PREDICTING THE WINNER OF AN ODI AND T20I GAME

A Project Report submitted in partial fulfilment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Dhruv Kumar Patwari, 221710302021

Mekala Muralidhar, 221710302037

P Sai Charan, 221710302047

Parimi Siri Chandan, 221710302048

Under the esteemed guidance of

Mr Arif Mohammad

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GITAM

(Deemed to be University)

HYDERABAD

MAY-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

I/We, hereby declare that the project report entitled **"Predicting the winner of an ODI and T20I game"** is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfilment of the requirements for the award of the degree of B.Tech. In Computer Science and Engineering. The work has not been submitted to any other college or university for the award of any degree or diploma.

Date: May 10, 2021

Registration No(s)	**Name**	**Signature**
221710302021	Dhruv Kumar Patwari	
221710302037	Mekala Muralidhar	
221710302047	P Sai Charan	
221710302037	Parimi Siri Chandana	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

This is to certify that the project report entitled **"Predicting the winner of an ODI and T20I game"** is a Bonafide record of work carried out by **Dhruv Kumar Patwari (221710302021), Mekala Muralidhar (221710302037), P Sai Charan (221710302047), Parimi Siri Chandana (221710302048)**, submitted in partial fulfilment of the requirement for the award of the degree of Bachelors of Technology in Computer Science and Engineering.

Mr. Arif Mohammad

Assistant Professor

Dr. S.Phani Kumar

Professor and HOD

TABLE OF CONTENTS

LIST OF FIGURES vii

1. ABSTRACT 1

2. INTRODUCTION 2

   2.1 OVERVIEW 2

   2.2 NEED ANALYSIS 4

   2.3 PROBLEM DEFINITION 4

   2.4 APPROVED OBJECTIVES 5

   2.5 METHODOLOGY USED 5

   2.6 PROJECT OUTCOMES/DELIVERABLES 6

3. MACHINE LEARNING 7

   3.1 INTRODUCTION 7

   3.2 IMPORTANCE OF MACHINE LEARNING 7

   3.3 USES OF MACHINE LEARNING 8

      3.3.1 Virtual Personal Assistants 9

      3.3.2 Product Recommendations 10

      3.3.3 Online Fraud Detection 10

   3.4 TYPES OF MACHINE LEARNING 11

      3.4.1 Supervised Learning 11

      3.4.2 Unsupervised Learning 12

      3.4.3 Reinforcement Learning 12

3.4.4 Transfer Learning 13
4. FLASK 14
4.1 INTRODUCTION 14
4.2 APPLICATION 14
4.3 DEBUG METHOD 14
4.4 URL BINDING 15
4.5 HTTP METHODS 16
4.6 DEPLOYMENT 17
4.6.1 Externally Visible Server 17
5. ELO RATING 18
5.1 INTRODUCTION 18
5.2 APPLICATION 19
5.3 DRAWBACK 21
5.4 ELO RATING IN THE CURRENT PROJECT 21
6. WORKFLOW DESIGN OF PROJECT 25
6.1 ARCHITECTURE 25
6.2 UML DIAGRAMS 26
7. IMPLEMENTATION 29
7.1 BEAUTIFUL SOUP 29
7.2 IMPLEMENTATION IN CURRENT PROJECT 30
7.3 FEATURE GENERATION 31

7.4 MODEL GENERATION 32

7.5 USER INTERFACE PICTURES 32

8 CONCLUSIONS 34

8.1 SOCIAL BENEFITS 34

8.2 FUTURE DIRECTION 34

REFERENCES 35

LIST OF FIGURES

Fig 2. 1 Formulas To Calculate Strengths 3

Fig 3. 1 Usage Of Machine Learning In Different Fields 8

Fig 3. 2 Uses of Machine learning 11

Fig 3. 3 Types of ML 11

Fig 4. 1 Flask Application Setup 14

Fig 4. 2 URL for routing in Flask app 15

Fig 4. 3 Response from app 16

Fig 4. 4 Different REST methods 16

Fig 5. 1 India's ELO Score since 1971 19

Fig 5. 2 ELO Rating Of all teams 20

Fig 5. 3 Defaults Used 21

Fig 5. 4 Functions Used to calculate ELO Score 22

Fig 5. 5 Calculated ELO Rating 22

Fig 5. 6 ELO Scores in Dataset 23

Fig 5. 7 Players Ranked 24

Fig 6. 1 Architecture 25

Fig 6. 2 Activity Diagram 26

Fig 6. 3 Activity Diagram 27

Fig 6. 4 Use case Diagram 28

Fig 7. 1 Code for importing the libraries 30

Fig 7. 2 Code to scrape match results... 30

Fig 7. 3 First ten rows of the scraped data 30

Fig 7. 4 Save the scraped dataset.. 30

Fig 7. 5 Cleaned Dataset 31

Fig 7. 6 Webapp Landing page 32

Fig 7. 7 With Entered Values 33

Fig 7. 8 Prediction Output 33

1. ABSTRACT

Predicting outcome of a game has taken a front seat since the introduction of statistical modelling. Cricket, being the second most followed sport in the world is a multimillion-dollar industry and predicting the outcome of the same is a lucrative endeavor. Simulating games of cricket and, more specifically, trying to predict the results match, which are played in three formats: test match which is played over 5 days, one-day international, and T20 match has sparked a lot of interest. Each cricket game depends on a lot of factors like pitch condition, weather condition and the type of ball used. This has fueled the interest of many to undertake the challenge to predict the winner of the same.

We use the supervised machine learning approach to predict the results of an international one-day (ODI) cricket game from a team's perspective. We first processed the data for model generation by filling out the missing entries with acceptable values then adjusting their format. We have then created new features like home team benefit, the ability of each team and its success in the last few matches.

The relative strength of the teams involved is a distinctive feature in predicting the winner. According to our findings. Modeling the strength of the team is the basis of our methodology, which models the individual performer batting and bowling results. We use his/her career figures and his/her recent results to model each player. In a batting and bowling averages player-independent variables have been taken into account to predict the result of the match. To shape our data, among other machine classifications, we used decision trees, random forests, logistic regression and ridge regression.

Each model was then determined by dividing the correct number of forecasts into the total number of predictions. We found the most precise model was the ridge regression model of all models. In addition to the variables such as the toss call and match position, we eventually concluded that the prediction is based on the stats of your player as average batting, bowling, and previous matches.

2. INTRODUCTION

2.1 OVERVIEW

Due to its use in sport over decades, statistical modelling has played a major role in on-site performance. After soccer, cricket is the world's second most common sport. Various environmental factors, wide media coverage and a considerable market for betting have all provided compelling reasons for shaping the game from various angles. The complex rules, the abilities and performances of the game on any given day and a number of other natural factors contribute to the definition of the final outcome of a game of cricket.

This makes it extremely difficult to predict the exact result of a game. The three formats where game is played are the test matches, One-Day Internationals, and Twenty-20 Internationals. Our research is focused on the most widely known version of the game, international for one day (ODI). We propose using proficiency and active participation in the latest games to estimate the battering and bowling potential of the 22 players participating in the match to predict the results of ODI cricket matches. One team calculates its relative advantage over the other with these player potentials. In order to forecast the winner by two other basis characteristics, the toss decision and match position as well as the relative team size, we use supervised learning algorithms.

Our project starts with the collection of information from websites like ESPN, Kaggle and Cricksheet. The data we have collected is a csv file with each file detailing the precise details of each correspondence. In order to group the data entries, we then developed python scripts for each match. Then we selected features such as the teams involved, the location, the toss decision and the winner of the match. We used the feature generation to produce more important elements like the strength of each side, the team's performance in recent matches and the chance of team batting first, based on all the matches that had previously been played among both teams.

Fig 2. 1 Formulas To Calculate Strengths

Relative Strength = Strength of Team A – Strength of Team B

Strength of Team = Batting Average of team – Bowling Average of team

Batting Average of Team = $\sum$ (Bat average of each player)/No. of players

Bowling Average of Team= $\sum$ (Bowl average of each player)/No. of players

In terms of relative strength and performance the strength and performance of each match are calculated. In a match, each player's average beats and bowles are used to determine the strength of each team. The power of each team is determined by the average batting of the team. Moreover, the team's average batting is determined by averaging all team players' averages of bats, and the team's average batting is measured by averaging all the team's bowlers' bowling averages.

In order to obtain the relative strength of the game, we draw Team A's strength from Team B. If relative strength is positive, Team A is higher than Team B. This means that team B is stronger than team A if the outcome is negative. Alternatively, both should do the same. We took account of the recent match results of each team, along with strength, as this shows whether or not they are in good form. For measurement of outcomes, the average batting of each team's previous 10 matches is used. Once we have got individual team results, we calculate the relative success of the team by extracting Team A from Team B. If the relative outcome is positive, Team A is in better shape than Team B. Team B is in better condition if the outcome is negative. They are in decent condition otherwise. The size reflects the extent to which one team exceeds the other.

Machine learning was used to create models and predict results after preprocessing and feature generation. We used several models and measured the accuracy of each one before deciding on the one with the best accuracy. Decision trees, random forests, logistic regression, and SVM were all used. We chose logistic regression over the other models because it had the highest accuracy of 65.5 percent. We used the K Fold cross validation technique to create the models, which enabled us to use the entire dataset for both training and testing. We created a GUI in Python using the flack module after creating models so that users can interact with our framework without having to go to the command prompt and type all the fields.

2.2 NEED ANALYSIS

Cricket was the first of the sports to make comparison and illustration of statistics. There was not much mathematical modelling for cricket compared with other sports. A Cricket Prediction Model was developed by Ganeshapillai and Guttag (2013) to decide when the initial pitcher is to be modified in advance. It is very like our system, in which historical data and data are combined to predict the outcome of a bowler.

