A MINI PROJECT ABSTRACT REPORT

On

PREDICTING ACCURACY OF PLAYERS IN THE CRICKET USING MACHINE LEARNING

Submitted to Jawaharlal Nehru Technological university for the partial

Fulfillment of the Requirement for the Award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

V. SOUMYA	-	20Q91A05M0
B. VAMSHI KRISHNA	-	20Q91A05M6
M. SAI KUMAR	-	20Q91A05M7
M. RANJITH	-	20Q91A05K6

Under the guidance of

Ms. B. NISHMA

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

MALLA REDDY COLLEGE OF ENGINEERING

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)

Accredited by NBA & NAAC, Recognized section 2(f) & 12(b) of UGC

New Delhi ISO9001:2015 certified Institution

Maisammaguda, Dhulapally (Post via Kompally), Secunderabad-50010

2023-2024

MALLA REDDY COLLEGE OF ENGINEERING

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)

Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC

New Delhi ISO 9001:2015 certified Institution

Maisammaguda, Dhulapally (Post via Kompally), Secunderabad – 500100

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This to certify that the Mini project report on “Predicting Accuracy of the Players in the Cricket by using Machine Learning” is successfully done by the following students of Department of Computer Science& Engineering of our college in partial fulfilment of the requirement for the award of B.Tech degree in the year 2023-2024. The results embodied in this report have not been submitted to any other University for the award of any diploma or degree.

V. SOUMYA	-	20Q91A05M0
B. VAMSHI KRISHNA	-	20Q91A05M6
M. SAI KUMAR	-	20Q91A05M7
M. RANJITH	-	20Q91A05K6

Submitted for the viva voce examination held on: ______________________

INTERNAL GUIDE	H O D	PRINCIPAL
Ms. B. NISHMA	**Dr. G. Radha Devi**	**Dr. Ashok Reddy**
Asst. Professor	**Asst. Professor**	**Professor**

DECLARATION

We, the final year students are hereby declaring the minor project report entitled "Predicting Accuracy of the Players in Cricket by using Machine Learning" has done by us under the guidance of Ms.B.NISHMA Assistant Professor, Department of CSE is submitted in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE OF ENGINEERING.**

The Results embedded in this project report have not been submitted to any other University or institute for the award of any degree or diploma.

Signature of the Candidates

V. SOUMYA	**-**	**20Q91A05M0**
B. VAMSHI KRISHNA	**-**	**20Q91A05M6**
M. SAI KUMAR	**-**	**20Q91A05M7**
M. RANJITH	**-**	**20Q91A05K6**

DATE:

PLACE: Maisammaguda

ACKNOWLEDGEMENT

First and foremost, we would like to express our immense gratitude towards our institution Malla Reddy College of Engineering, which helped us to attain profound technical skills in the field of Computer Science & Engineering, there by fulfilling our most cherished goal.

We are pleased to thank **Sri Ch. Malla Reddy**, our Founder, Chairman **MRGI**, **Sri Ch. Mahender Reddy**, Secretary, **MRGI** for providing this opportunity and support throughout the course.

It gives us immense pleasure to acknowledge the perennial inspiration of **Dr. Ashok Reddy** our beloved principal for his kind co-operation and encouragement in bringing out this task.

We would like to thank **Dr. T. V. Reddy** our vice principal, **Dr. G. Radha Devi** HOD, CSE Department for their inspiration adroit guidance and constructive criticism for successful completion of our degree.

We would like to thank **Ms. B. Nishma** Assistant Professor our internal guide, for his valuable suggestions and the guidance during the exhibition and completion of the project.

Finally, we avail this opportunity to express our deep gratitude to all staff who have contribute their valuable assistance and support making our project success.

V. SOUMYA	-	**20Q91A05M0**
B. VAMSHI KRISHNA	-	**20Q91A05M6**
M. SAI KUMAR	-	**20Q91A05M7**
M. RANJITH	-	**20Q91A05K6**

ABSTRACT

The most important and decisive task in any sport is selection of players. Evaluation of individual performance and selection of the players in cricket is most critical job. The player’s performances depend on numerous factors such as the location where the match being play, past records, his current form, average rate, strike rate, run scored at a particular venue, number of innings played against the opposition teams etc. The member of selection board, the coach and the captain of the team is conscientious for player selection. They explore special statistic, records and characteristic of the players to select the finest playing11 for each match. Throughout selection process of the players the batsman and bowlers are rated on basis of the batting and bowling average correspondingly. However in the game like cricket it is always important the situation in which it is responsibility of the batsmen to scores at most runs and bowlers to claims wickets. In this project we attempt to predict the performance of the players. For the prediction model these 2 problems are taken into account as goal as classification problem where the number of ‘runs’ and number of ‘wickets’ are classified into dissimilar range using different classifier algorithm. We exploit ‘Decision tree’, ‘Naive Bayes’, ‘Random Forest’ and ‘Multiclass SVM’ classifiers which produce the useful model to Predict for these 2 problems. Out of these 4 classifier algorithm the random forest classifier produce more accurate result then compared to other classifier and SVM produce least useful results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

LIST OF ABBREVATION

S.NO	Short Form	EXPANSION
1.	UML	Unified Modelling Language
2.	SVM	Support vector Machine
3.	SARSA	State-Action-Reward-State-Action
4.	DDPG	Deep Deterministic Policy Gradient

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION:

Cricket is game which is being played between two team of eleven players in each team where scoring the ‘runs’ or taking the ‘wickets’ is major responsibility of players. This is as a rule done by hitting the ball across the boundary or by taking run by running in between pitch. This pitch is prepared of two set of three wooden post. This pitch is called “wickets”. The “pitch” is of 20-metre (22 Yard) in span with the wickets at each side, with each side consist of two bails of 3 stumps. The cricket field is made either circular shaped or oval-shaped grassy ground. The diameter of the ground ranges between 137metre(450feet) to 100metre (500 feet). Team is made up of department of bowler, fielder, batsmen and allrounder and one wicket keeper. The wicket keeper is also a fielder who stand behind the wicket or stump and keep an eyes to focus on batsmen and ready to get catch or stumps the batsmen. Wicket keeper is fielder who is allowable to put on the gloves and external leg guards. The task of the batting department is to contribute toward the wining of match by scoring runs by striking the ball bowled at the wicket using his bat. fielding department responsible to prevent runs and tries to dismiss the batsmen by taking catch or run out and bowling department is to take maximum wickets and to put a ceiling on the other team from scoring runs at same time. Allrounder means those players who can performs in the match by batting as well as bowling and they donate towards the team by taking wickets and scoring the runs. After the completion of an inning or after dismissed of all the player of opposite team, the teams switch the roles. Each player donates enroute for the overall performance of the team by giving their best performance in each match.

1.2 OBJECTIVE:

The objectives for the described project can be summarized as follows:

1. Predictive Modelling:

- **Objective:** Develop predictive models for two key performance indicators in cricket: the number of runs scored by batsmen and the number of wickets taken by bowlers.

2. Classification Problem:

- **Objective:** Treat the prediction tasks as classification problems, where the predicted outcomes (runs or wickets) are categorized into different ranges.

3. Feature Consideration:

- **Objective:** Consider multiple factors influencing player performance, including location, past records, current form, average rate, strike rate, runs scored at specific venues, and the number of innings played against various opposition teams.

4. Selection Aid for Teams:

- **Objective:** Provide a tool that can assist selection boards, coaches, and team captains in making informed decisions about the best playing eleven for each match.

5. Algorithm Exploration:

- **Objective:** Evaluate the performance of different classification algorithms (Decision Tree, Naive Bayes, Random Forest, Multiclass SVM) for the prediction tasks.

6. Comparative Analysis:

- **Objective:** Conduct a comparative analysis of the classifiers to identify which algorithm produces the most accurate results for predicting both runs and wickets.

7. Model Usefulness:

- **Objective:** Assess the usefulness of the predictive models in the context of player selection, considering the responsibilities of batsmen to score the most runs and bowlers to claim wickets.

8. Decision Support System:

- **Objective:** Aim to create a decision support system that aids in the selection process by providing insights based on statistical analysis, records, and player characteristics.

9. Continuous Improvement:

- **Objective:** Establish a framework for continuous improvement, allowing the models to adapt to changing conditions and incorporate new data to maintain their accuracy over time.

10. Interpretability:

- **Objective:** Explore the interpretability of the models, understanding how they make predictions and which features contribute most significantly to those predictions.

11. Real-world Applicability:

- **Objective:** Ensure that the developed models are practical and can be applied in real-world scenarios, contributing to the efficiency and effectiveness of the player selection process.

By aligning our project with these objectives, we can create a structured and goal-oriented approach towards predicting player performance and enhancing the player selection process in cricket.

CHAPTER-2

LITERATURE SURVEY

2.1 LITERATURE SURVEY:

A literature survey is the most important step in the software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot for external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above considerations are taken into account for developing the proposed system.

A literature review is a body of text that aims to review the critical points of current knowledge including substantive findings as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources, and as such, do not report of an abstract accomplishment.

Most often associated with academic-oriented literature, such as a thesis, a literature review usually precedes a research proposal and results section. Its main goal is to situate the current study within the body of literature and to provide context for the particular reader.

Literature survey 1

Title: Bowler Performance Prediction for One-day International Cricket Using Neural Networks

Author: S. Muthuswamy and S. S. Lam

Description:

In one-day international cricket, bowler selection is crucial to a team's success. The performance of bowlers changes depending on the teams they face. Choosing the proper bowlers to face a certain team is still a skill. The ability to accurately estimate runs scored and wickets taken by a bowler is crucial in the bowler selection process. A collection of predictors that potentially impact bowler performance was established based on relevant literature. After that, a multiple linear regression approach was employed to find the factors that are empirically responsible for the bowlers' performance. This research aims to forecast a bowler's performance against specific opponents and venues. This research employs a machine learning approach by using the XG Boost, Random Forest Classifier, and Support Vector regression methods.

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Major work done in our project is to build prediction models that can be used to predict the performance and accuracy of any player in a given match using some supervised machine learning algorithm. In this model they rating of players has been done using different attributes of bowling and batting such as consistency, current form, form against opponent and venue the location where the match being player. This rating ranges from 1 to 5 for both batting and batting. We used weka tool for the selection of these listed attributes.

3.2 DRAWBACKS

1) Less accuracy.

2) Low Efficiency.

3.3 PROPOSED SYSTEM

We have proposed a system which overcomes the major weakness of manual work which is time consuming and required man power to manually maintain the records and statistics of each player. In this proposed system user will be provide with an interface for the coach and captain using which they will easily predict the accuracy of batsmen as well as bowler.

3.4 ADVANTAGES OF PROPOSED SYSTEM

1) High accuracy.

2) High efficiency.

3.5 SYSTEM REQUIREMENTS

3.5.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system and not how it should be implemented.

• System	:	i3 or above.
• Ram	:	4 GB.
• Hard Disk	:	40 GB.

3.5.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

• Operating system	:	Windows8 or Above.
• Coding Language	:	python.

