION

The web application can predict the final score of the innings in a live IPL T-20 cricket match. The system also gives a probable winner for the match before the match even starts by using historic data of the two teams competing. Each match result can be visualized by the users of the system for the clarity of the difference in predicted result and actual result. This result visualized can be used for post-match analysis by the teams to improve on their performance for the next matches. A variety of Machine Learning algorithms were put to test, each giving a different level of accuracy. We chose the most accurate algorithms, and our prediction models yield a reasonably accurate result.

VII. FUTURE WORK

The scope of the application can be expanded to include One Day International and Test matches. Integration of all three game formats into a single application is possible for all types of games, including domestic, international, and multi-franchise tournaments, but it will necessitate extensive historical data for all previous games in every format without any missing links, which is currently unavailable. The existing modules can be improved by including additional features that affect the match:

The victory predictor can be improved by taking into consideration the climatic conditions for the venue which indirectly concerns the pitch quality.

The live predictor model can be altered by incorporating factors like current bowler, batsmen on strike and non-strike, etc. for more accurate results. The unavailable statistics like individual player data in our current dataset, when available could be used to improve the accuracy of our current Machine Learning model, as many of those features could significantly impact the outcome of the game.

ACKNOWLEDGEMENT

The authors would like to thank Kiran Kumari, Assistant Professor at K. J. Somaiya College of Engineering Mumbai, for her suggestions to improve the flow and the formatting of the paper.

REFERENCES

[1] Hodge, Victoria J., et al. "Win prediction in multi-player esports: Live professional match prediction." IEEE Transactions on Games (2019).

[2] Passi, Kalpdrum, and Niravkumar Pandey. "Increased prediction accuracy in the game of cricket using machine learning." arXiv preprint arXiv:1804.04226 (2018).

[3] Lamsal, Rabindra, and Ayesha Choudhary. "Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning." arXiv preprint arXiv:1809.09813 (2018).

[4] Bandulasiri, Ananda. "Predicting the winner in one day international cricket." Journal of Mathematical Sciences & Mathematics Education 3.1 (2008): 6-17.

[5] Mustafa, Raza Ul, et al. "Predicting the cricket match outcome using crowd opinions on social networks: A comparative study of machine learning methods." Malaysian Journal of Computer Science 30.1 (2017): 63-76.

[6] Hatharasinghe, Manuka Maduranga, and Guhanathan Poravi. "Data Mining and Machine Learning in Cricket Match Outcome Prediction: Missing Links." 2019 IEEE 5th International Conference for Convergence in Technology (I2CT). IEEE, 2019.

[7] Brooks, Robert D., Robert W. Faff, and David Sokulsky. "An ordered response model of test cricket performance." Applied Economics 34.18 (2002): 2353-2365.

[8] Iyer, Subramanian Rama, and Ramesh Sharda. "Prediction of athletes performance using neural networks: An application in cricket team selection." Expert Systems with Applications 36.3 (2009): 5510-5522.

[9] Jayanth, Sandesh Bananki, et al. "A team recommendation system and outcome prediction for the game of cricket." Journal of Sports Analytics 4.4 (2018): 263-273.

[10] 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.

[11] Sankaranarayanan, Vignesh Veppur, Junaed Sattar, and Laks VS Lakshmanan. "Auto-play: A data mining approach to ODI cricket simulation and prediction." Proceedings of the 2014 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2014.

[12] Kumar, Sonu, and Sneha Roy. "Score prediction and player classification model in the game of cricket using machine learning." International Journal of Scientific & Engineering Research IJSER 9.8 (2018).

[13] Singh, Tejinder, Vishal Singla, and Parteek Bhatia. "Score and winning prediction in cricket through data mining." 2015 international conference on soft computing techniques and implementations (ICSCTI). IEEE, 2015.

[14] Mustafa, Raza Ul, et al. "Predicting the cricket match outcome using crowd opinions on social networks: A comparative study of machine learning methods." Malaysian Journal of Computer Science 30.1 (2017): 63-76.

[15] Vistro, Daniel Mago, Faizan Rasheed, and Leo Gertrude David. "The Cricket Winner Prediction With Application Of Machine Learning And Data Analytics." International Journal of Scientific & Technology Research 8.09 (2019).

[16] Nimmagadda, Akhil, et al. "Cricket score and winning prediction using data mining." International Journal for Advance Research and Development 3.3 (2018): 299-302.

[17] IPL Live Scores. (n.d.). CricBuzz. Retrieved March 1, 2021, from https://m.cricbuzz.com/

[18] Scikit Learn. (n.d.). Scikit-Learn. Retrieved December 1, 2020, from https://scikit-learn.org/0.21/documentation.html