Tulabandhula and Rudin created a real-time forecasting and judgment method for car racing (2014). The model determines what the correct time is and how many pneumatic changes should be made. Wood (1945) used the geometrical distribution to model the overall score; Kimber and Hansford (1993) recommended a non-parametric approach based on scored runs to evaluate batting performance.

The One Day International is the most popular and famous form of cricket. It is played for 50 overs a side. As in other athletic competitions, winning is the principal goal. Few studies (De Silva, 2001) look at the scale, but most of them concentrate on the winning influences. The Bayesian classificator uses Kaluarachchi and al (2010) to forecast the match's results in light of different factors that affect the game such as home team advantage, day/night impact and toss, among others. Sankaranarayanan et al.(2014) employed an automatic learning algorithm to estimate a one-day match based on past information and game data.

Sohail Akhtar and Philip Scarf (2012) have made session-by-session forecasts for test cricket match outcomes. At the beginning of each session, a number of multinomic logistic regression models will be used to predict the likely result. This odds help a team leader or manager determine in the next session whether to adopt an offensive or defensive batting style. These odds can be improved, however, by taking into account factors such as team selection and performance during the last matches, the average batting and bowling of each player within each team and the winning probability of a team in a particular location against a particular team.

2.3 PROBLEM DEFINITION

Due to its use in sport over decades, statistical modelling has played a significant role in on-site success. After soccer, cricket is the world's second most common sport. Different natural forces, broad media coverage and a considerable demand for betting have all offered convincing explanations for shaping the game from various angles. The complex rules, the skills and outcomes of the game on any given day and a number of other normal variables contribute to the definition of the ultimate result of a cricket match. This makes it incredibly difficult to determine the exact result of a game.

2.4 APPROVED OBJECTIVES

In order to determine the result of a cricket match, consider not only the toss, place, day-to-day but also the team composition, the average batting and bowling per player in each player, the team's performance in their preceding matches and the chance to win first at a particular venue against a particular team.

To help the cricket board of the team choose a certain tournament player.

2.5 METHODOLOGY USED

The project was broken into smaller tasks.

Data preprocessing, feature generation, model creation and testing, and user interface development.

Our project starts with the collection of details from websites like ESPN, Kaggle and Cricksheet. The data we have collected is a csv file with each file detailing the precise details of each correspondence. In order to group the data entries, we then developed python scripts for each match. Then we selected features such as the teams involved, the location, the toss decision and the winner of the match. We then used feature generation to produce more pertinent features, such as the power of each side, recent outcomes, team probability, and a certain team's probability at a certain location depending on some match between the two teams In terms of relative strength and results the strength and performance of each match are calculated. In a match, each player's average beats and bowles are used to determine the strength of each team.

The power of each team is determined by the average batting of the team. Moreover, the team's average batting is measured by averaging all team players' averages of bats, and the team's average batting is estimated by comparing all the team's bowlers' bowling averages. In order to obtain the relative strength of the game, we draw Team A's strength from Team B. If relative strength is favourable, Team A is higher than Team B. This means that team B is stronger than team A if the verdict is unfavourable. Alternatively, both should do the same. We took into account the effects of the past games along with the strength

For measurement of outcomes, the average batting of each team's previous 10 matches is used. Once we have got individual team data, we calculate the relative success of the team by extracting Team A from

Team B. If the relative outcome is positive, Team A is in better condition than Team B. Team B is in poorer condition if the verdict is unfavourable. They are in fine form otherwise. The size reflects the extent of which one team exceeds the other.

Machine learning was used to create the models that predict the results after preprocessing and feature generation. We used several models and measured the accuracy of each one before deciding on the model with highest accuracy score. Ridge Regression, random forests, decision trees, logistic regression, and SVM were all used. We chose ridge regression over the other models because it provided the best accuracy. We used the K Fold cross validation technique to create the models, which enabled us to use the entire dataset for both training and testing. After creating models, we created a graphical user interface (GUI) in Python using the Flask module so that users can interact with our framework without having to go to the command prompt.

2.6 PROJECT OUTCOMES/DELIVERABLES

Learning Machine model that predicts ODI matches results in advance based on variables such as team composition, averages for batting and bowling players in the last ten matches and their success in other parameters, such as toss decision, venue and home team benefit. A Flask GUI for user-friendly interaction with the model. Precision is done as a bar chart of different models.

3. MACHINE LEARNING

3.1 INTRODUCTION

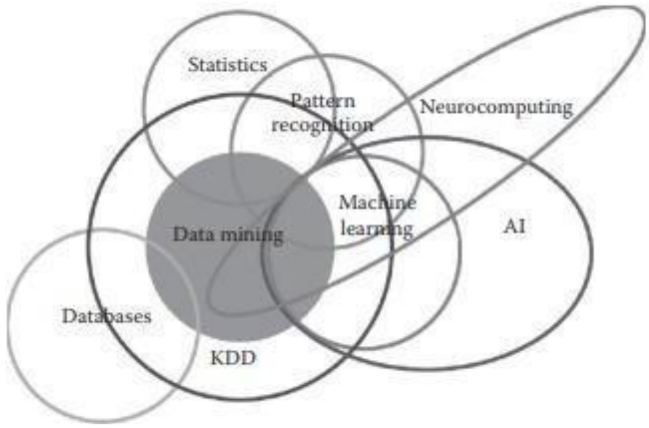
In the last two decades, machine learning has been an essential component, but mainly a mystery, of information technology, and with it it. As data volumes are constantly the, we have cause to conclude that the key element of technical development is intelligent data processing.

Human designers often produce products which are not as effective in their environments as they can. In reality, at the time of design, some elements of the working environment were not well known. Computer training methods may be used to enhance modern flying machine prototypes. For human beings to directly encrypt the amount of knowledge about such events available. Machines which eventually receive this information could catch more than people want to document. Over time, the world changes. The need for continuous redesign is eliminated by machines that can be adapted to their environment. The language is changing. There is an endless flow of new events in the cosmos. It's impractical to keep redesigning AI systems adapt to unknown data, but machine learning methods are not able be able to track a lot of it.

3.2 IMPORTANCE OF MACHINE LEARNING

Machine learning is an artificial inteligence division that uses smart software that makes it possible for computers to do their work effectively. Statistical learning approaches are the basis of intelligent applications for creating machine intelligence. Since machine learning algorithms need data in order to learn, the discipline must be linked to database science. Similarly, words like Knowledge Discovery from Data (KDD), data mining, and pattern recognition are popular. One might be perplexed as to how to interpret the big picture in which such a connection is depicted.

Fig 3. 1 Usage Of Machine Learning In Different Fields



Humans are capable of performing certain tasks with ease or effort, but we are unable to clarify how we do so. For example, we can easily recognize our friends' voices. If you ask us how we understand the voices, we have a tough time explaining it. We can't build algorithms for such situations because we don't understand the phenomenon (in this case, speech recognition). Machine learning algorithms may assist in closing this knowledge gap.

The definition is straightforward. We are not attempting to comprehend the fundamental mechanisms that enable us to learn. We write computer programmes that teach machines how to learn and perform tasks like prediction. Learning's aim is to build a model that takes the input and generates the desired result. Often we can understand the model, but other times it can seem like a black box to us, with no intuitive explanation for how it works. The model can be thought of as a rough representation of the mechanism that we want machines to imitate. It's likely that we'll get errors for some feedback in this case, but the model will almost always have right answers. As a result, accuracy of results will be another metric of a machine learning algorithm's success (in addition to speed and memory usage).

3.3 USES OF MACHINE LEARNING

Artificial Intelligence (AI) is becoming increasingly prevalent. It's possible that you're still using it in some way and aren't even aware of it. One of the most popular implementations of AI is machine training (ML), where computers, software and devices work through cognition (very similar to human brain). Here, we share a few examples of machine learning, which are powered by ML every day, but which we cannot understand. There are a few of the uses and implementations of machine learning.

3.3.1 Virtual Personal Assistants

Siri, Alexa and Google Now, to name a couple, are digital personal assistants. They assist, as the name suggests, if asked for information on the internet. Only trigger them and wonder "What's my schedule today?" or "What's the flight from Germany to London?" Your personal assistant can scan for facts, recall your questions or submit a command to other resources (such as telephone apps) for data collection. You should also give your helpers orders, e.g. "Alert the next morning for 6 o'clock," or "Remember to attend the Visa Office tomorrow after the morning."

Machine learning is a core element of these personal assistants in gathering and refining information on the basis of previous experiences with them. This data collection is then used to produce reports that are tailored to your needs.

Traffic predictions At some time we all used GPS browsing facilities. Our current locations and speeds are registered for traffic control on a central server. This data is then used to construct a traffic diagram. Although this helps to deter traffic and to analyse the traffic jams, the fundamental problem is that the number of GPS cars is restricted. Machine learning helps to estimate regions where congestion is dependent on daily experience.

Online transport networks: The app estimates the fare when you book a taxi. When sharing these facilities, how do you prevent detours? The approach is machine learning. Jeff Schneider, engineer at Uber ATC, has revealed in an interview that he uses machine learning to recognise hours of price rise by forecasting rider requirements. In the whole duty period, ML plays an important part.

Machine learning for its own and user uses is used by social media organisations, from personalising the news stream to enhanced ad targeting. Here are some examples of functionality that you might search, use and appreciate without knowing that they are all machine learning apps in your social media accounts.

You know: Machine learning is based on a fundamental concept: learning from previous encounters. Facebook tracks the individuals with whom you communicate, the accounts you use, your hobbies, your work, or your community, etc. Centered on continuous learning, you propose a list of Facebook users with whom you might become friends.