3.6 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER-4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

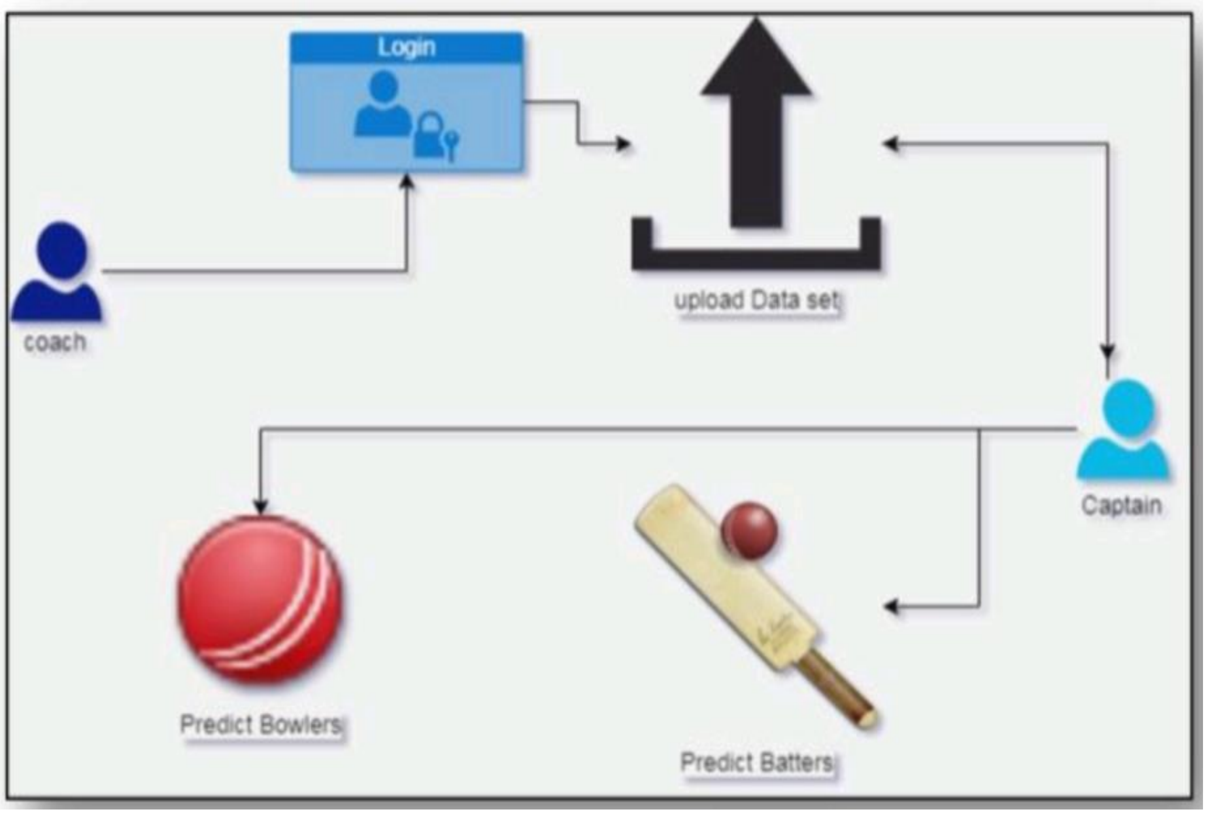


Fig 4.1.1: System Architecture Model

4.2 METHODOLOGIES

4.2.1 MODULES NAME

This project having the following 5 modules:

- TensorFlow
- Numpy
- Pandas
- Matplotlib
- Scikit-learn

4.2.2 MODULES EXPLANATION

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional

container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

4.3 UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modelling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

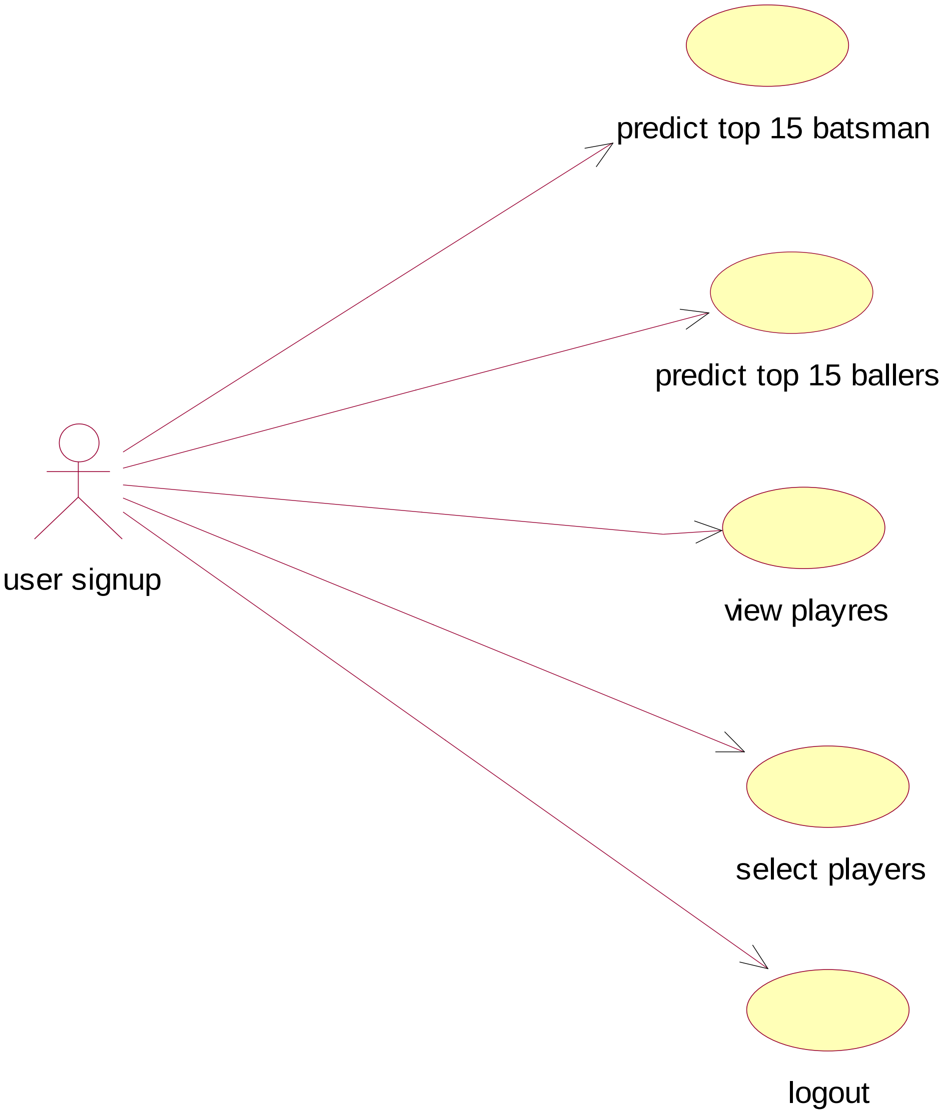


Fig 4.3.1 Use case Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

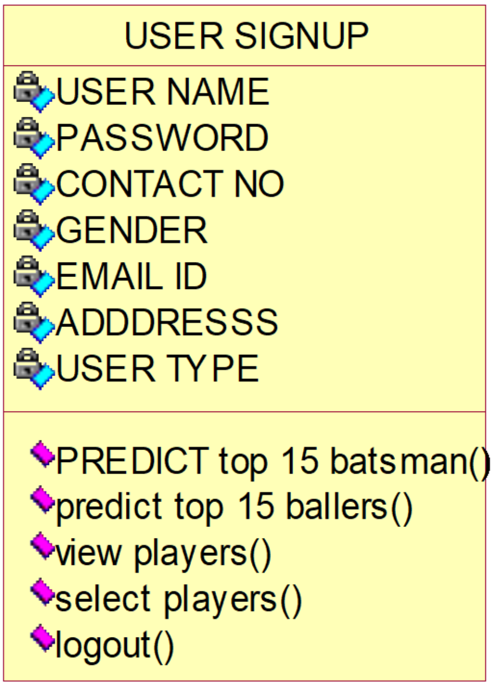


Fig 4.3.2 Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

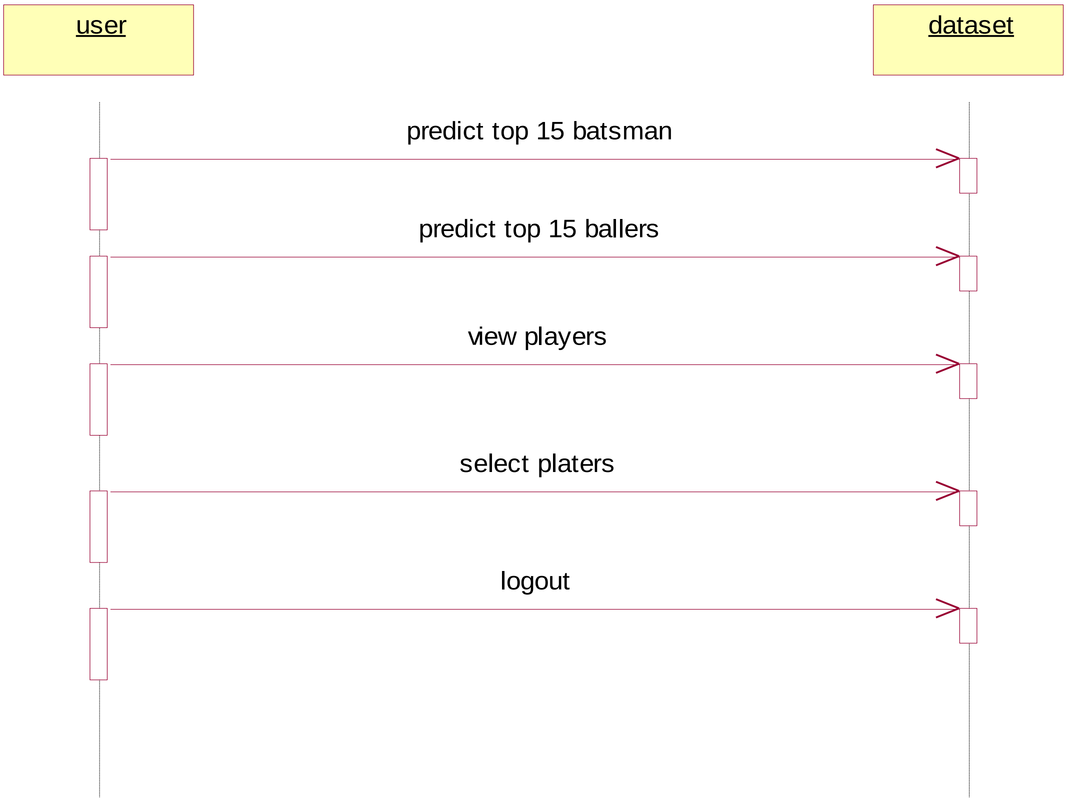


Fig 4.3.3 Sequence Diagram

COLLABORATION DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

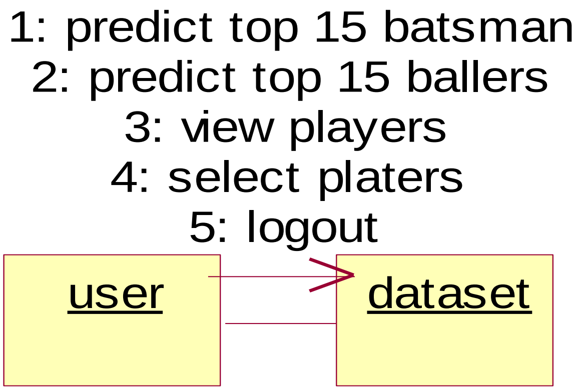


Fig 4.3.4: Collaboration Diagram

CHAPTER-5
SYSTEM IMPLEMENTATION

5.1 MACHINE LEARNING

Machine learning is a system that can learn from examples through self-improvement and without being explicitly coded by the programmer. The breakthrough comes with the idea that a machine can singularly learn from the data to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to make actionable insight. Machine learning is closely related to datamining and Bayesian predictive modelling. The machine receives data as input and uses an algorithm to formulate answers.

A typical machine learning tasks or to provide a recommendation. For those who have a Netflix account all recommend movies or series are based on the users historical data. Tech companies are using unsupervised learning to improve the users experience with personalizing recommendations.

