A Project Report on

### PREDICTING FUTURE PRICES OF IPL PLAYERS USING MACHINE LEARNING

Submitted to

### Jawaharlal Nehru Technological University, Hyderabad

in partial fulfillment of requirements for the award of the degree of

## BACHELOR OF TECHNOLOGY

## In

## COMPUTER SCIENCE AND ENGINEERING

By

Ghanate Arpitha (16BD1A0544) Gaddi Rohan (16BD1A054C)

K.Anirudh Reddy (16BD1A054E)

G.Jabili (16BD1A054G)

#### Under the esteemed guidance of Mr. Srinivas Mekala

Assistant Professor

Department of CSE



#### Department of Computer Science and Engineering KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTUH

3-5-1206, Narayanaguda, Hyderabad - 500029.

###### 2019 – 2020

###### KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTU, Hyderabad 3-5-1206, Narayanaguda, Hyderabad - 500029.



## CERTIFICATE

This is to certify that the project entitled “**PREDICTION OF CRICKET USING MACHINE LEARNING”** being submitted by **Ms. GHANATE ARPITHA (16BD1A0544), Mr. GADDI ROHAN (16BD1A054C), Mr. K.ANIRUDH REDDY (16BD1A054E), Ms. G.JABILI (16BD1A054G)** students of **Keshav Memorial Institute of Technology, JNTUH** in partial fulfillment of the requirements of the award of the Degree of **Bachelor of Technology in Computer Science and Engineering** as a specialization is a record of bonafide work carried out by them under my guidance and supervision in the academic year 2018 – 2019

INTERNAL GUIDE HEAD OF THE DEPARTMENT Mr. Srinivas Mekala Dr. J.V.S SRINIVAS

Assistant professor Professor and HOD

Department of CSE Department of CSE

Submitted for the Project Viva Voce examination held on …………….

EXTERNAL EXAMINER

## DECLARATION

We hereby declare that the project report entitled **“CRICKET PREDICTION USING MACHINE LEARNING”** is done in the partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. This project has not been submitted anywhere else.

Ghanate Arpitha (16BD1A0544)

Gaddi Rohan (16BD1A054C)

**K.Anirudh Reddy (16BD1A054E)**

G.Jabili (16BD1A054G)

## ACKNOWLEDGEMENT

We take this opportunity to thank all the people who have rendered their full support to our project work.

We render our thanks to **Dr. Maheshwar Dutta**, B.E.M.Tech., Ph.D., Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. Nitin Kumar**, Director and **Mrs. Deepa Abhishek Ganu**, Director for providing an excellent environment in the college.

We are also thankful to **Dr. J.V.S. Srinivas**, Head of Department for providing us with both time and amenities to make this project a success within the given schedule.

We are also thankful to our Internal Guide **Mr. Srinivas Mekala**, for his valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire **CSE Department** faculty who helped us directly and indirectly in the completion of the project.

We sincerely thank our friends and family for their constant motivation.

Ghanate Arpitha (16BD1A0544)

Gaddi Rohan (16BD1A054C)

K.Anirudh Reddy (16BD1A054E)

G.Jabili (16BD1A054G)

## CONTENTS

|  |  |
| --- | --- |
| **DESCRIPTION** | **PAGENO** |
| ABSTRACT | i |
| LIST OF FIGURES | ii |
| LIST OF ABBREVIATIONS | iii |

## CHAPTER-1

|  |  |  |
| --- | --- | --- |
| **1. INTRODUCTION** | | **1-2** |
| 1.1 | Purpose of the Project | 1 |
| 1.2 | Problem with Existing System | 1 |
| 1.3 | Proposed System | 2 |
| 1.4 | Scope of the Project | 2 |

## CHAPTER-2

|  |  |  |  |
| --- | --- | --- | --- |
| **2. SOFTWARE REQUIREMENTS SPECIFICATIONS** | | | **3-6** |
| 2.1 | Requirements Specification Document | | 3 |
|  | 2.1.1 | Functional Requirements | 3  3 |
|  | 2.1.2 | Non-Functional Requirements |
| 2.2 | Software Requirements | | 4 |
| 2.3 | Hardware Requirements | | 6 |

## CHAPTER-3

|  |  |  |
| --- | --- | --- |
| **3. LITERATURE SURVEY** | | **7-15** |
| 3.1 | Software reliability | 8 |
| 3.2 | Learning Types | 9 |
| 3.3 | Technologies Used | 10 |

|  |  |  |
| --- | --- | --- |
| 3.4 | Classification task | 14 |
| 3.5 | Optimization | 14 |
| 3.6 | Main challenges of Machine Learning | 15 |

## CHAPTER-4

|  |  |  |  |
| --- | --- | --- | --- |
| **4. SYSTEM DESIGN** | | | **16-31** |
| 4.1 | Introduction to UML | | 22 |
| 4.2 | UML Diagrams |  | 24 |
|  | 4.2.1 | Use Case Diagram | 18 |
| 4.2.2 | Sequence Diagram | 20 |
| 4.2.3 | Data Flow Diagram | 21 |