Face Recognition: Facebook identifies the friend immediately when you share a picture of yourself with a friend. Facebook analyses the positions and the predictions on the frame, determines the features of each person on your friend's list and matches them. The whole backend method is complicated and takes care of the factor of accuracy, but the front end seems to be easy to apply.

Similar Pins: Machine learning is a tool for drawing useful information from videos and photographs, the core feature of computer vision. To recognise the objects in images and then recommend associated pins, Pinterest uses computer vision.

Search engines such as google use machine learning for optimising search results. Whenever you do a search, backend algorithms watch how the answers are replied to. The search engine considers that the answers are important for your query if you open the top product and stay on page for a long period of Zeit. Likewise, the search engine concludes that you have not met the criteria if you reach the second or third page of search results without opening them. The results of the search are thus enhanced with the background algorithms.

3.3.2 Product Recommendations

A few days ago you purchased something online, and now emails are receiving tips for shopping. If not, you may have discovered that such items that match your tastes can be found on the purchasing website or app Product reviews are based on your website/app history, prior orders, items you want or added to your cart, brand tastes, etc.

3.3.3 Online Fraud Detection

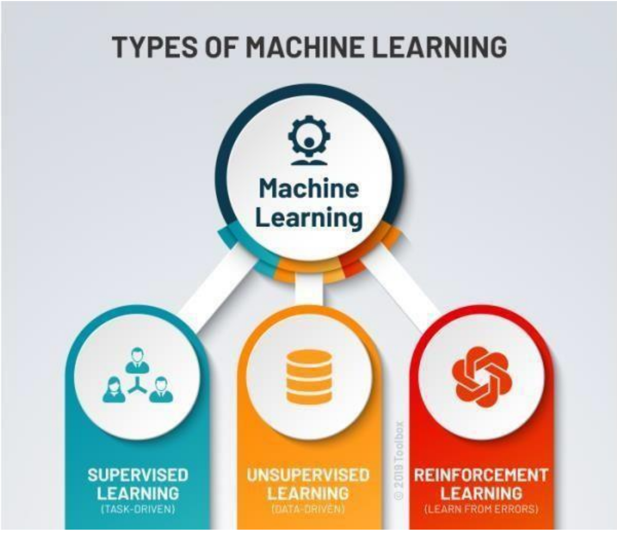
The opportunity to make cyberspace a safer world is shown by machine teaching, including identification of financial fraud online. For instance, Paypal uses money-laundering machine learning. The company uses tools to allow millions of transactions to be compared and legitimate or illegal transactions between buyers and sellers to be distinguished between them.

Fig 3. 2 Uses of Machine learning

3.4 TYPES OF MACHINE LEARNING

The three types of Machine learning which are widely used in today's world; these are as follows.

Fig 3. 3 Types of ML



3.4.1 Supervised Learning

Controlled learning is one of the most fundamental ways of machine learning. The algorithm of the machine learning is trained in this case on labelled results. Although this method requires the proper labelling of results, supervised learning can be very successful when used under the appropriate circumstances. The ML algorithm has a minimal dataset for supervised learning. This data collection for testing is a part of the broader data set. It provides a simple understanding of the situation, solution and data points that are to be managed in the algorithm.

The training data set is very similar to the final data set in terms of characteristics and it gives the algorithm with the marked parameters that it needs to solve the issue. The algorithm then defines an equation between the cause and effect of the variables in the dataset by identifying connections between the defined parameters. At the end of the preparation, the algorithm understands the role of the data and the relationships between input and output.

3.4.2 Unsupervised Learning

Unattended machine learning can be benefited from the freedom to deal with unlabeled data. This makes sure the data collection is not readable by human effort to run the programme on even larger datasets. The algorithms will decide the exact nature of any association between two data points with the labels in controlled learning. Uncontrolled learning does not have to be handled, on the other hand, and hidden mechanisms are created. The algorithm recognises abstract and human-input interactions between data points. The simplicity of these secret architectures is attributed to the building of unattended learning algorithms. In place of a fixed statement of problems, unsupervised learning algorithms can adapt to data by dynamically modifying hidden structures. This makes for more post-deployment development compared to supervised learning algorithms.

3.4.3 Reinforcement Learning

It is motivated directly by the way people learn from data in their everyday lives. It contains an algorithm for trial and error that expands on and learns from new situations. Positive results are celebrated or 'strengthened,' whereas poor results are prevented. Increase learning operates by putting the algorithm in a setting built on the psychological theory of conditioning, with an interpreter and a reward scheme. The result of each algorithm iteration is submitted to the interpreter who decides whether the result is beneficial or not.

The translator confirms it by rewarding the algorithm in the case where the machine seeks the correct answer. The algorithm is used to replicate the procedure before a better answer is obtained if it is not favourable. The reward scheme is directly proportional to the effectiveness of its result in the vast majority of cases. The way to find the shortest route between two points on the map is no absolute value in conventional reinforcement learning situations. Rather, an efficiency score is assigned that is expressed as a proportion rating. The higher the rating, the more compensated is the algorithm.

3.4.4 Transfer Learning

Transfer education is defined as the opportunity to integrate a pre-trained model with personalised training results. This means you can use the models features and apply your own samples instead of trying to rebuild them from scratch.

An image classification model, for instance, was trained with thousands of pictures, which helps you to mix new custom image spectrums with the pre-trained image model to produce a new image classification instead of making your own. With this function you can quickly and easily build a customised classifier.

There are several Maschine Learning (ML) models educated in a user-friendly class by people around the world. They are also concentrated on recent research. This is beneficial, since the challenge can be solved instead of how a model can be trained and data generated. Any of the advantages of a pretrained model are as follows: Training data cannot be collected your own; data in the right and labelled format can be time-consuming and costly for a machine learning algorithm to learn about it.

Rapidly prototyping a concept with less time and resources - a pre-trained model will be good enough for you, so you should not reinvent the wheel so you can rely on the expertise of the model to carry out your creative project.

Using specialised tests—as these examples mostly rely on popular research, you will get a greater idea of how they perform in the real world and will have more visibility with them.

These variants are generally easier to use and it's well known due to their popularity.

Any pre-trained models are capable of understanding. This is the method of taking and adding what you've learned from one machine. You can retrain to recognise dogs if you gave a model which was originally taught to identify new cats' training data. Since you don't start from a blank canvas, you can practise this even faster. The model should add the knowledge that cats have learned to recognise the new thing - dogs have eyes and ears, but when they know where to find those characteristics, we are in the middle of it. Retrieve the model in a fraction of the time using your own results.

4. FLASK

4.1 INTRODUCTION

Flask is a Python-built web server background. It was founded by Armin Ronacher the founder of the international Python community Pocco. The tool kit and the Jinja2 template engine are integrated into Flask. Both projects are Pocco.

4.2 APPLICATION

To test your Flask installation, open the editor and type Hello.py into it.

Fig 4. 1 Flask Application Setup

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
   return 'Hello World’

if __name__ == '__main__':
   app.run()
```

Importing the flask module into the project is needed. The WSGI programme we're working on is a Flask type object. The current module's name (__name__) is passed to the Flask function as an argument. In the Flask class, the Route() function is a decorator that tells the application which URL to call a corresponding function.

4.3 DEBUG METHOD

A Flask programme is started with the run () process. However, after each code update, the programme should be manually restarted during development. Enable debug assistance to avoid this inconvenience. If the code changes, the server will restart. It also provides a debugger to monitor possible errors in construction of the programme.

The debug mode is permitted by setting the application's debug attribute to True before the application is executed or bypassing debug parameter to execute ().

4.4 URL BINDING

The URL for () feature comes in handy if you need to generate a URL for a particular function on the fly. The function accepts one or more keyword arguments, each of which refers to a separate URL component, and the first argument is the function name.

The URL for the () function is shown in this script.

Fig 4. 2 URL for routing in Flask app

```
from flask import Flask, redirect, url_for
app = Flask(__name__)

@app.route('/admin')
def hello_admin():
   return 'Hello Admin'

@app.route('/guest/<guest>')
def hello_guest(guest):
   return 'Hello %s as Guest' % guest

@app.route('/user/<name>')
def hello_user(name):
   if name =='admin':
      return redirect(url_for('hello_admin'))
   else:
      return redirect(url_for('hello_guest',guest = name))

if __name__ == '__main__':
   app.run(debug = True)
```

The above script's user(name) function takes a value from the URL as an argument.

The user () feature decides whether or not a statement belongs in the category of 'admin.' If the application fits, the guest parameter will be used for the function hello admin () by the URL for ()and the guest () hello function if not.

Save and execute the Python shell code above.

Type http://localhost:5000/user/admin in your browser's address bar.

The browser application response is

Fig 4. 3 Response from app

```
Hello Admin
```

Enter the following URL in the browser − **http://localhost:5000/user/mvl**

The application response now changes to −