Machine learning is also used for a variety of tasks like fraud detection, predictive maintenance, portfolio optimization, automated tasks and so on.

5.1.1 Working of Machine Learning

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feeds a previously unseen example, the machine has difficulties predicting.

The core objective of machine learning is the learning and inference. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the data. One crucial part of the data scientist is to choose carefully which data to provide to the machine.

The list of attributes used to solve a problem is called a feature vector. You can think of a feature vector as a subset of data that is used to tackle a problem. The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used to describe the data and summarize it into a model.

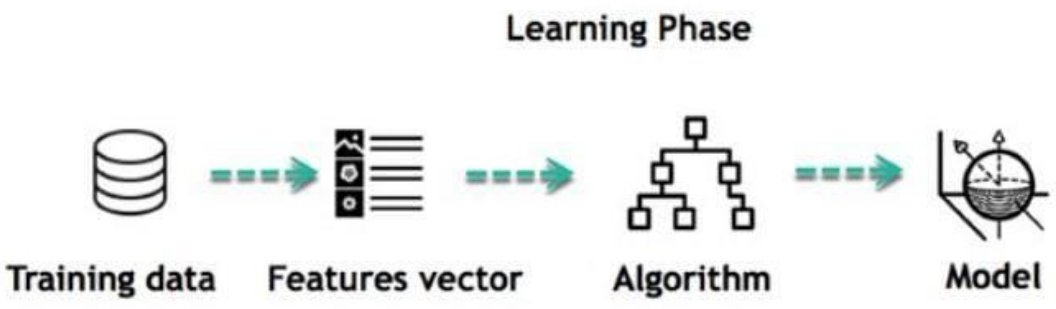


Fig 5.1.1.1: Learning phase of Machine Learning.

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant. This is the model inferring.

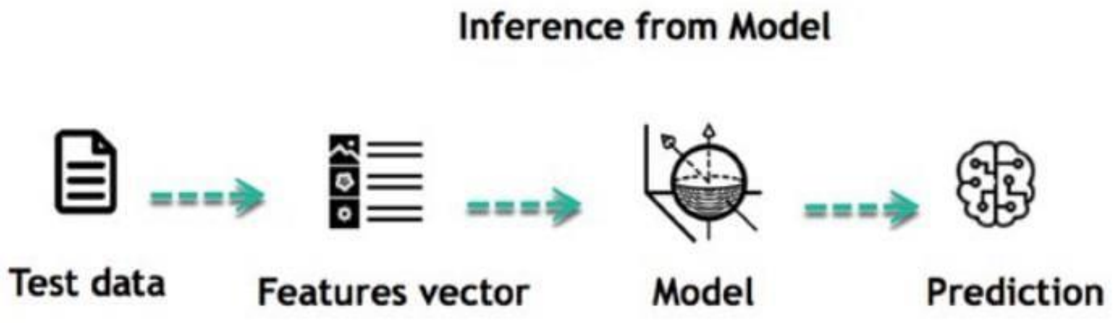


Fig 5.1.1.2: Machine Learning Inference from Model.

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

The life of Machine Learning programs is straightforward and can be summarized in the following points:

- Define a question

- Collect data
- Visualize data
- Train algorithm
- Test the Algorithm
- Collect feedback
- Refine the algorithm
- Loop 4-7 until the results are satisfying
- Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

5.1.2 Types of Machine Learning Algorithms

Machine learning can be grouped into three broad learning tasks: Supervised, Unsupervised, and Reinforcement Learning.

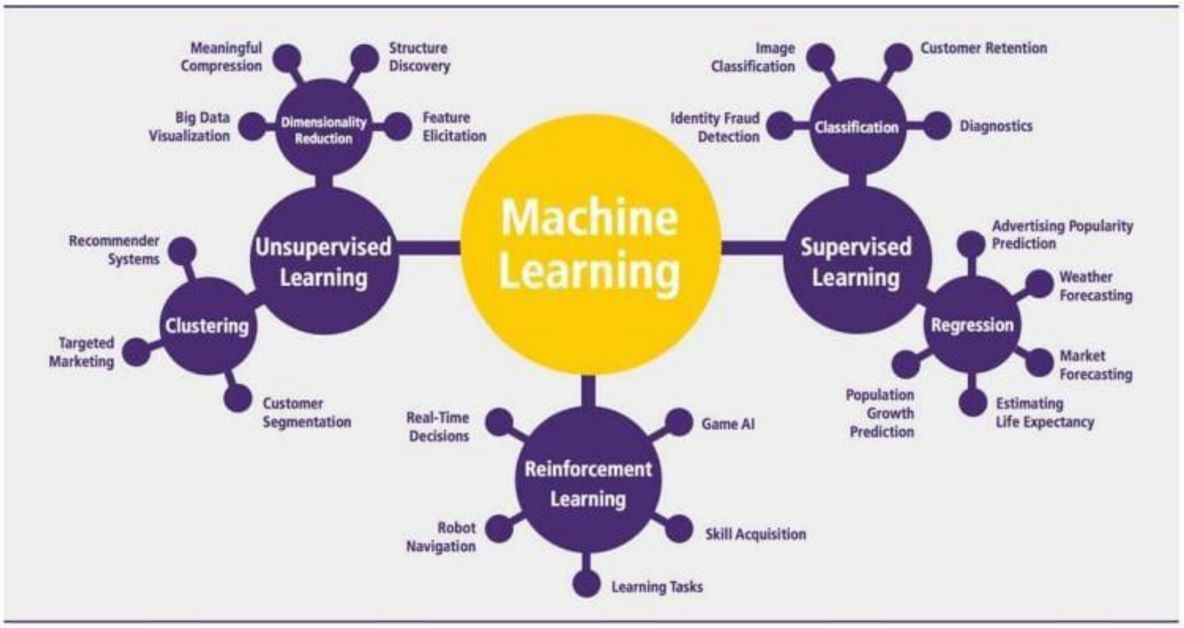


Fig 5.1.2.1: Types of Machine Learning Algorithms.

1. Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans. You can use supervised learning when the output data is known. The algorithm will predict new data.

Supervised Learning includes the following algorithms:

- Linear Regression
- Logistic Regression
- Decision Tree
- Naïve Bayes
- Support Vector Machine
- Random Forest
- AdaBoost
- Gradient Boosting Trees

There are two categories of supervised learning:

Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female. The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

2. Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns) You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you.

Unsupervised Learning includes following algorithms:

- K-means clustering
- Gaussian mixture model
- Hierarchical clustering
- Recommender system
- PCA/T-SNE

3. Reinforcement Learning

Reinforcement learning is a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games.

Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bots.

One of the most famous algorithms is:

- Q-learning
- Deep Q network
- State-Action-Reward-State-Action (SARSA)
- Deep Deterministic Policy Gradient (DDPG)

5.2 PYTHON

Python programming language is used for building the machine learning models.

5.2.1 INTRODUCTION

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

5.2.2 HISTORY

Python laid its foundation in the late 1980s.The implementation of Python was started in the December 1989 by Guido Van Rossum at CWI in Netherland. ABC programming language is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.

5.2.3 PYTHON FEATURES

1) Easy to Use

2) Interpreted Language

3) Cross-platform language

4) Free and Open Source

5) Object-Oriented language

6) Extensible

7) Large Standard Library

8) GUI Programming

9) Integrated

5.2.4 PYTHON APPLICATIONS

Python as a whole can be used in any sphere of development. Let us see what are the major regions where Python proves to be handy.

1) Console Based Application

2) Audio or Video based Applications

3) 3D CAD Applications

4) Web Applications

5) Enterprise Applications

6) Applications for Images

5.2.5 How to Install Python on Windows and Mac:

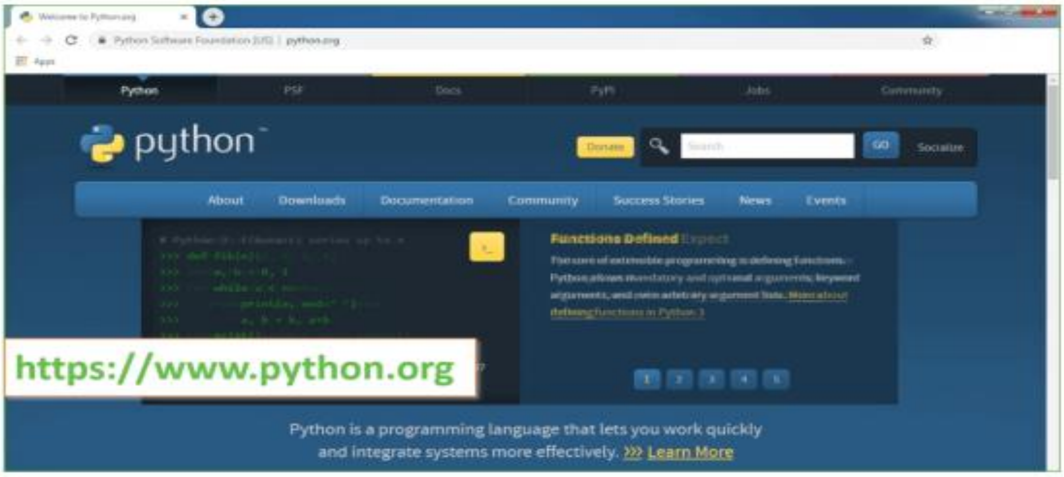
There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org



Functions Defined

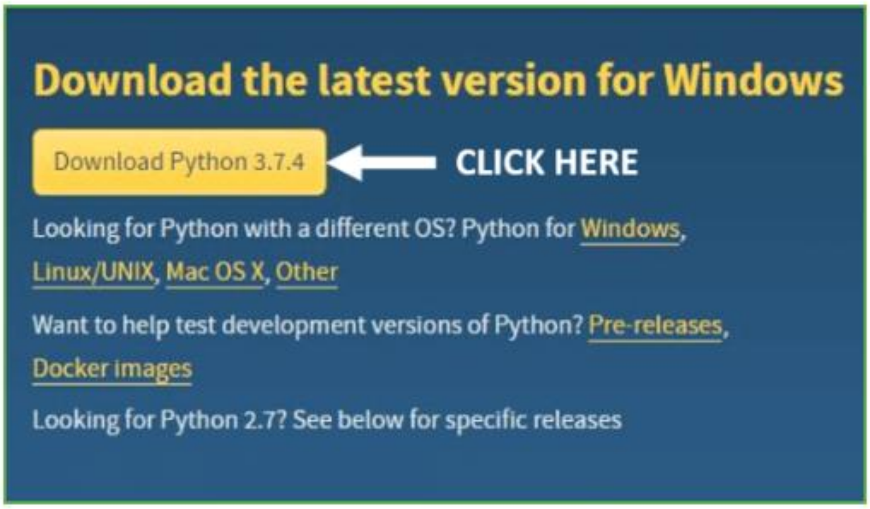
The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. More about defining functions in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Fig 5.2.5.1 Python official site page

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



Download the latest version for Windows

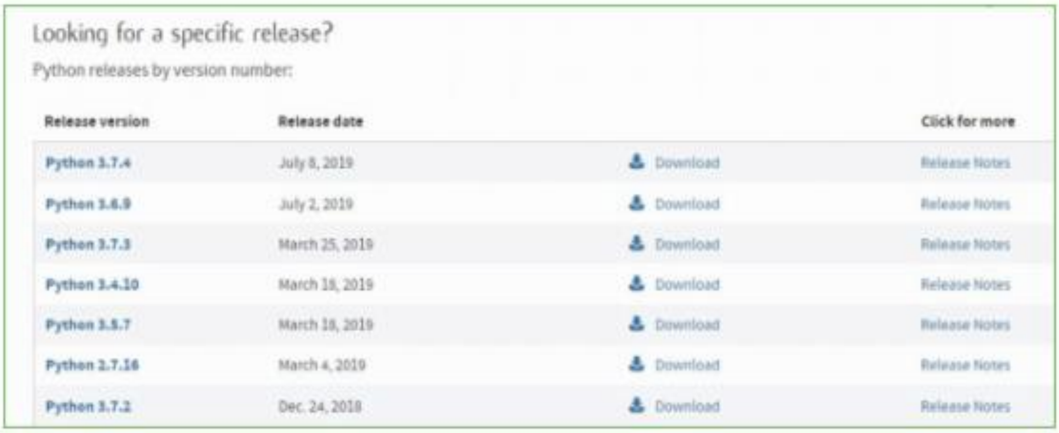
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Fig 5.2.5.2 Python Download page

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



Looking for a specific release?

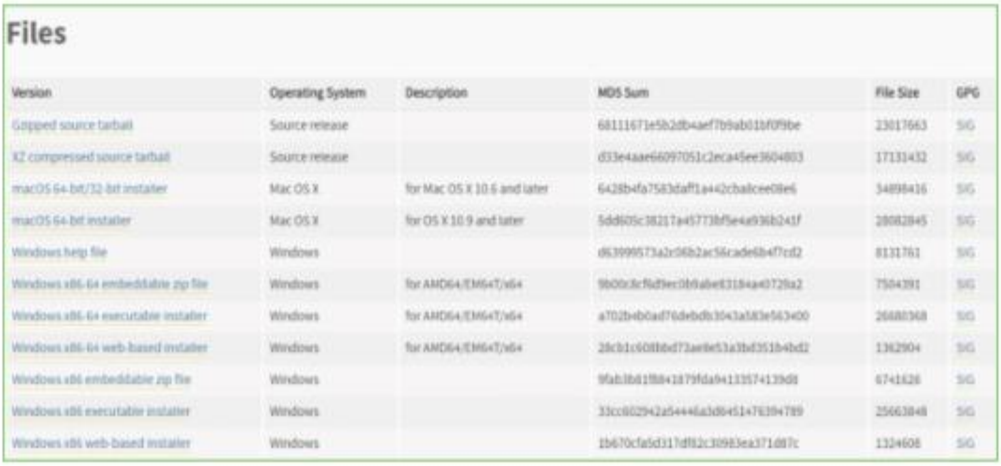
Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Fig 5.2.5.3 Python versions

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0bfabe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

Fig 5.2.5.4 Windows

To download Windows 32-bit python, you can select any one from the three options:

- Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

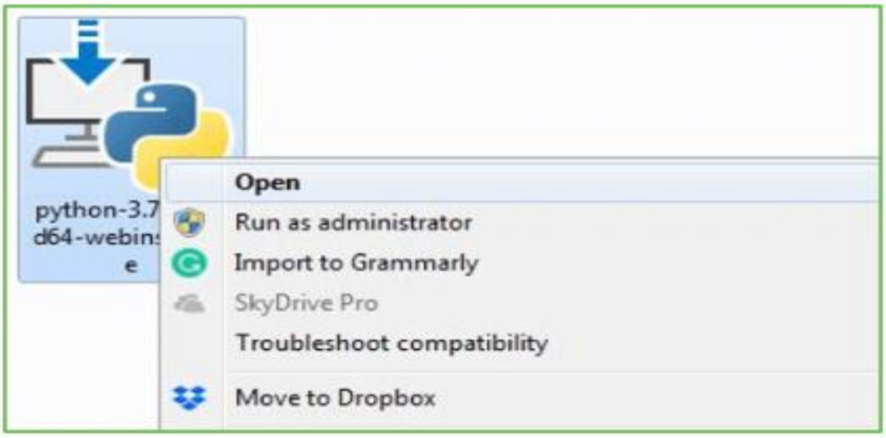
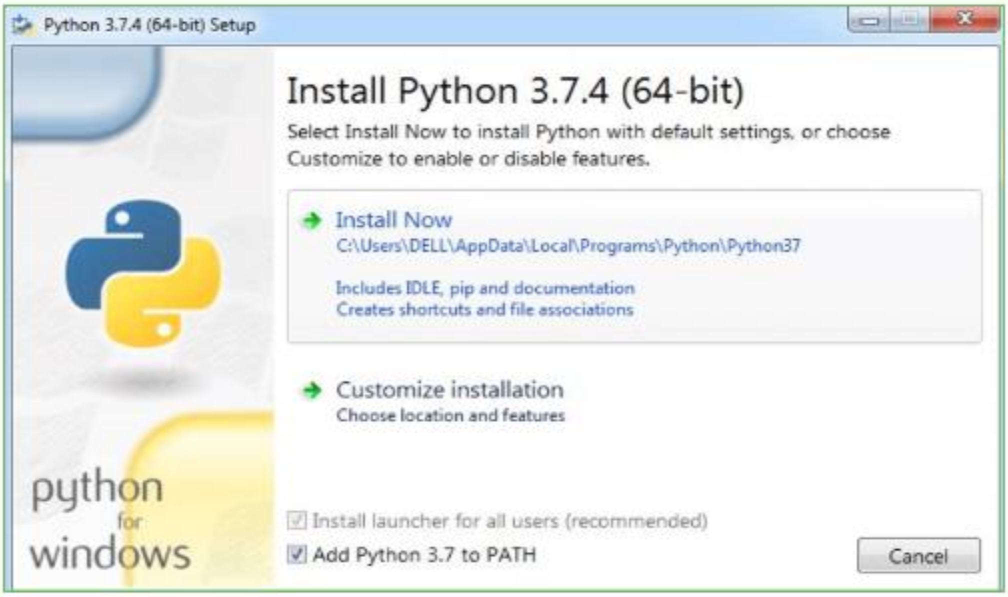


Fig 5.2.5.5 open page

Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

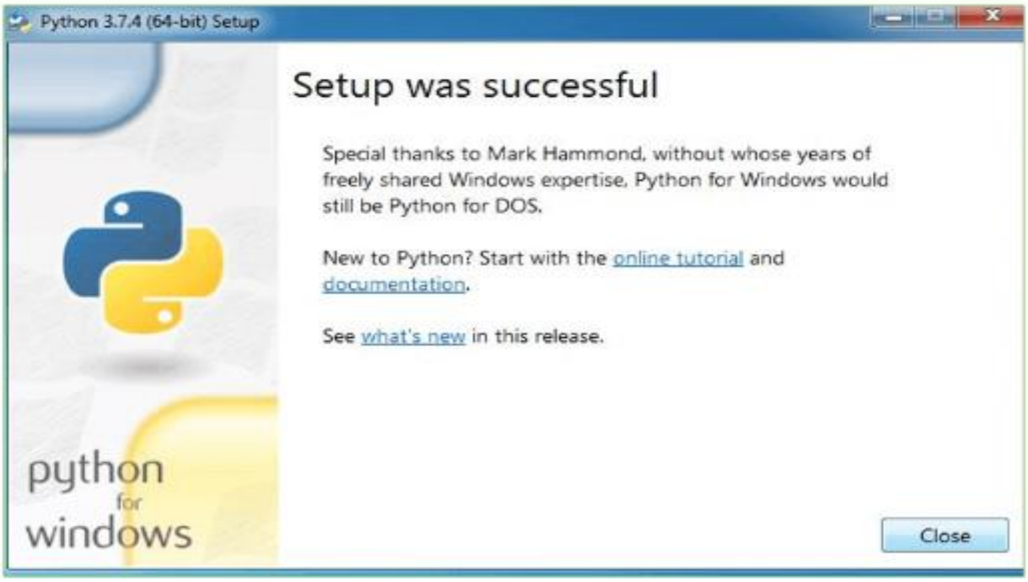
Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

Fig 5.2.5.6 Environment setup

Step 3: Click on Install NOW After the installation is successful. Click on Close.



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Fig 5.2.5.7 Setup successful

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type "cmd".

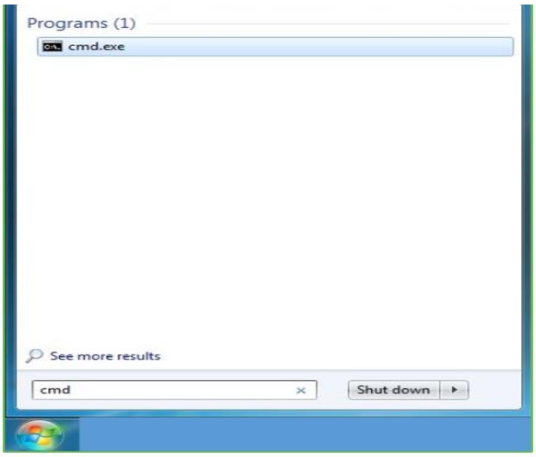
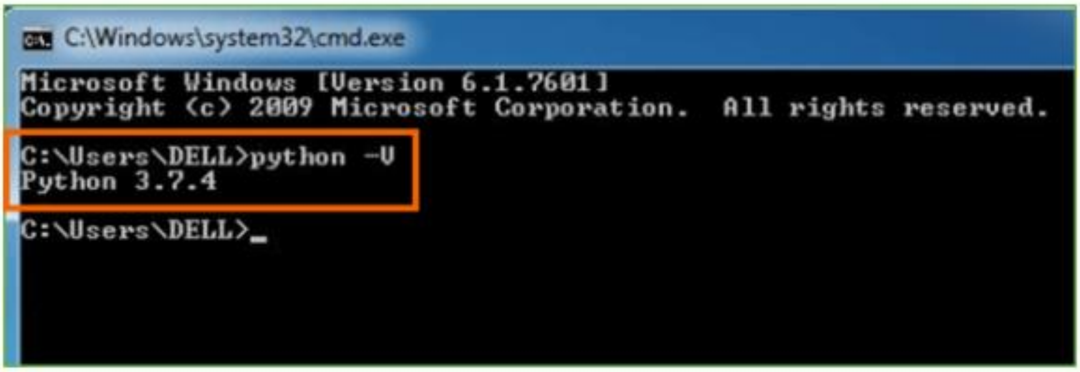


Fig 5.2.5.8 Command prompt

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

Fig 5.2.5.9 Python Installation successful

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.