```
Hello mvl as Guest
```

4.5 HTTP METHODS

The basis of the Internet data sharing is the HTTP protocol. A variety of methods to extract information from URLs are described by this protocol.

The various HTTP processes are summarised in the table below.

Fig 4. 4 Different REST methods

Sr.No.	Methods & Description
1	**GET** Sends data in unencrypted form to the server. Most common method.
2	**HEAD** Same as GET, but without response body
3	**POST** Used to send HTML form data to server. Data received by POST method is not cached by server.
4	**PUT** Replaces all current representations of the target resource with the uploaded content.
5	**DELETE** Removes all current representations of the target resource given by a URL

The GET requests are answered by default by the flask direction. However, the method argument to the route () decorator will change this choice.

4.6 DEPLOYMENT

4.6.1 Externally Visible Server

Only the computer on which the programming environment is mounted will access Flask applications on the development server. This is the default behaviour as a user may execute arbitrary code in the debug mode on the cpu.

Setting the hostname to 0.0.0.0 when debug is deactivated renders the development server accessible to the network user on the local machine.

There are several options to deploy a mobile server Flask, depending on what you have. An application must be build on a real web server to move from a development to a comprehensive production environment.

All of the sites below, all of which provide a free plan for small apps, can be used to implement a small application.

On these cloud networks, Flask systems may be deployed. Flask applications can also be run on the cloud infrastructure of Google. Use the LocalTunnel utility without using the DNS or firewall settings to share the programme on localhost.

You have a few choices if you want to use a specific web site instead of one of the shared sites listed above.

Mod wsgi is a WSGI compliant Apache module for hosting Web applications based in Python on an Apache server.

Developing wsgi file The mod wsgi code, running to get the application object, is included in that file.

You must inform mod wsgi where your programme is located when setting up Apache.

5. ELO RATING

5.1 INTRODUCTION

We can quickly review the ELO rating system's regular as well as a version. Each group will be awarded an ELO rating to assess its current intensity based on the results of a number of previous matches. Let H 0 and A 0 represent the latest ratings of the home and remote teams at the beginning of the match. ELO teams thus say that they score H and A on average respectively with H = 1 1 + c(A 0 H 0,0)/d and A = 1 H = 1 1 + c (H 0 A 0), A = 1 H + c(H 0 A 0)/d and A = 1 H + c(H 0,A 0,2) /d, for the match concerned. Let H = if the home team wins or 0.5 if the match is tied to the real home team result.

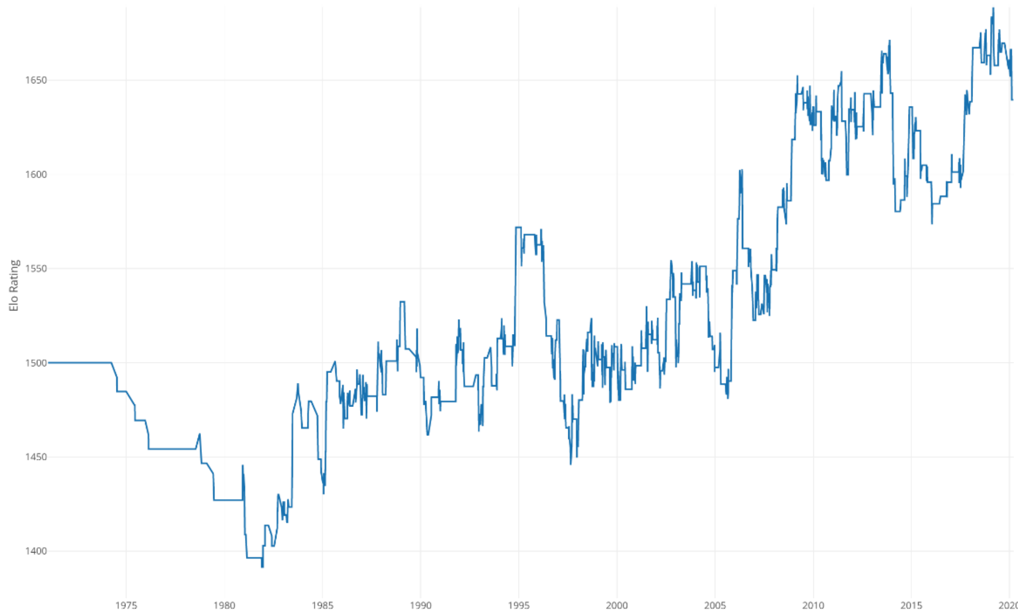
The real score of the remote team is A = 1 H. The ELO ratings have been modified after the match and the current home team ranking is H 1 = H 0 + k (HH). (a) The following: (a) The current ranking of the remote team, A 1, is equally calculated. Post - match and follow the squad, the ranking is adjusted. It should be noted that after a number of matches, determining ELO scores requires some initial ratings for each side.

The scores cannot be said to be exact force measures unless appropriate prior match outcomes are taken into consideration. The rating system referred to above is known as the fundamental ELO metric. The result is a curious version of a basic ELO ranking which can be used for football, awarding a 3–0 win stronger than a 2–1 victory, allowing the rating upgrade coefficient k to reflect on the target gap. One choice is to use k (1) as a fixed parameter and k0 > 0 as an absolute goal difference, with the expression k = k0(1 +), k0 > 0 as the setting parameters. This approach will be defined by the goal-based ELO ranking.

There are three criteria in the standard ELO ranking system: c, d, and k. If c and d could be viewed as an appropriate rating scale, k must be carefully selected. A team ranking will not change as quickly as possible when its performance increases if the value of k is too low, while if k is too high, the rating will differ too much from match to match. In the objective ELO ranking, there are four conditions, with k replaced with k0 and.

This is how India's graph looks after Elo scores are applied to international cricket matches since 1971.

Fig 5. 1 India's ELO Score since 1971



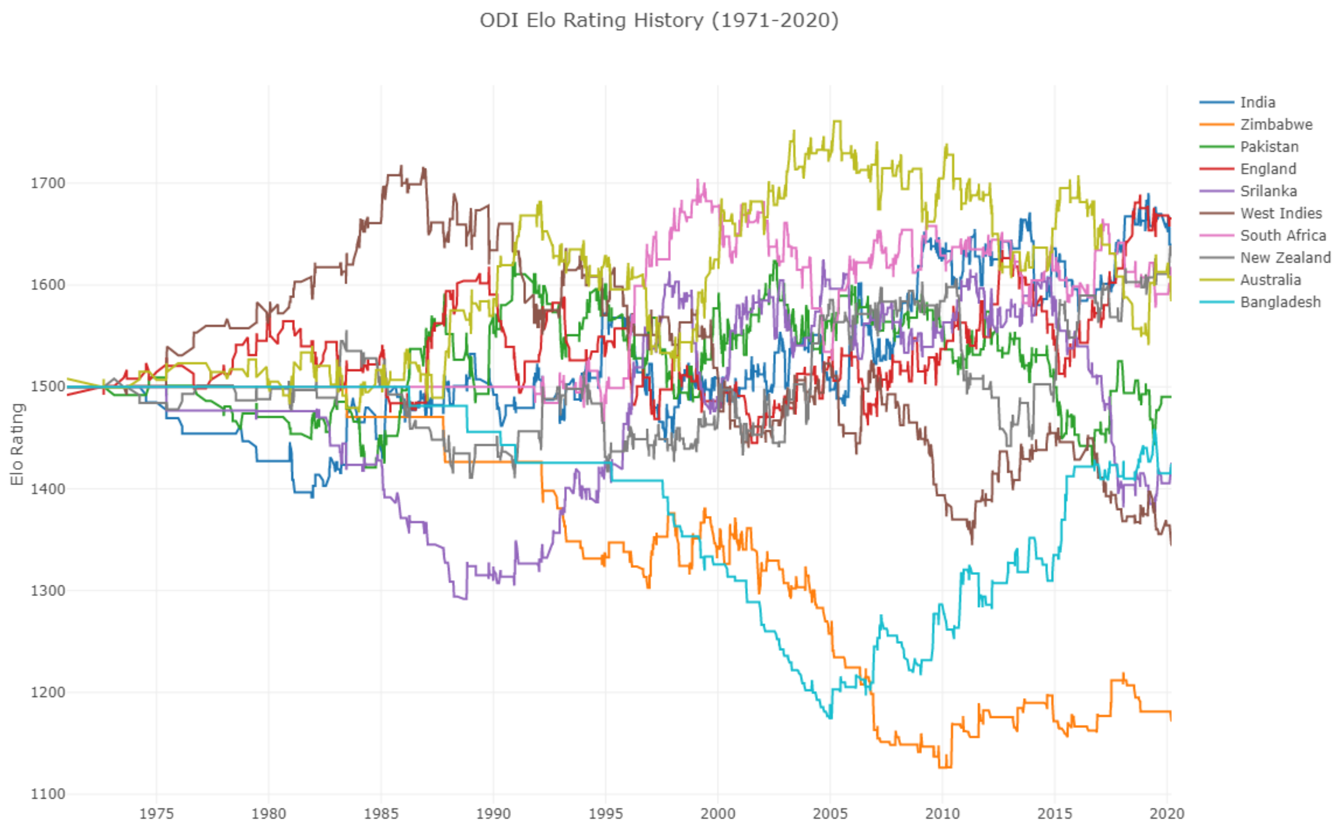
5.2 APPLICATION

A system for measuring players' relative capabilities in non-existent games such as chess, created by Arpad Elo a Hungarian-America physics professor. The Elo Rating method can be understood in principle quite well. The average ranking begins for each match. The winning player removes points from the losing player's score, which is then used to decide how well the next game will be won. The player is an international cricket team member in this situation.

Before we go any further, let's clarify a few points about the analysis that will be discussed.

Only matches between the 10 ODI teams with a significant background in ODI cricket have been considered: Australia, England, New Zealand, Bangladesh, Pakistan, South Africa, Sri Lanka, West Indies, India and Zimbabwe. According to the consented results, these top 10 teams played 3486 completed ODIs. We're losing out on 350 One-Day Internationals (ODIs) played against these top 10 contenders by other ODI teams. (However, one of the top ten teams took 85 per cent of the games.) Each team will begin the journey with 1500 rating points in this Elo Rating review. The ELO rating algorithm employs the K-Factor, a multiplication factor that decides rating fluctuations. K has a significant impact on the ratings if it is set too high. Minor adjustments over time if K is set too low. A K-Value of 16 was used in this case. (Similar research that used K-Value 24 yielded similar results.)

Fig 5. 2 ELO Rating Of all teams



A few simple observations that you might already be aware of:

In terms of global domination over a long period, no one comes close to Australia. The death of West Indie is heartbreaking. The downward spiral that started in 1987 continues unabated. They are currently just better than Zimbabwe. Until 2015, England had been a below-average ODI team, with a few exceptions. In the four years leading up to the 2019 World Cup, they went from being a below-average team to one of the top four (tied actually but anyway)

5.3 DRAWBACK

The downside of this rating system is that it does not consider player ratings or the atmosphere of the match. If a team sends a B-team on tour or rests any of their key players for a game and loses, they will still be penalized based on their Elo rating at the time of the match.

Take, for example, India's 2010 visit to Zimbabwe. At the time, India was one of the top-ranked teams, while Zimbabwe was ranked last. As a result, India chose to rest a few key players for the tour, despite Suresh Raina's leadership. As a result, India's top two shocks came in the series against Zimbabwe. The next one, though, is this Asia Cup match in 2012. It was a game that will be remembered for a long time. In this match, Sachin Tendulkar scored his 100th international century. India, on the other hand, came up short.

5.4 ELO RATING IN THE CURRENT PROJECT

ELO rating was used in this project to provide another parameter for comparing the teams currently playing. All teams start to win an ELO of 1500 and then go up or down depending on their performance. The maximum point a team can win or lose in a game is 400 points, and the factor by which their points are reduced or increased is 22.

Fig 5. 3 Defaults Used