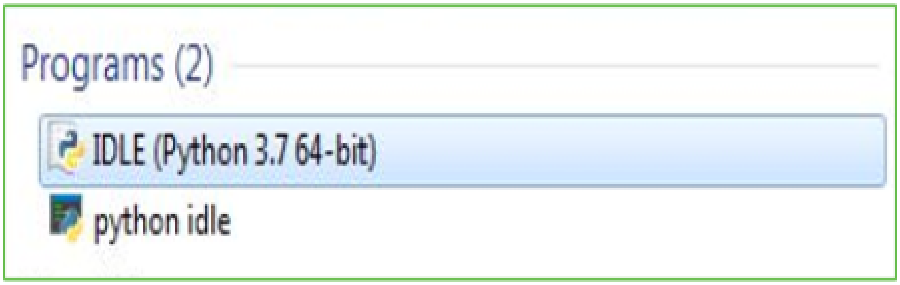
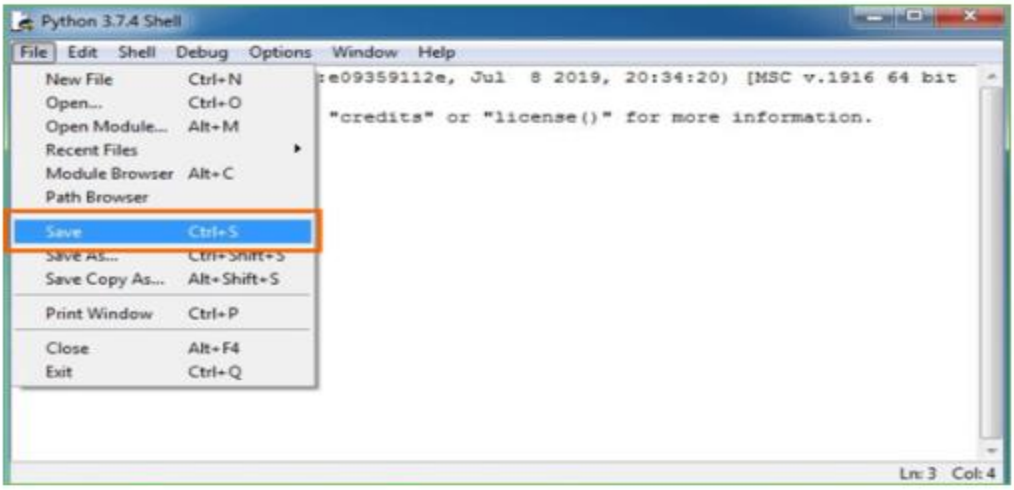


Fig 5.2.5.10. Python Idle

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on save**



```
:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit
"credits" or "license()" for more information.
```

Fig 5.2.5.11 Python files save

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g enter print

5.3 VISUAL STUDIO CODE

Visual Studio Code (VS Code) is a free and lightweight source code editor developed by Microsoft. It is widely used by developers for various programming languages and platforms. Visual Studio Code (VS Code) can be used in several ways:

1. Coding and Development: VS Code provides a powerful code editor with features like syntax highlighting, code completion, and IntelliSense, which facilitate writing ML algorithms for traffic

prediction. You can write and debug your ML code directly in VS Code, making it convenient for implementing and testing different ML models.

2. Python and ML Libraries: Python is a popular programming language for ML tasks, including traffic prediction. VS Code has excellent support for Python development, including a built-in Python interpreter and integration with popular ML libraries such as scikit-learn, TensorFlow, and PyTorch. You can leverage these libraries to build and train ML models for traffic prediction within the VS Code environment.

3. Jupyter Notebooks: VS Code supports Jupyter Notebooks, which are interactive documents that allow you to combine code, visualizations, and explanatory text. Jupyter Notebooks are commonly used in ML tasks for data exploration, model development, and result analysis. You can create and work with Jupyter Notebooks in VS Code, making it easier to iterate on and document your traffic prediction experiments.

4. Data Visualization: VS Code has extensions and integrations with data visualization libraries such as Matplotlib and Plotly, enabling you to create insightful visualizations of your traffic data. Visualizing the data can help you understand patterns, trends, and anomalies, which are crucial for developing accurate traffic prediction models.

5. Git Integration and Collaboration: Traffic prediction projects often involve collaboration and version control. VS Code's built-in Git integration allows you to manage your code repository directly within the editor. You can easily commit, push, and pull changes, collaborate with team members, and track project history, ensuring smooth collaboration and code management.

6. Terminal and Command-Line Tools: Traffic prediction projects may require running command-line tools or scripts for data preprocessing, model training, or evaluation. VS Code provides an integrated terminal, allowing you to execute command-line operations without leaving the editor. You can run scripts, manage dependencies, and interact with the commandline tools required for your traffic prediction ML workflow.

7. Extension Ecosystem: VS Code has a vast extension ecosystem, including ML-specific extensions, that can enhance your traffic prediction workflow. These extensions provide additional functionality, such as data exploration tools, model evaluation metrics, automated hyperparameter tuning, and deployment options. You can explore and install relevant ML extensions from the VS Code Marketplace to augment your traffic prediction ML capabilities.

In summary, VS Code offers a flexible and feature-rich environment for developing and implementing ML models for traffic prediction. It provides coding support, integration with ML libraries, Jupyter Notebook capabilities, data visualization tools, collaboration features, command-line access, and a wide range of extensions to enhance your traffic prediction ML workflow.

5. 4 ALGORITHM

5.4.1 Random Forest

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or the same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithms of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks. The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.

2. Build a decision tree based on these N records.

3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

4. For the classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

5.4.2 Decision Tree

A Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

5.4.3 Support Vector Machine

Support Vector Machine is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an Ndimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

5.4.4 Algorithm

1. Import the necessary libraries: `warnings`, `pandas`, `numpy`.

2. Set up warning filters to ignore warnings.

3. Import the required files.

4. Perform data preprocessing steps.

5. Perform feature extraction.

6. Create an Excel writer and write the DataFrame to a new sheet in the Excel file.

7. Close the Excel writer and save the Excel file.

8. Perform additional data preprocessing steps.

9. Perform data visualization and plot graphs.

10. Split the data into independent and dependent features.

11. Split the data into training and testing sets.

12. Implement the Random Forest classifier:

- Create an instance of the Random Forest classifier.

- Fit the model using the training set.

- Make predictions on the test data.

- Calculate the accuracy score.

-Generate a confusion matrix.

13. Implement the Decision Tree classifier:

- Create an instance of the Decision Tree classifier.

- Fit the model using the training set.

- Make predictions on the test data.

- Calculate the accuracy score.

- Generate a confusion matrix.

14. Implement the Support Vector Machine (SVM):

- Create an instance of the SVM classifier.

- Fit the model using the training set.

- Make predictions on the test data.

- Calculate the accuracy score.

- Generate a confusion matrix.

15. Compare the accuracy of the three models using a bar chart.

16. Save the trained Random Forest model using 'pickle'.

17. Load the saved model using 'pickle'.

18. The code execution is complete.

5.5 SOURCE CODE

```
from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

import pymysql

from django.http import HttpResponse

from django.core.files.storage import FileSystemStorage

import os

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from hmmlearn import hmm
import operator
import pickle
global username
def Batsman(request):
    if request.method == 'GET':
        global uname
        output = '<table border=1 align=center>'
        output+='<tr><th><font size=3 color=black>Player No</font></th>'
        output+='<th><font size=3 color=black>Player Name</font></th>'
        output+='<th><font size=3 color=black>Performance Accuracy</font></th></tr>'
       dataset = pd.read_csv("Dataset/bat.csv", usecols = ['name_x', 'runs_x', 'balls', 'strike_rate', 'fours', 'sixes', 'how_out', 'run_rate'])
        le1 = LabelEncoder()
        le2 = LabelEncoder()
        scaler = StandardScaler()
        dataset['name_x'] = pd.Series(le1.fit_transform(dataset['name_x'].astype(str)))
        dataset['how_out'] = pd.Series(le2.fit_transform(dataset['how_out'].astype(str)))
        dataset = dataset.values
        X = dataset[:,0:dataset.shape[1]-1]
        Y = dataset[:,dataset.shape[1]-1]
        X = scaler.fit_transform(X)
        print(X.shape)
        print(Y.shape)
```

```
if os.path.exists("model/bat"):
    with open('model/bat', 'rb') as file:
        model = pickle.load(file)
    file.close()
else:
    model = hmm.GaussianHMM(10, "full", n_iter=500)
    model.fit(X)
    with open('model/bat', 'wb') as file:
        pickle.dump(model, file)
    file.close()
    testData = pd.read_csv("Dataset/test_bat.csv", usecols = ['name_x', 'runs_x', 'balls', 
'strike_rate', 'fours', 'sixes', 'how_out'])
player = testData['name_x']
testData['name_x'] = pd.Series(le1.transform(testData['name_x'].astype(str)))
testData['how_out'] = pd.Series(le2.transform(testData['how_out'].astype(str)))
testData = testData.values
X = scaler.transform(testData)
performance = []
predict = model.predict(X)
selected = []
for i in range(len(predict)):
    if player[i] not in selected:
        selected.append(player[i])
        performance.append([player[i], predict[i]])
performance.sort(key = operator.itemgetter(1), reverse = True)
```

```
    for i in range(0,15):
        player_name = performance[i][0]
        accuracy = performance[i][1]
        output+='<tr><td><font size=3 color=black>'+str(i+1)+'</font></td>'
        output+='<td><font size=3 color=black>'+str(player_name)+'</font></td>'
        output+='<td><font size=3 color=black>'+str(accuracy)+'</font></td></tr>'
        output+="</table><br/><br/><br/><br/><br/><br/>"
    context= {'data':output}
    return render(request, 'ViewPrediction.html', context)
 def index(request):
  if request.method == 'GET':
    return render(request, 'index.html', {})
def UserLogin(request):
  if request.method == 'GET':
    return render(request, 'UserLogin.html', {})
  def Register(request):
  if request.method == 'GET':
    return render(request, 'Register.html', {})
  def Signup(request):
  if request.method == 'POST':
    username = request.POST.get('username', False)
    password = request.POST.get('password', False)
    contact = request.POST.get('contact', False)
    gender = request.POST.get('gender', False)
    email = request.POST.get('email', False)

    address = request.POST.get('address', False)

    usertype = request.POST.get('usertype', False)

    output = "Error in signup process"

    con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = '', database =
'cricket',charset='utf8')

    with con:

        cur = con.cursor()

        cur.execute("select username FROM signup")

        rows = cur.fetchall()

        for row in rows:

            if row[0] == username:

                output = username+" Username already exists"

                break

    if output == 'Error in signup process':

        db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password =
'', database = 'cricket',charset='utf8')

        db_cursor = db_connection.cursor()

student_sql_query="INSERT INTO
signup(username,password,contact_no,gender,email,address,usertype)
VALUES('"+username+"','"+password+"','"+contact+"','"+gender+"','"+email+"','"+address+"','"
+usertype+"')"

        db_cursor.execute(student_sql_query)

        db_connection.commit()

        print(db_cursor.rowcount, "Record Inserted")

        if db_cursor.rowcount == 1:
            output = 'Signup Process Completed'
    context= {'data':output}
    return render(request, 'Register.html', context)


def UserLoginAction(request):
    if request.method == 'POST':
    global uname
    username = request.POST.get('username', False)
    password = request.POST.get('password', False)
    user_type = ""
    index = 0
    con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = ", database = 'cricket',charset='utf8')
    with con:
        cur = con.cursor()
        cur.execute("select username,password,usertype FROM signup")
        rows = cur.fetchall()
        for row in rows:
            if row[0] == username and password == row[1]:
                user_type = row[2]
                uname = username
                index = 1
                break
    if index == 1:
        context= {'data':"Welcome "+uname+"<br/>You Logged in as "+user_type}
        return render(request, 'UserScreen.html', context)
    else:
        context= {'data':'login failed'}
        return render(request, 'UserLogin.html', context)


def Ballers(request):
    if request.method == 'GET':
    global uname
    output = '<table border=1 align=center>'
    output+='<tr><th><font size=3 color=black>Player No</font></th>'
    output+='<th><font size=3 color=black>Player Name</font></th>'
    output+='<th><font size=3 color=black>Performance Accuracy</font></th></tr>'
    #reading dataset
     dataset = pd.read_csv("Dataset/ball.csv", usecols = ['name_x', 'run_conceded', 'maidens',
'wickets', 'overs', 'economy', 'wides', 'no_balls', 'fours',
                                          'sixes', 'zeros', 'runs', 'over', 'run_rate'])
    dataset.fillna(0, inplace = True)
    le1 = LabelEncoder()
    scaler = StandardScaler()
    dataset['name_x'] = pd.Series(le1.fit_transform(dataset['name_x'].astype(str)))
    X = dataset.values
    X = scaler.fit_transform(X)
    if os.path.exists("model/ball"):
        with open('model/ball', 'rb') as file:
            model = pickle.load(file)
    file.close()
  else:
    #now training HMM model on training dataset
    model = hmm.GaussianHMM(10, "full", n_iter=5000)
    model.fit(X)
    with open('model/ball', 'wb') as file:
      pickle.dump(model, file)
    file.close()
    testData = pd.read_csv("Dataset/test_ball.csv", usecols = ['name_x', 'run_conceded',
'maidens', 'wickets', 'overs', 'economy', 'wides',
                                    'no_balls', 'fours', 'sixes', 'zeros', 'runs', 'over', 'run_rate'])
  testData.fillna(0, inplace = True)
  player = testData['name_x']
  testData['name_x'] = pd.Series(le1.transform(testData['name_x'].astype(str)))
  testData = testData.values
  X = scaler.transform(testData)
  performance = []
  predict = model.predict(X)
  selected = []
  for i in range(len(predict)):
    if player[i] not in selected:
      selected.append(player[i])
      performance.append([player[i], predict[i]])
  performance.sort(key = operator.itemgetter(1), reverse = True)
  for i in range(0,15):
    player_name = performance[i][0]
    accuracy = performance[i][1]
    output+='<tr><td><font size=3 color=black>'+str(i+1)+'</font></td>'
    output+='<td><font size=3 color=black>'+str(player_name)+'</font></td>'
    output+='<td><font size=3 color=black>'+str(accuracy)+'</font></td></tr>'
output+="</table><br/><br/><br/><br/><br/><br/>"
context= {'data':output}
return render(request, 'ViewPrediction.html', context)
```

CHAPTER-6

TESTING

6.1 Testing

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

DEVELOPING METHODOLOGIES

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

6.2 Types of Tests

6.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

6.2.3 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2.4 Performance Test

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

6.2.5 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.
The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

6.2.6 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance Testing for Data Synchronization:

- ➢ The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node

- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

6.2.7 Build the test plan

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

6.3 TEST CASES

Test cases can be divided in to two types. First one is Positive test cases and second one is negative test cases. In positive test cases are conducted by the developer intention is to get the output. In negative test cases are conducted by the developer intention is to don't get the output.

TEST PLAN

The test procedure is started by building up a thorough arrangement to test the general usefulness and extraordinary highlights on an assortment of stage mixes. Exacting quality control methods are utilized. The procedure checks that the application meets the necessities indicated in the framework prerequisites report and is sans bug.

Any project can be separated into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing serves to character the potential bugs in the individual segment, so the segment that has bugs can be recognized and can be redressed from mistakes.

Test Case Id	Test Case Name	Test Case Description	Test Steps			Test Case Status	Test PrioritY
			Step	Expected	Actual		
01	Start the Application	Host the application and test if it starts making sure the required software is available	If it doesn't Start	We cannot run the application.	The application hosts success.	High	High
02	Home Page	Check the deployment environment for properly loading the application.	If it doesn’t load.	We cannot access the application.	The application is running successfully.	High	High
03	User	Verify the	If it	We	The	High	Hi

	Mode	working of the application in freestyle mode	doesn't Respond	cannot use the Freestyle mode.	application displays the Freestyle Page		gh
04	Data Input	Verify if the application takes input and updates	If it fails to take the input or store in The Database	We cannot proceed further	The application updates the input to application	High	High

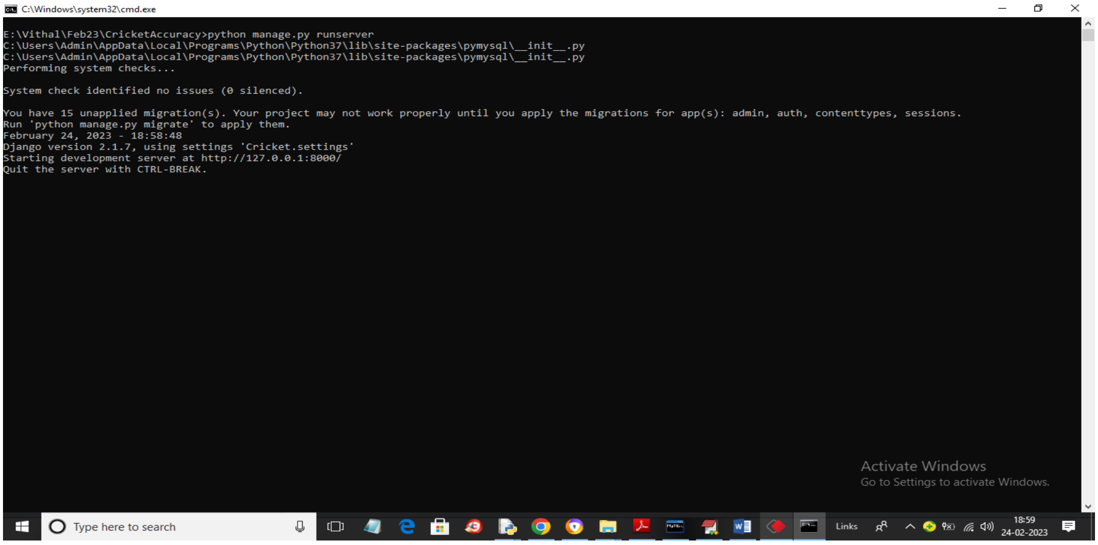
Table 6.3.1 Test case

CHAPTER-7

RESULTS

7.1 SCREENSHOTS

To run project first create database in MYSQL by copying content from DB.txt file and paste in MYSQL and then double click on ‘run.bat’ file to start python server and get below output .