```
mean_elo = 1500
elo_width = 400
k_factor = 22
```

Using these defaults, we calculated the ELO scores of each team after every game they played. Using the functions we wrote, we were able to calculate the ELO Scores of each team after each game. The functions are modeled after the formula mentioned above. The functions used can be found in Fig. 5.4. Theses functions are then run on the data set we scraped from the web and it then calculates the ELO Scores.

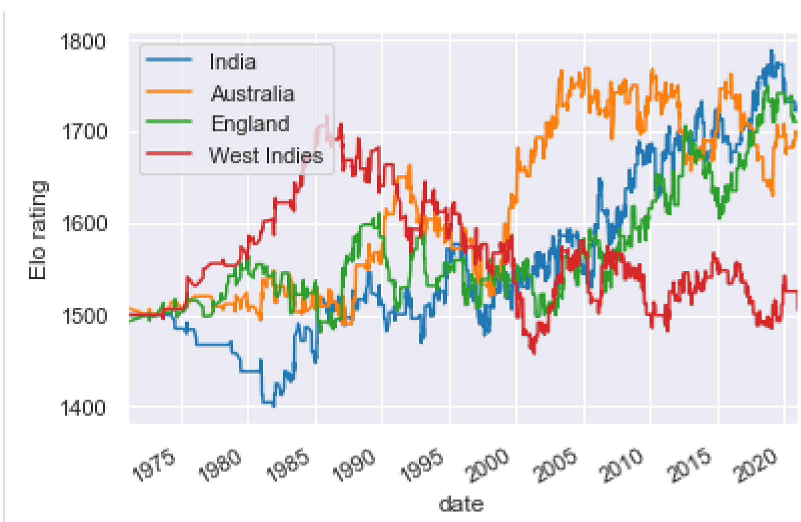
After the ELO Scores are calculated, we graphed the scores of a few teams to see how the performed over the years. The graph is in Fig 5.5. The graph clearly shows that India started off as a weak team and is now one of the strongest teams. And West Indies, on the other hand, has gone in the opposite direction.

The domination of Australian cricket is also clearly visible in the graph. This proves that the calculations performed are good and that we were able to reproduce the observations that we have had over the years.

Fig 5. 4 Functions Used to calculate ELO Score

```
def winMarginRuns(winmargin, runsPerWicket, winner_elo, loser_elo):
    points = (((winmargin/runsPerWicket)+3)**0.8)/(7.5 + (0.006*(winner_elo - loser_elo)))
    return points

def winMarginWickets(winmargin, winner_elo, loser_elo):
    points = ((winmargin+3)**0.8)/(7.5 + (0.006*(winner_elo - loser_elo)))
    return points

def update_elo_wickets(winner_elo, loser_elo, winMargin):
    expected_win = expected_result(winner_elo, loser_elo)
    change_in_elo = k_factor * (1-expected_win)
    change_in_elo *= winMarginWickets(winMargin, winner_elo, loser_elo)
    winner_elo += change_in_elo
    loser_elo -= change_in_elo
    return winner_elo, loser_elo

def update_elo_runs(winner_elo, loser_elo, winMargin, runsPerWicket):
    expected_win = expected_result(winner_elo, loser_elo)
    change_in_elo = k_factor * (1-expected_win)
    change_in_elo *= winMarginRuns(winMargin, runsPerWicket, winner_elo, loser_elo)
    winner_elo += change_in_elo
    loser_elo -= change_in_elo
    return winner_elo, loser_elo

def expected_result(elo_a, elo_b):
    expect_a = 1.0/(1+10**((elo_b - elo_a)/elo_width))
    return expect_a
```

Fig 5. 5 Calculated ELO Rating



In Fig. 5.6, we have shown how it actually looks in the database to each team. Using this function, we then rank the players in the 2020 IPL as it is a small sample size, and we can easily calculate the same. We are doing this for all the players who have ever played needs a lot of resources which we don't have at our disposal at the moment. The players ranked can be seen in Fig. 5.7.

Fig 5. 6 ELO Scores in Dataset

Ground	Match Date	Runs per Wicket	w_elo_before_game	w_elo_after_game	l_elo_before_game	l_elo_after_game
Abu Dhabi	Jan 24, 2021	43.250000	1440.000000	1445.000000	1449.000000	1445.000000
Chattogram	Jan 25, 2021	29.625000	1566.000000	1573.000000	1518.000000	1510.000000
Abu Dhabi	Jan 26, 2021	26.105263	1445.000000	1454.000000	1445.000000	1435.000000

Fig 5. 7 Players Ranked

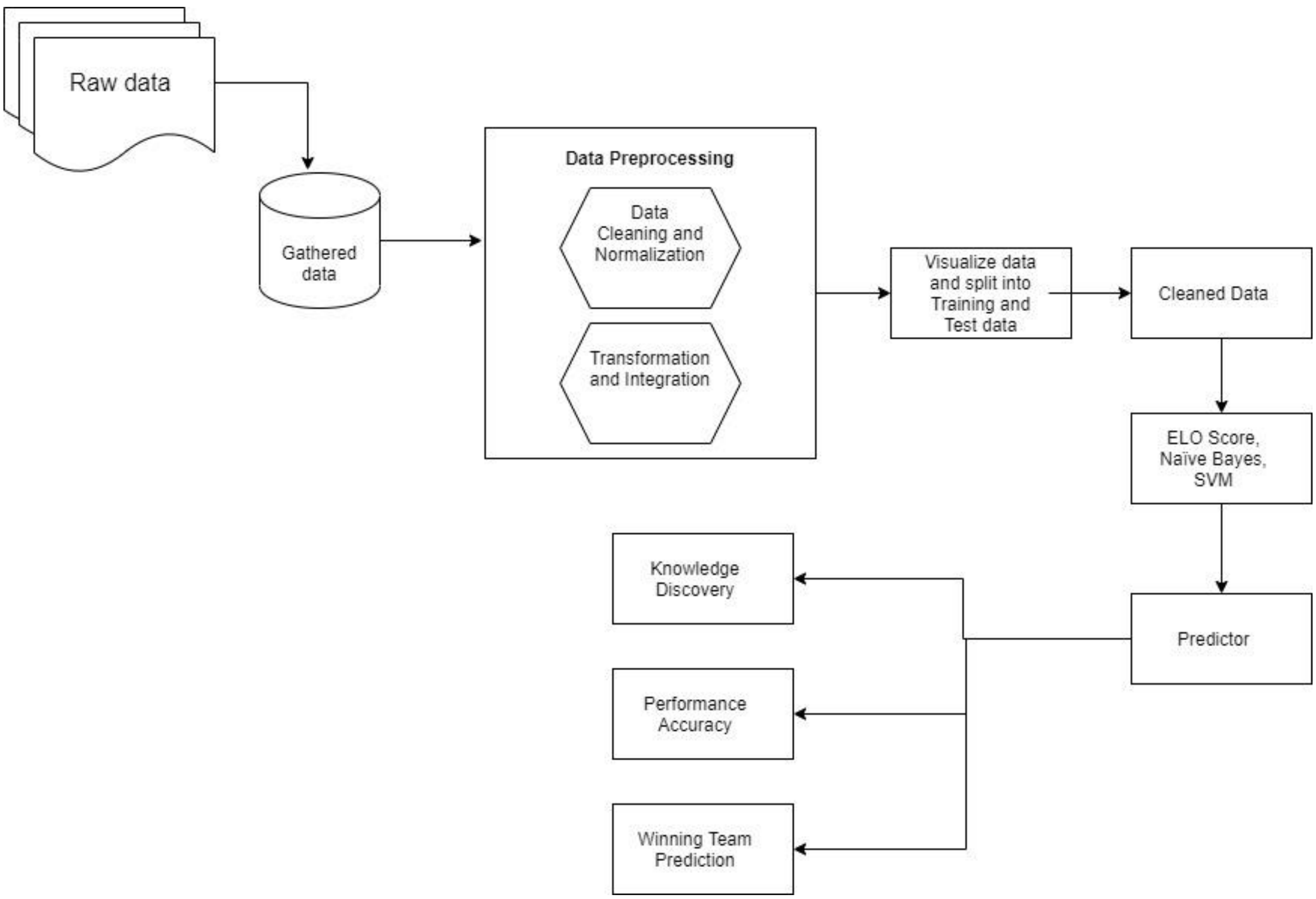
Rank	Player	Team	RAA	Wins	EFscore
1	JJ Bumrah	Mumbai Indians	379	1.281	0.238
2	KL Rahul	Kings XI Punjab	330	1.113	0.194
3	K Rabada	Delhi Capitals	320	1.082	0.232
4	Ishan Kishan	Mumbai Indians	261	0.881	0.157
5	S Dhawan	Delhi Capitals	249	0.842	0.168
6	TA Boult	Mumbai Indians	230	0.777	0.177
7	AB de Villiers	Royal Challengers Bangalore	227	0.767	0.142
8	SA Yadav	Mumbai Indians	225	0.759	0.137
9	JO Holder	Sunrisers Hyderabad	181	0.613	0.132
10	DA Warner	Sunrisers Hyderabad	178	0.602	0.142
11	YS Chahal	Royal Challengers Bangalore	177	0.599	0.156
12	MA Agarwal	Kings XI Punjab	169	0.57	0.117
13	F du Plessis	Chennai Super Kings	165	0.558	0.123
14	WP Saha	Sunrisers Hyderabad	157	0.531	0.086
15	JC Archer	Rajasthan Royals	156	0.528	0.189
16	V Kohli	Royal Challengers Bangalore	154	0.52	0.118
17	Rashid Khan	Sunrisers Hyderabad	152	0.513	0.165
18	EJG Morgan	Kolkata Knight Riders	134	0.454	0.126
19	Shubman Gill	Kolkata Knight Riders	128	0.432	0.109
20	KS Williamson	Sunrisers Hyderabad	123	0.415	0.092
21	SS Iyer	Delhi Capitals	122	0.412	0.125
22	RD Gaikwad	Chennai Super Kings	119	0.402	0.082
23	CH Gayle	Kings XI Punjab	113	0.381	0.087
24	CV Varun	Kolkata Knight Riders	109	0.368	0.135

6. WORKFLOW DESIGN OF PROJECT

6.1 ARCHITECTURE

The architecture of the project is as follows. First, we collect all the data required for the project and then we clean and process the data into a usable form. This is then visualized to find out any relations that we have missed, and then we find the ELO Score of teams and players and add the same to the database. Then we feed this data into an ML model. Before which we split the data into test and train datasets for post validation.

Fig 6. 1 Architecture



6.2 UML DIAGRAMS

Fig 6. 2 Activity UML Diagram

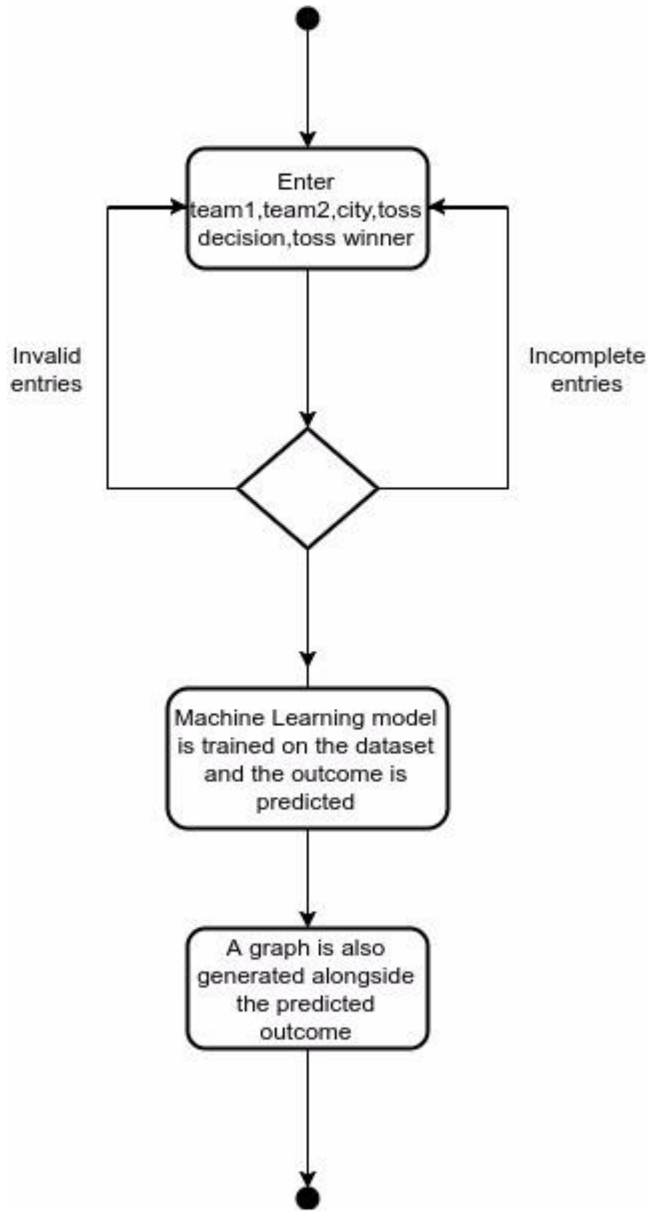


Fig 6. 3 Sequence UML Diagram

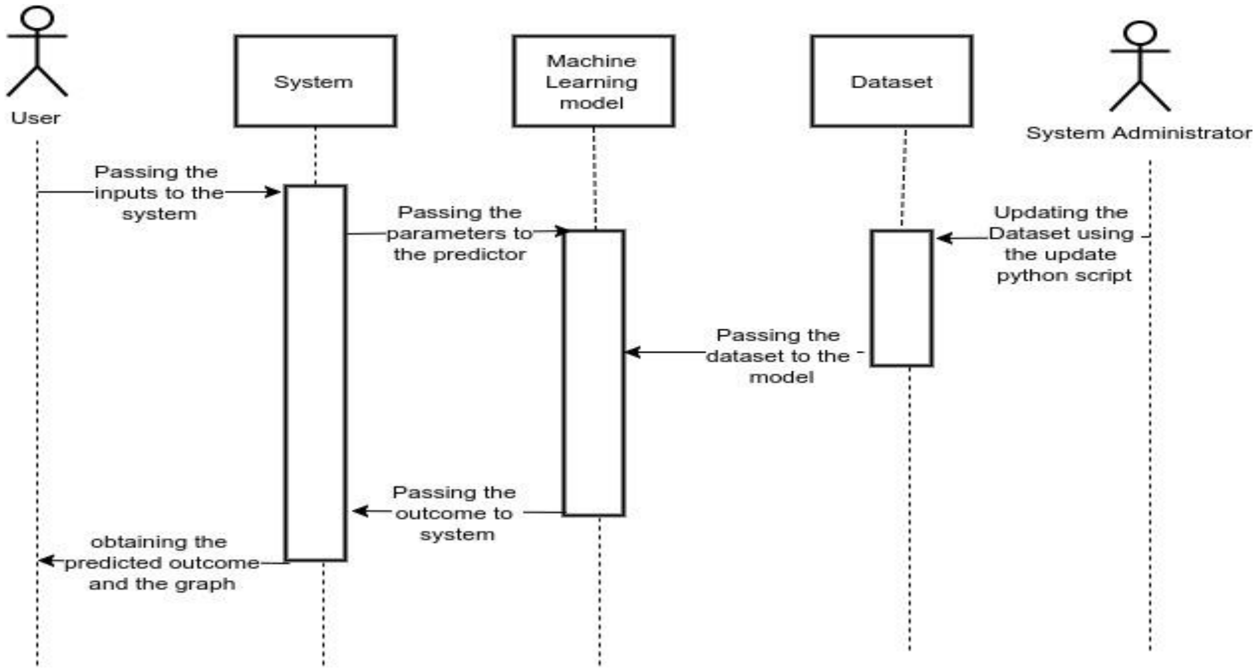
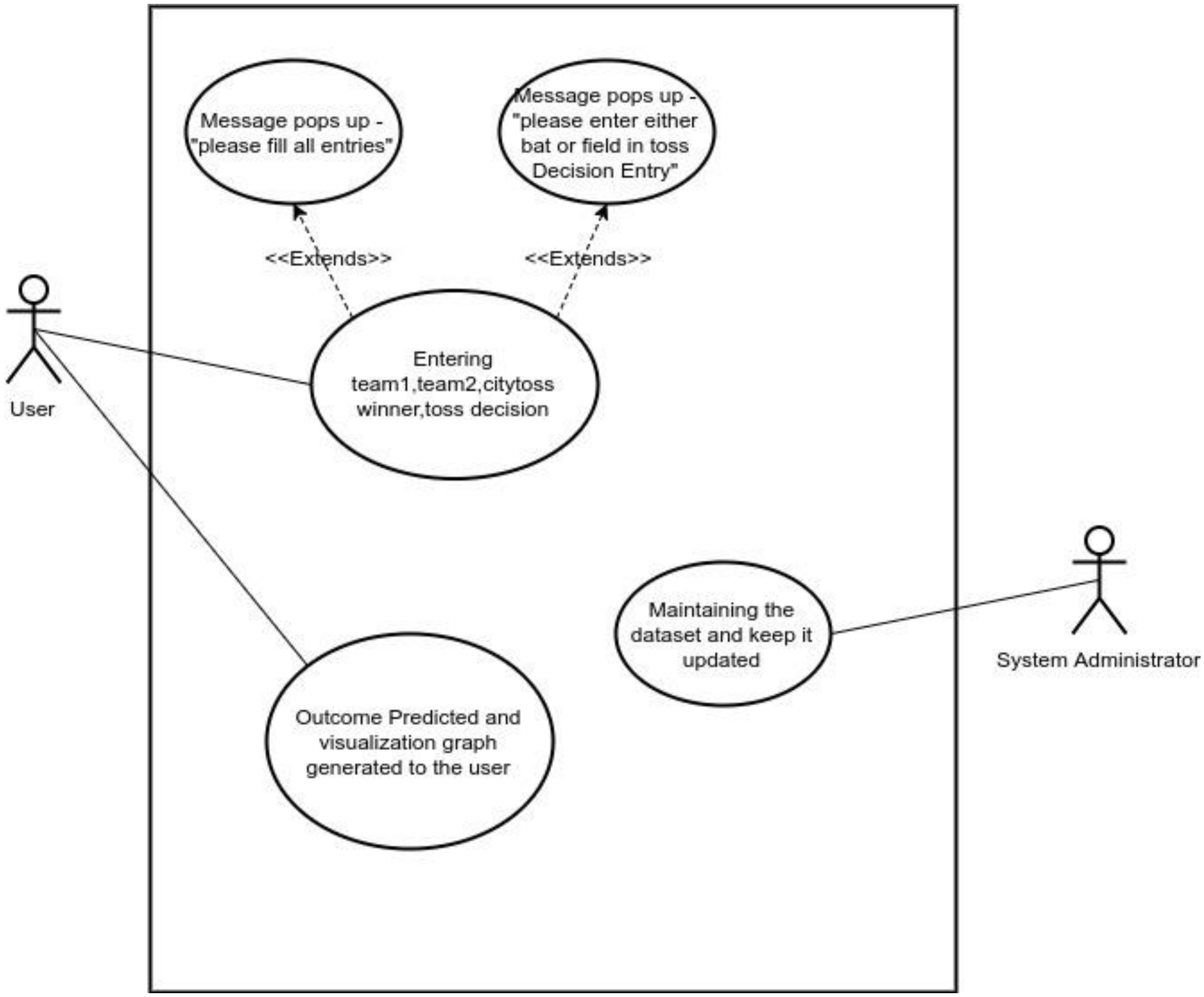


Fig 6. 4 Use case Diagram



7. IMPLEMENTATION

The data used for this project was scraped from various websites using Beautiful Soup 4. The following is a brief introduction to Beautiful Soup 4.

7.1 BEAUTIFUL SOUP

You didn't write this terrible page. You are just attempting to get some facts. Beautiful soup is happy to help. Since 2004, programmers have saved hours or days of work with limited processing time from screen scraping projects.

A nice soup is a Python library for projects, including screen scraping, where quick turnaround is required. Three reasons make it vital:

Lovely soup is a toolkit with a few simple methods and Pythonic idioms for the search, search and modification of the parsley tree to dissect a file. No huge amount of code is needed to write an application.

Documents received will immediately be translated to Unicode and documents released will be converted by Beautiful Soup to UTF-8. Decoding is not required until one is stated in the text and one cannot be detected by Beautiful Soup. You just have to settle on the initial encryption after that.

It is a Python parser with regular Python parsers, such as lxml and html5lib, which enable you to play with a variety of parsing or business speeds.

Beautiful soup scans all that you offer and cares for the crossing of the tree. The following commands are: "Find All links," "Find all links in the external class" and "Find all links whose URLs correspond to "foo.com," or "Find the table with a bold text heading, and then send me the text"

You can also access important data previously cached behind web pages that have been incorrectly built. Projects that take hours to finish in minutes with Beautiful Soup.

7.2 IMPLEMENTATION IN CURRENT PROJECT

Fig 7. 1 Code for importing the libraries