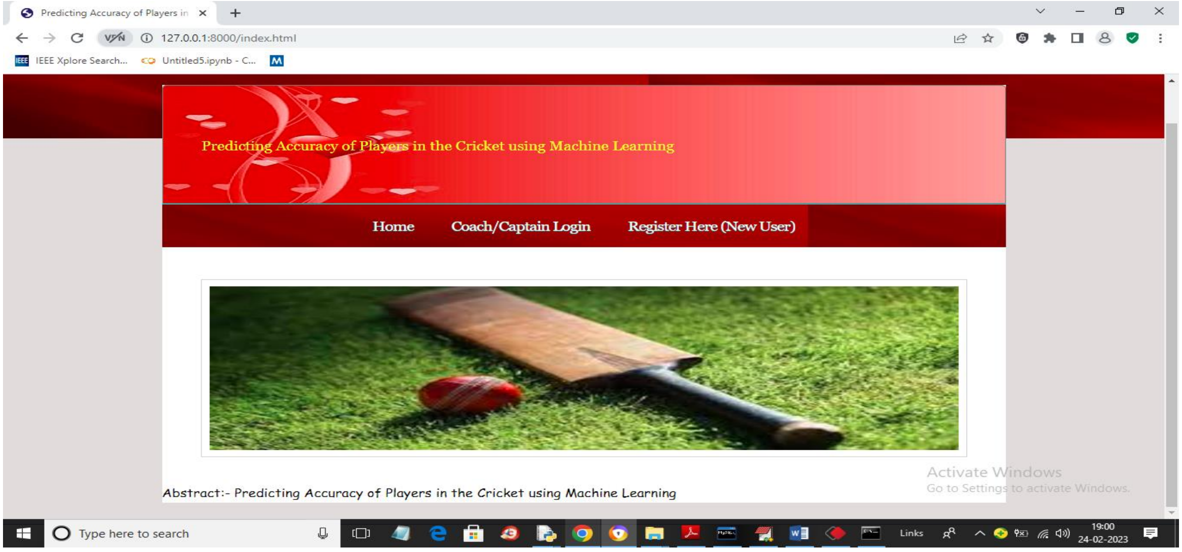
```
E:\Vithal\Feb23\CricketAccuracy>python manage.py runserver
C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\pymysql\__init__.py
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 24, 2023 - 18:58:48
Django version 2.1.7, using settings 'Cricket.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Fig 7.1.1 Python server

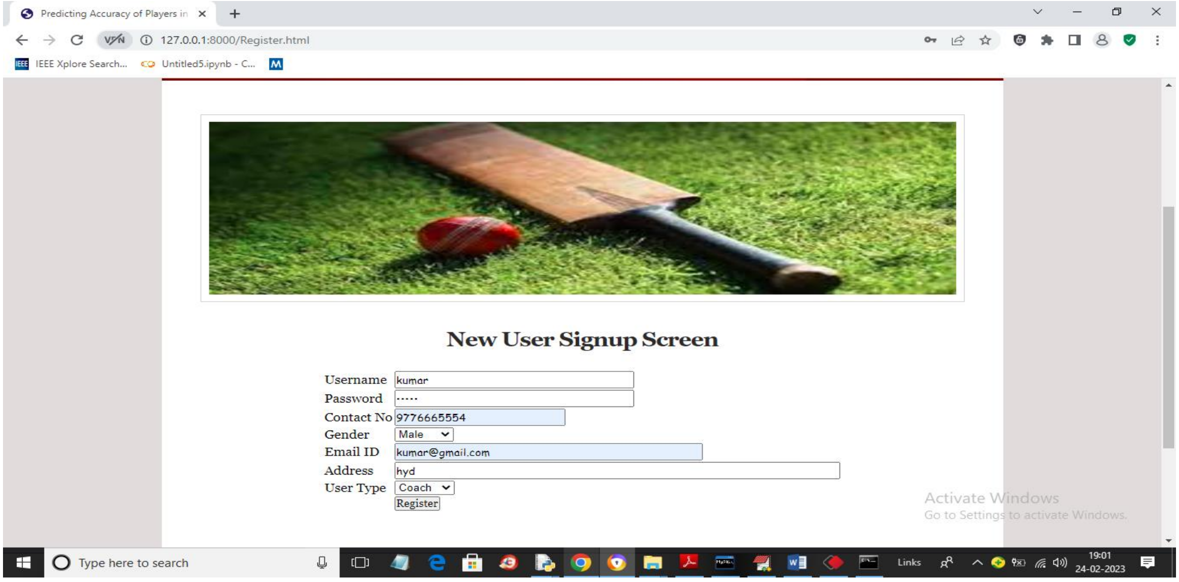
In above screen python server started and now open browser and enter URL as ‘http://127.0.0.1:8000/index.html’ and press enter key to get below page



Predicting Accuracy of Players in
127.0.0.1:8000/index.html
IEEE Xplore Search...
Untitled5.ipynb - C...
Predicting Accuracy of Players in the Cricket using Machine Learning
Home
Coach/Captain Login
Register Here (New User)
Abstract:- Predicting Accuracy of Players in the Cricket using Machine Learning
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
19:00
24-02-2023

Fig 7.1.2 Home page

In above screen click on ‘Register Here’ link to signup user

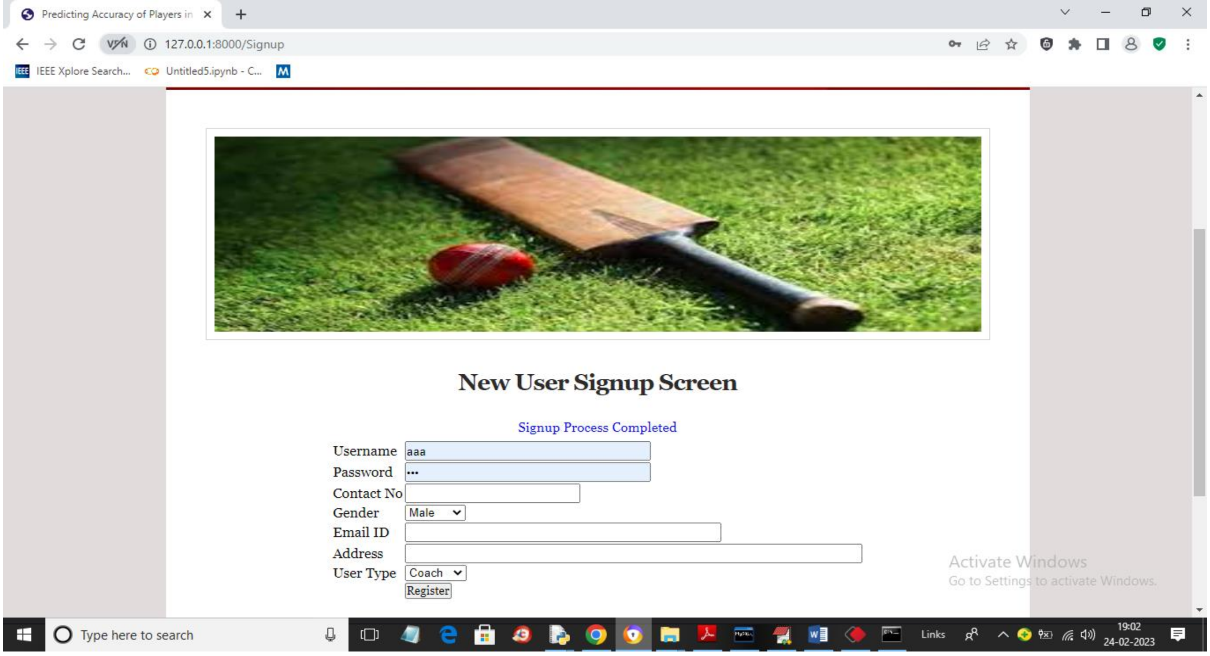


New User Signup Screen

Username kumar
Password •••••
Contact No 9776665554
Gender Male
Email ID kumar@gmail.com
Address hyd
User Type Coach
Register

Fig 7.1.3 Register page

In above screen user is signing up and after pressing button will get below output.



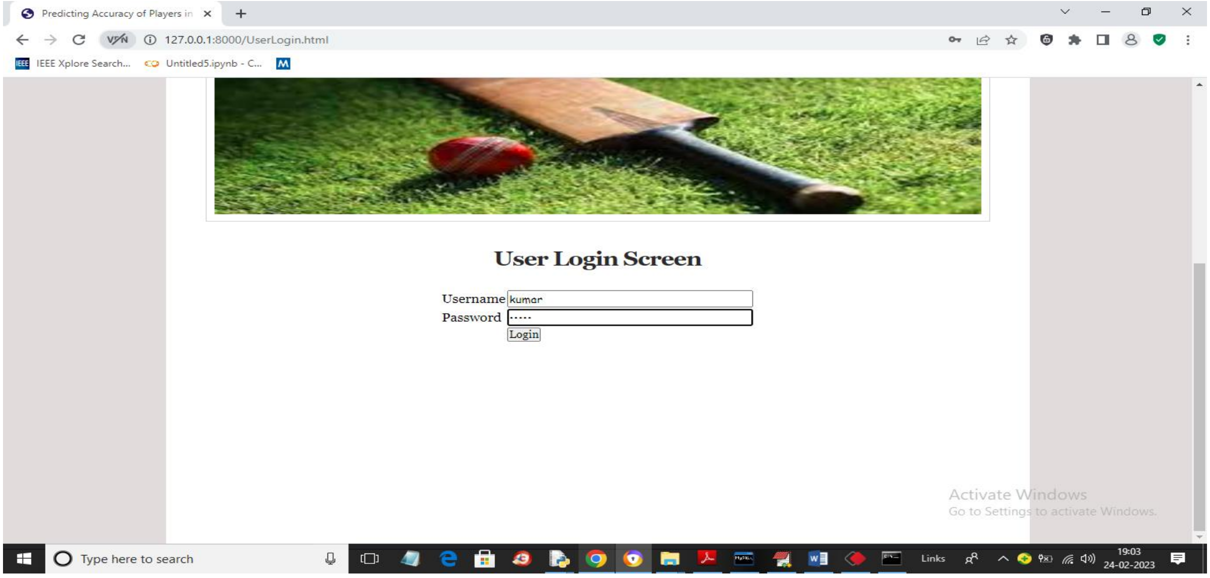
New User Signup Screen

Signup Process Completed

Username aaa
Password •••
Contact No
Gender Male
Email ID
Address
User Type Coach
Register

Fig 7.1.4 Signup completed

In above screen we can see ‘Signup Process Completed’ and now can click on ‘Coach/Captain Login’ link to get below login screen

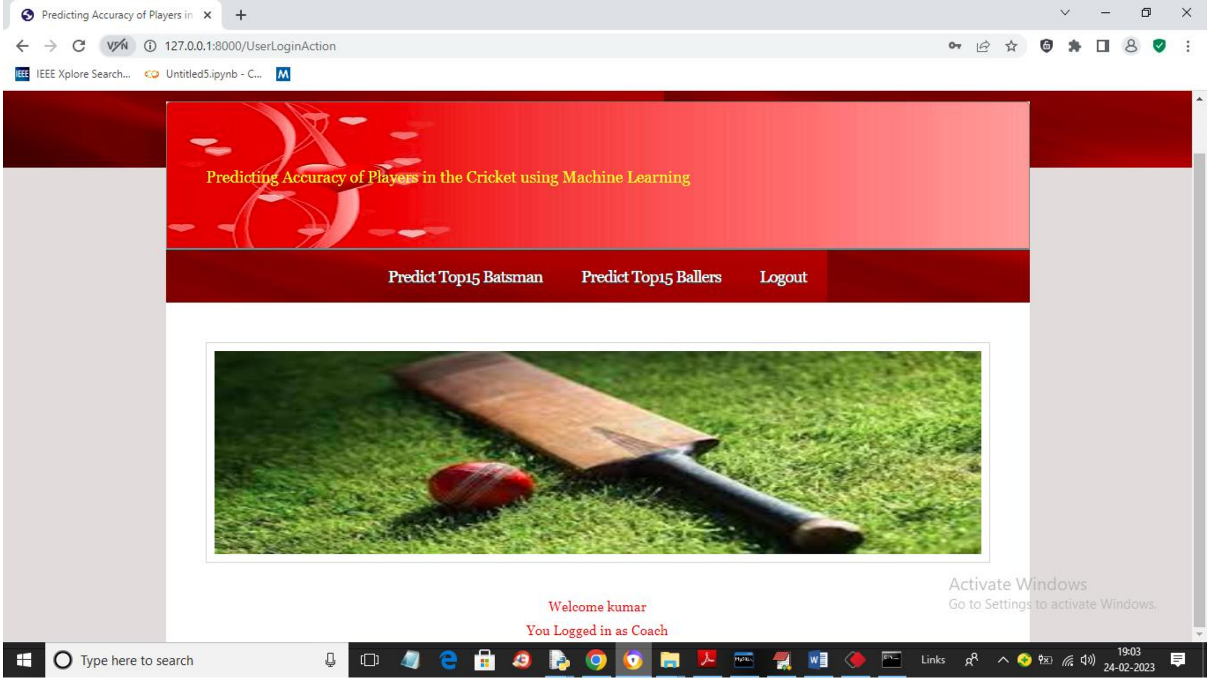


User Login Screen

Username kumar
Password •••••
Login

Fig 7.1.5 Login page

In above screen user is login and after login will get below page

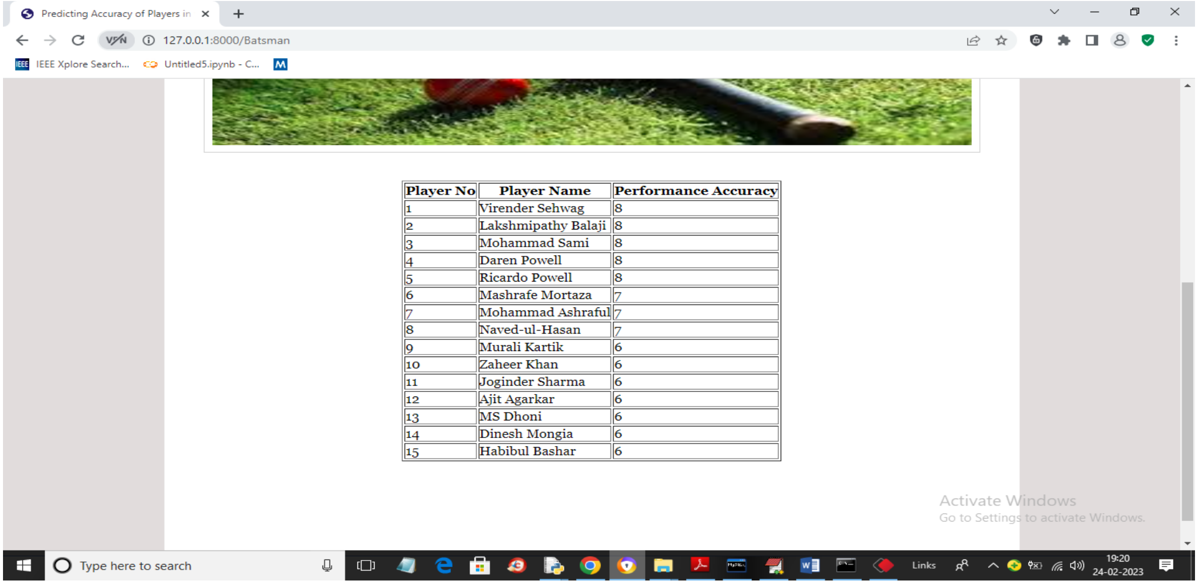

Predicting Accuracy of Players in
127.0.0.1:8000/UserLoginAction
IEEE Xplore Search...
Untitled5.ipynb - C...
Predicting Accuracy of Players in the Cricket using Machine Learning
Predict Top15 Batsman
Predict Top15 Ballers
Logout
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
19:03
24-02-2023

Welcome kumar

You Logged in as Coach

Fig 7.1.6 Welcome page

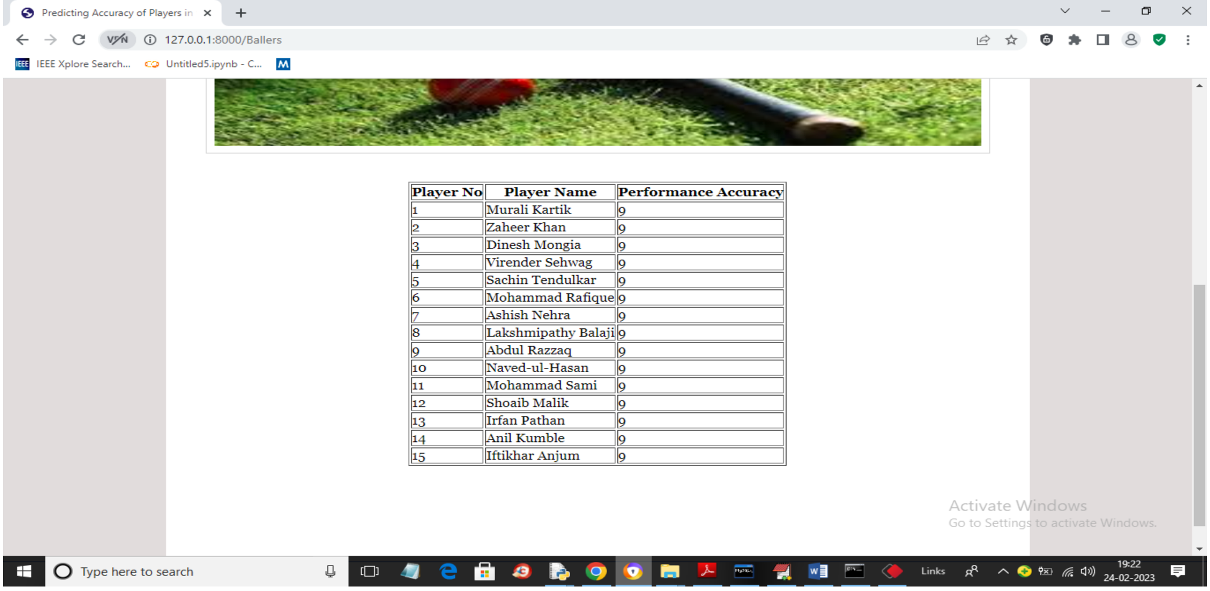
In above screen we can see user is login as ‘Coach’ and now click on ‘Predict Top15 Batsman’ link to get below output



Player No	Player Name	Performance Accuracy
1	Virender Sehwag	8
2	Lakshmipathy Balaji	8
3	Mohammad Sami	8
4	Daren Powell	8
5	Ricardo Powell	8
6	Mashrafe Mortaza	7
7	Mohammad Ashraful	7
8	Naved-ul-Hasan	7
9	Murali Kartik	6
10	Zaheer Khan	6
11	Joginder Sharma	6
12	Ajit Agarkar	6
13	MS Dhoni	6
14	Dinesh Mongia	6
15	Habibul Bashar	6

Fig 7.1.7 Batsman Prediction

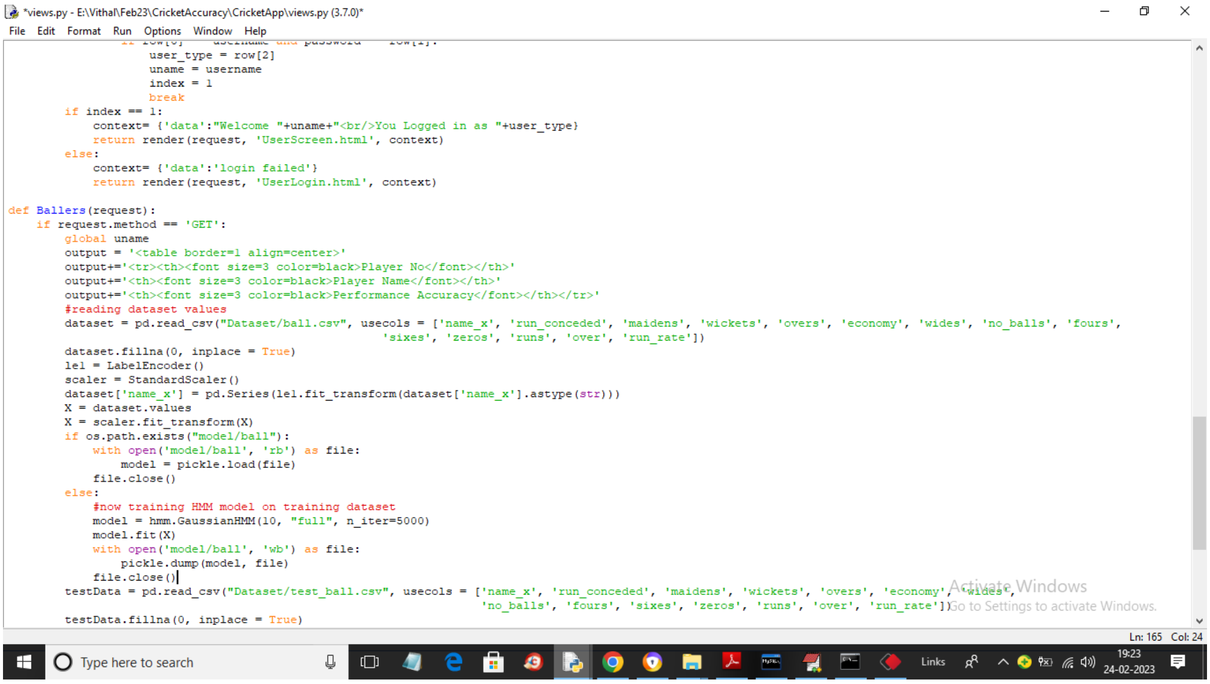
In above screen we can see the top performing players based on past performance and our dataset contains players from worldwide so will get other country player also. Similarly you can click on ‘Top15 Ballers’ link to get below output.



Player No	Player Name	Performance Accuracy
1	Murali Kartik	9
2	Zaheer Khan	9
3	Dinesh Mongia	9
4	Virender Sehwag	9
5	Sachin Tendulkar	9
6	Mohammad Rafique	9
7	Ashish Nehra	9
8	Lakshmipathy Balaji	9
9	Abdul Razzaq	9
10	Naved-ul-Hasan	9
11	Mohammad Sami	9
12	Shoaib Malik	9
13	Irfan Pathan	9
14	Anil Kumble	9
15	Iftikhar Anjum	9

Fig 7.1.8 Bowlers Prediction

In above screen we can see list of top15 ballers and similarly you can click on any link to get prediction. In below screen you can see code for HMM training to rank each player between 1 to 10 scale based on their performance


views.py - E:\Vithal\Feb23\CricketAccuracy\CricketApp\views.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 165 Col: 24
Type here to search
Links
19:23
24-02-2023
Activate Windows
Go to Settings to activate Windows.