```
import requests
import pandas as pd
from bs4 import BeautifulSoup as bs
import numpy as np
```

```
year = 1971
```

```
link = 'https://stats.espncricinfo.com/ci/engine/records/team/match_results.html?class=2;id='+str(year)+';type=year'
```

Beautiful soup lets us loop over the links and scrape data. This makes it easy to gather large amounts of data in a short time. The following is a snippet of the code used to scrape the ODI match records from the ESPN Cricinfo website.

Fig 7. 2 Code to scrape match results

```
for year in range(1972, 2022):
    link = 'https://stats.espncricinfo.com/ci/engine/records/team/match_results.html?class=2;id='+str(year)+';type=year'
    df = pd.read_html(link)[0]
    print(year)
    masterDF = pd.concat([masterDF, df])
```

Fig 7. 3 First ten rows of the scraped data

Team 1	Team 2	Winner	Margin	Ground	Match Date	Scorecard
Australia	England	Australia	5 wickets	Melbourne	Jan 5, 1971	ODI # 1
England	Australia	England	6 wickets	Manchester	Aug 24, 1972	ODI # 2
England	Australia	Australia	5 wickets	Lord's	Aug 26, 1972	ODI # 3
England	Australia	England	2 wickets	Birmingham	Aug 28, 1972	ODI # 4
New Zealand	Pakistan	New Zealand	22 runs	Christchurch	Feb 11, 1973	ODI # 5
England	New Zealand	England	7 wickets	Swansea	Jul 18, 1973	ODI # 6
England	New Zealand	no result	NaN	Manchester	Jul 20, 1973	ODI # 7
England	West Indies	England	1 wicket	Leeds	Sep 5, 1973	ODI # 8
England	West Indies	West Indies	8 wickets	The Oval	Sep 7, 1973	ODI # 9
New Zealand	Australia	Australia	7 wickets	Dunedin	Mar 30, 1974	ODI # 10

Fig 7. 4 Save the scraped dataset

```
masterDF.to_excel('result.xlsx')
```

This data is then cleaned. The teams are given a unique number used to identify them, and each venue is also given a unique number. This is done to bring the dataset to a version where there are no categorical variables present. This helps in the better computation of the required row.

Fig 7. 5 Cleaned Dataset

id	Season	city	date	team1	team2	toss_winn	toss_decis	result	dl_applie	winner	win_by_ru	win_by_w	Strength T	Strength T
335982	2008	1	18-04-2008	1	5	1	1	normal	0	5	140	0	1.139	0.987
335983	2008	2	19-04-2008	2	8	8	2	normal	0	8	33	0	1.141	1.05
335984	2008	3	19-04-2008	3	6	6	2	normal	0	3	0	9	1.292	0.987
335985	2008	4	20-04-2008	4	1	4	2	normal	0	1	0	5	1.378	1.139
335986	2008	5	20-04-2008	5	7	7	2	normal	0	5	0	5	0.987	1.288
335987	2008	6	21-04-2008	6	2	2	2	normal	0	6	0	6	0.987	1.141
335988	2008	7	22-04-2008	7	3	7	2	normal	0	3	0	9	1.288	1.292
335989	2008	8	23-04-2008	8	4	4	1	normal	0	8	6	0	1.05	1.378
335990	2008	7	24-04-2008	7	6	6	1	normal	0	6	0	3	1.288	0.987

7.3 FEATURE GENERATION

We also created more important features in this phase, such as strength of the side, success in recent matches, the chance of first batting teams and the chance to win a certain team at a certain place on the basis of all previous games between the teams on that particular site. In terms of relative stability and efficiency, the strength and performance of each match are calculated. Each player's batting and bowling average in a given match is used to decide each team's ability. The power of each team is calculated by excluding the team's average batting from its average bowling. In addition, by averaging all team members' batting averages, the total team batting can also be calculated. The average team bowling is determined by averaging all team bowlers' bowling averages.

We take Team A's strength from Team B's strength in order to achieve the total strength. If relative strength is good, Team A is better than Team B. This means that team B would be stronger than team A if the verdict is unfavourable. Otherwise, both would do the same. In addition to power, we consider the recent match results each team has achieved, as this shows whether they are in good shape or not. For the measurement of outcomes, the score beat average of each team in the previous ten matches. After we achieve the outcomes of each team, we calculate the relative success of the team by extracting the performance of Team A from team B. If the relative outcome is positive, Team A is better than Team B. Team B is in poorer condition if the verdict is unfavourable. They are in fine shape, otherwise. The size represents the extent to which one team exceeds the other.

7.4 MODEL GENERATION

We used machine learning to build models which can be used after preparation and generation of features to predict outcomes. We also used various models of machines, calculated each one's accuracy and then selected the one with the highest precision. Decision trees, random woodland, logistical regression and regression of Ridge have also been employed. In contrast with the other models we selected ridge regression because it was more exact. We have used the K Fold validation technology to produce the models to use the whole dataset for training and research. We will first collect information from the user through a Flask-based GUI for potential match predictions. Then we turn those inputs into numeric forms, such as 1 for Team A and 0 for Team B, and generate other key features using the modules used in stage 2. We then transform all information into a data frame for pandas, which we convert to the prediction model. The model calculates and predicts the winner of the match.

7.5 USER INTERFACE PICTURES

Fig 7. 6 Webapp Landing page

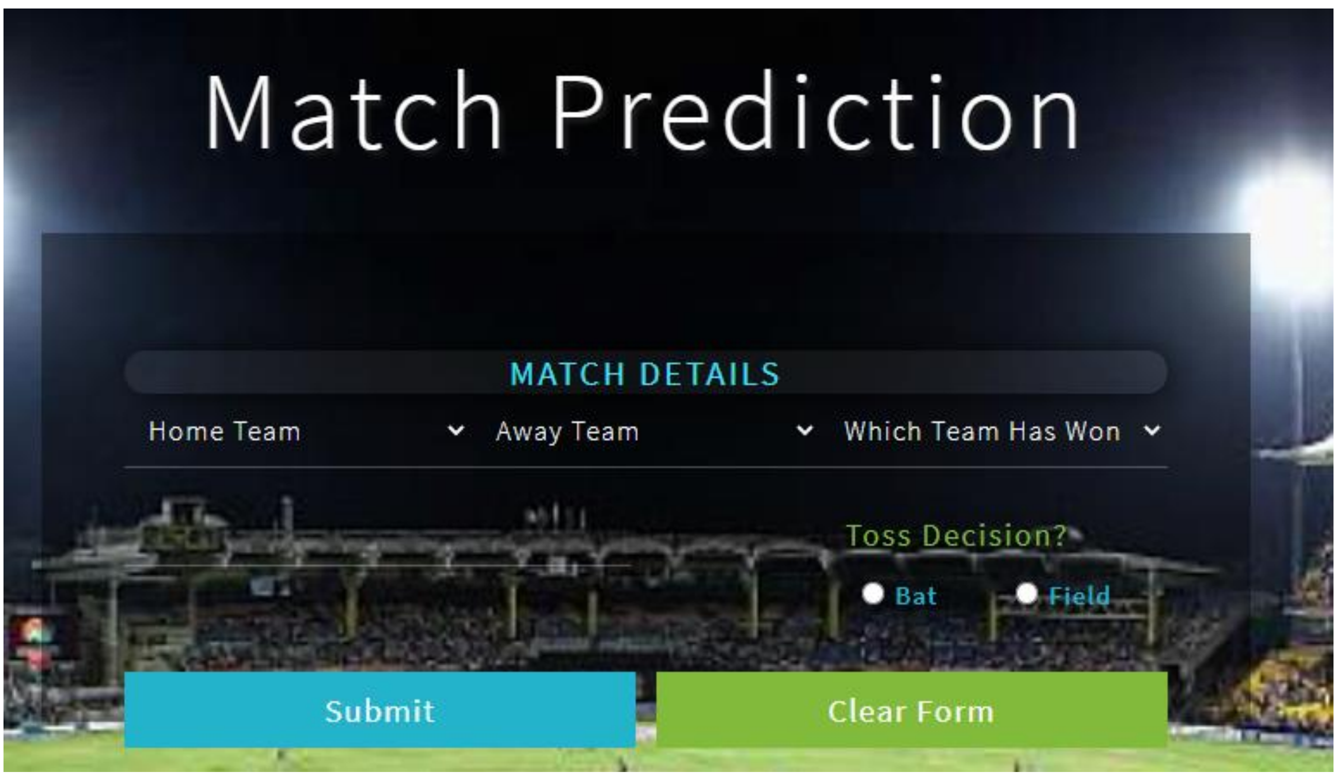


Fig 7. 7 With Entered Values

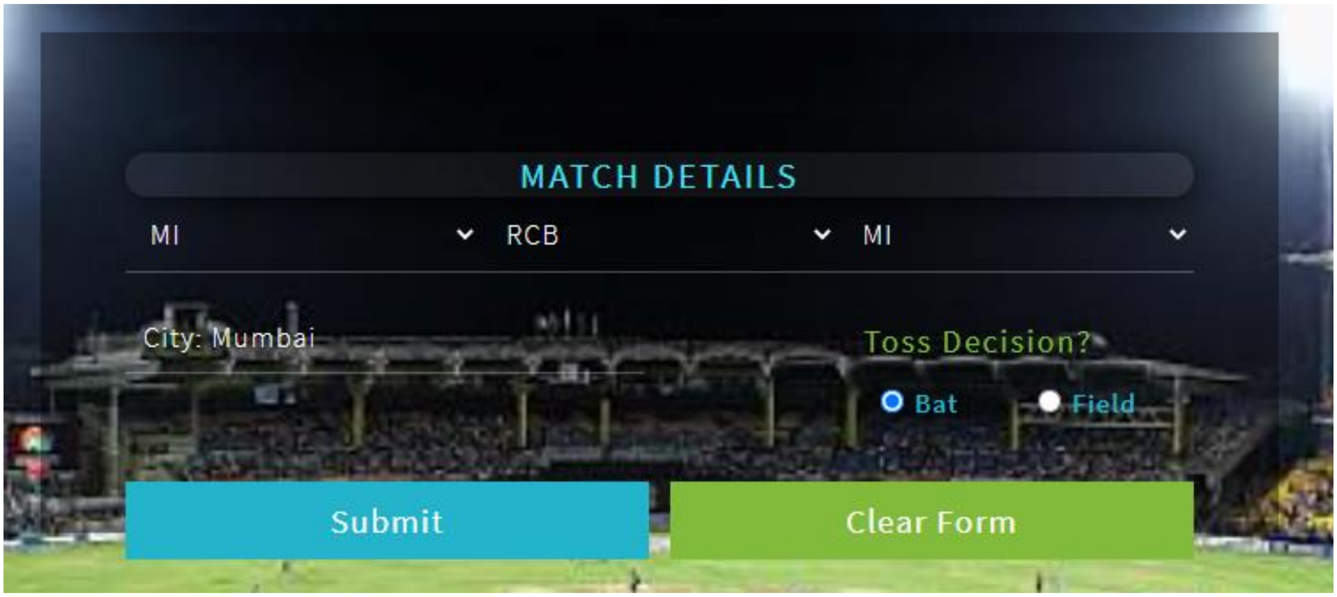


Fig 7. 8 Prediction Output

8 CONCLUSIONS

In recent years, with the the popularity of cricket, a system is required to forecast a match in advance. While there are various instruments on the market, they do not have the precise nature of the variables. Our project seeks to incorporate crucial considerations, such as squad size, player performance in recent games, the bat and bowling averages in each of the teams and the probability that a team will be battling against a given foe at a particular location. The project aims to give current players an edge. Both these important aspects were taken into account as well as the problem and position and a classifier which provides better results was created.

8.1 SOCIAL BENEFITS

Even before the match begins, cricket fans will figure out who has the best chance of winning.

By determining which team composition has the best chance of winning, the Cricket Board will pick individual players for the upcoming tournament.

If a team wins the toss, it may also help captains decide whether to bat or field at a particular stadium.

8.2 FUTURE DIRECTION

The project currently includes key features from past matches and predicts the results before the game starts. However, the present match can be used to forecast the result. In subsequent experiments, however, pre-existing data would be combined with ongoing matching data to produce even better results. It is also possible to extend the projection to include not only the player, but also the forecast runs scored by both teams.

REFERENCES

A. Journals/Articles

1. A. Bandulasiri, “Predicting the Winner in One Day International Cricket”, *Journal of Mathematical Sciences & Mathematics Education*, **Vol. 3, No. 1, pp 6-17, (2008).**
2. L.M. Hvattum, H. Arntzen, “Using ELO ratings for match result prediction in association football”, *International Journal of Forecasting,* **Norway Molde University College, Norway, 26 pp 460-463, (2010).**
3. A Kaluarachchi, SV Aparna. “A classification-based tool to predict the outcome in cricket”, *In the conference, IEEE Information and Automation for Sustainability (ICIAFs),* **pp 251-253, (2010).**
4. Parag Shah, Mitesh Shah, “Predicting ODI Cricket Result”, *Journal of Tourism, Hospitality and Sports,* **pp 19-21 (2015).**
5. Viraj Phanse, Sourabh Deorah , “Evaluation and Extension to the Duckworth Lewis Method: A Dual Application of Data Mining Techniques”, *Data Mining Workshops, 2011 IEEE 11th International Conference on, Vancouver, BC, Canada.* **(ICDMW) (2011).**
6. Madan Gopal Jhawar, Vikram Pudi . “Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach”. **Report No: IIIT/TR/2016/-1. (2016).**
7. Sasank Viswanadha, Kaustubh Sivalenka, Madan Gopal Jhawar, Vikram Pudi. “Dynamic Winner Prediction in Twenty20 Cricket: Based on Relative Team Strengths”. *Journal of the Operational Research Society*, **(June 2004).**
8. Monalisha Pattnaik, Anima Bag, “Fitting of Logistic Regression Model for Prediction of Likelihood of India Winning or Losing in Cricket Match". *International Conference on Management and Information Systems,* **September 18-20, (2015).**
9. Parag Shah. “Predicting Outcome of Live Cricket Match Using Duckworth- Lewis Par Score”, *International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS)* **Volume 6, Issue 7S, (July 2017).**

B. Thesis

10. Munir, Fahad, Hasan, Md. Kamrul, Ahmed, Sakib, Md. Quraish, Sultan. *Predicting a T20 cricket match result while the match is in progress.* **Bachelor in Computer Science Thesis, BRAC University, Bangladesh (2015).**

C. Conference Proceedings

11. Gray, S. F. and Lee, T-A. L., How to fix a one-day international cricket match, *In Proceedings of AFAANZ 2003 Annual Conference*, *Proceeding's title AFAANZ 2003 Annual Conference: Program and Abstracts,* **(2003).**

D. e-websites / downloads

12. CRICKET - Elo Rating Analysis of ODI teams, https://medium.com/dataorc/cricket-elo-rating-analysis-of-odi-teams-981905cdab79

13. Python language instructions, https://www.geeksforgeeks.org/python-language-introduction/

14. What is python language how it is useful for mech engineers, https://learnmech.com/what-is-python-language-how-it-is-useful-for-mech-engineers/

15. New York University-Course-CS 100, https://www.coursehero.com/file/p79fpbu7/Variables-are-nothing-but-reserved-memory-locations-to-store-values-This-means/

16. Python data Analysis using Pandas, https://www.geeksforgeeks.org/python-data-analysis-using-pandas/

17. Python Pandas, https://www.tutorialspoint.com/python_pandas/index.htm

18. Pandas' website, https://pandas.pydata.org/

19. Flask, https://www.tutorialspoint.com/flask/index.htm

20. Official Flask Website, https://flask.palletsprojects.com/en/1.1.x/

21. Flask Wikipedia, https://en.wikipedia.org/wiki/Flask_(web_framework)

22. Quick guide on NumPy, https://www.tutorialspoint.com/numpy/numpy_quick_guide.htm

23. Official NumPy website, https://numpy.org/learn

24. Official Sklearn website, https://sklearn.org/

25. Naive Bayes Classifier, https://www.geeksforgeeks.org/naive-bayes-classifiers/