```
                if row[0] == username and password == row[1]:
                    user_type = row[2]
                    uname = username
                    index = 1
                    break
        if index == 1:
            context= {'data':"Welcome "+uname+"<br/>You Logged in as "+user_type}
            return render(request, 'UserScreen.html', context)
        else:
            context= {'data':'login failed'}
            return render(request, 'UserLogin.html', context)

def Ballers(request):
    if request.method == 'GET':
        global uname
        output = '<table border=1 align=center>'
        output+='<tr><th><font size=3 color=black>Player No</font></th>'
        output+='<th><font size=3 color=black>Player Name</font></th>'
        output+='<th><font size=3 color=black>Performance Accuracy</font></th></tr>'
        #reading dataset values
        dataset = pd.read_csv("Dataset/ball.csv", usecols = ['name_x', 'run_conceded', 'maidens', 'wickets', 'overs', 'economy', 'wides', 'no_balls', 'fours',
                                                     'sixes', 'zeros', 'runs', 'over', 'run_rate'])
        dataset.fillna(0, inplace = True)
        lel = LabelEncoder()
        scaler = StandardScaler()
        dataset['name_x'] = pd.Series(lel.fit_transform(dataset['name_x'].astype(str)))
        X = dataset.values
        X = scaler.fit_transform(X)
        if os.path.exists("model/ball"):
            with open('model/ball', 'rb') as file:
                model = pickle.load(file)
            file.close()
        else:
            #now training HMM model on training dataset
            model = hmm.GaussianHMM(10, "full", n_iter=5000)
            model.fit(X)
            with open('model/ball', 'wb') as file:
                pickle.dump(model, file)
            file.close()
        testData = pd.read_csv("Dataset/test_ball.csv", usecols = ['name_x', 'run_conceded', 'maidens', 'wickets', 'overs', 'economy', 'wides',
                                                                   'no_balls', 'fours', 'sixes', 'zeros', 'runs', 'over', 'run_rate'])

        testData.fillna(0, inplace = True)
```

Fig 7.1.9 HMM training

In above screen read red colour comments to know about HMM training.

CHAPTER-8

CONCLUSION

8.1 CONCLUSION

Players selection posses a vital role in the team's triumph. The selection committee board member, coach and captain of team is responsible for selection of the best players for team for each match. The player's performances depend on various factors such as the location where the match being play, past records, his current form, average rate, strike rate, run scored at a particular venue, number of innings played against the opposition teams etc. Taking into consideration these information they employ an accurate prediction model which predict the accuracy of the batsmen and bowlers. In this project we modelled datasets based on players earlier record. Decision Tree, Naïve Bayes, Random Forest and support Vector Machine supervising machine learning algorithm were evaluated and used. Random forest algorithm found to be produce more accurate and useful outcome among the other classifier algorithm. Whereas the SVM produce unexpected and less useful result.

CHAPTER-9

FUTURE ENHANCEMENTS

9.1 FUTURE ENHANCEMENTS

The project you described focuses on predicting the performance of cricket players using classification algorithms. Future enhancements for this project could include the following:

Feature Engineering:

Explore additional features that might contribute to better predictions. This could include player fitness levels, recent injuries, player roles (e.g., opener, middle-order batsman), and performance in different formats of the game (Test, One Day International, Twenty20).

Data Augmentation:

Increase the volume of the dataset by incorporating more historical data. This could involve collecting statistics from a larger number of matches, seasons, or tournaments.

Fine-Tuning Hyperparameters:

Experiment with different hyperparameters for the classification algorithms. This could involve tuning the parameters of the decision tree, random forest, and SVM classifiers to achieve better results. Techniques like grid search or randomized search can be employed for this purpose.

Ensemble Methods:

Explore ensemble methods that combine predictions from multiple classifiers. Techniques like stacking, where the outputs of multiple models are used as inputs for a meta-model, could potentially improve predictive accuracy.

Dynamic Model Updating:

Develop a system that allows for dynamic updating of the model based on real-time player performance. This would require continuous data integration and retraining of the model to adapt to changing player form and conditions.

Incorporate Player and Team Dynamics:

Consider incorporating team dynamics and player interactions into the model. Team chemistry, player partnerships, and opposition team strengths could influence individual player performance.

Explainability and Interpretability:

Enhance the interpretability of the model's predictions. This could involve using techniques to explain why a particular prediction was made, which is especially important in decision-making scenarios.

User Interface for Decision Makers:

Develop a user-friendly interface for the selection board, coach, and captain to interact with the model. This could include a dashboard displaying predicted player performances, relevant statistics, and other insights.

Cross-Validation and Robustness Testing:

Implement rigorous cross-validation techniques to ensure the model's robustness. This helps in assessing how well the model generalizes to new and unseen data.

Collaboration with Cricket Experts:

Collaborate with cricket experts to refine the model and incorporate domain-specific knowledge. Experts can provide valuable insights that might not be captured by statistical models alone.

Remember that the success of the model depends not only on the algorithms used but also on the quality and relevance of the data and the features selected. Continuous refinement and iteration based on feedback and new data will contribute to the ongoing success of the predictive model.

CHAPTER-10

REFERENCES

10.1 REFERENCES

[1] S. Muthuswamy and S. S. Lam, "Bowler Performance Prediction for One-day International Cricket Using Neural Networks," in Industrial Engineering Research Conference, 2008.

[2] G. D. I. Barr and B. S. Kantor, "A Criterion for Comparing and Selecting Batsmen in Limited Overs Cricket," Operational Research Society, vol. 55, no. 12, pp. 1266-1274, December 2004.

[3] S.R. Iyer and R. Sharda, "Prediction of athletes performance using neural networks: An application in cricket team selection," Expert Systems with Applications, vol. 36, pp. 5510-5522, April 2009.

[4] I.P. Wickramasinghe, "Predicting the performance of batsmen in test cricket," Journal of Human Sport & Excercise, vol. 9, no. 4, pp. 744-751, May 2014.

[5] Lemmer, H. H. (2008). Analysis of players' performances in the first cricket twenty 20 world cup series. South African Journal for Research in Sport, 30(2), pp.71-77.

[6] Lewis, A. J. (2005). Towards fairer measures of player performance in one-day cricket. Journal of the Operational Research Society, 56, pp.804-815.

[7] Saikia , H., Bhattacharjee, D., & Bhattacharjee, A. (2012). Is IPL Responsible for Cricketers Performance in Twenty20 World Cup? International Journal of Sports Science and Engineering, 6(2), pp.96-110.

[8] Brooks, R. Bussiere, L. F. Jennions, M. D., & Hunt, J. (2003). Sinister strategies succeed at the cricket World Cup. Proceedings of the Royal Society.

[9] "Free Download web scraping tool -web scraper | ParseHub", parsehub,[Online].Available: https://www.parsehub.com.

[10] "Data extracted from the web," Import.io, [Online].Available .