4.2.4 Class Diagram 21

## CHAPTER-5

|  |  |  |  |
| --- | --- | --- | --- |
| **5. IMPLEMENTATION** | | | **22-25** |
| 5.1 | Pseudo Code |  | 22 |
| 5.2 | Code Snippets | | 23 |
|  | 5.2.1 | Web Scraping | 23 |
|  | 5.2.2 | Converting extracted data into data frame | 24 |
|  | 5.2.3 | Merging datasets of different years | 24 |
|  | 5.2.4 | Removing NULL values from data | 25 |
|  | 5.2.5 | Removing outliers from data | 25 |
|  | 5.2.6 | Splitting the data and fitting Model | 25 |

## CHAPTER-6

|  |  |  |
| --- | --- | --- |
| **6. TESTING** | | **26-32** |
| 6.1 | Introduction to Testing | 26 |
| 6.2 | Types of Testing | 31 |

## CHAPTER-7

|  |  |  |
| --- | --- | --- |
| **7. SCREEN SHOTS** | | **33-38** |
| 7.1 | Web Scraping | 33 |
| 7.2 | Converting extracted data into data frame | 34 |
| 7.3 | Merging datasets of different years | 35 |
| 7.4 | Removing NULL values from data | 36 |
| 7.5 | Boxplot | 37 |
| 7.6 | Predicted Data | 38 |

[Future-Enhancements 39](#_TOC_250002)

[Conclusion 40](#_TOC_250001)

[References 41](#_TOC_250000)

**ABSTRACT**

The Indian Premier League (IPL) is a professional Twenty20 cricket league in India contested during April and May of every year by teams representing Indian cities and some states.

Every year, the teams are formed by bidding for players in the official auction. The objective of this project is to calculate and predict individual player value. This is useful for Team owners and cricket experts, in evaluating and predicting an approximate value of each player and helps in planning the formation of teams. The conventional way to do this is by rigorous statistical analysis; we hope to attain automation in this sector using supervised learning techniques and regression analysis.

In model development, we first analyse, predict and compare the data to the current year statistics. With further data input, we can calculate and predict the future outcomes.

## LIST OF FIGURES

|  |  |
| --- | --- |
| **DESCRIPTION** | **PAGENO** |
| 1. Use case Diagram | 18 |
| 2. Sequence Diagram | 20 |
| 3. Data Flow Diagram | 21 |
| 4. Web Scraping | 33 |
| 5. Converting the extracted data into a data frame | 34 |
| 6. Merging the datasets of the different years | 35 |
| 7. Removing the NaN values | 36 |
| 8. Boxplot showing outliers in the data | 37 |
| 9. Predictions using the fitted linear regression model | 38 |

## LIST OF ABREVATIONS

|  |  |
| --- | --- |
| ML | Machine Learning |
| PY | Python |
| DB | Data Base |
| PK | Pickle |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |

**CHAPTER-1**

1. **INTRODUCTION**

#### Purpose of the Project

The main purpose of the project is, to let the machine analyze, fit and produce a dataset that gives us the future base prices of players in an IPL auction.

Considering that, IPL popularity has boomed in India over the past few years, before every IPL there is lots of speculation about, what price each player will be sold for, in the player auction. There are many experts who try analyzing the data and coming up with certain numbers. Our machine is programmed to replicate those statistics.

This machine can be heavily by cricket experts, team owners and even players themselves.

However, for this project, our goal is not necessarily to achieve a 100% accurate future prediction of the player’s value, but to get an estimate of it for the upcoming year, based on multiple factors and variables. Therefore, making the project much more achievable to model the statistical dependencies to produce an accurate prediction.

#### Problem with the Existing System

Today, all statistical analysis is done manually, with a group of experts. However, with so many players and many more variables, it is a mountain of a task to determine and estimate each player’s value, taking up a valuable amount of time. This can instead be taught to the machine, which can then process huge amounts of data, beyond the human capability to get accurate results within a fraction of seconds.

#### Proposed System

We train the machine, to gather player performance data from the web, and store it in a dataset. This data is then refined and categorized based on the year. We remove any null values that maybe present by replacing them with the mean of the statistical data present.

The year wise data is then compiled into a single dataset, to be fit into a linear regression model, which analyses the data to give a set of estimated values.

The accuracy of this data can be cross-referenced by predicting values of the current year by taking data from the previous years.

#### Scope of the Project

This project offers a great vision with respect to the IPL. With millions invested in players, a buyer in an auction can be made aware in what ballpark the price of that particular player is based on his previous performances. This opens up many new avenues in the field of Team Building.

**CHAPTER-2**

1. **SOFTWARE REQUIREMENT SPECIFICATIONS**

#### 2.1 Requirements Specification Document

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven project, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign.

#### Functional Requirements

* Search the web for player performance data.
* IDE interface to connect the device system.
* Compilation of data into datasets categorized by years.
* Elimination of any Null values present in the data.
* Training the system with sample data of previous years and fitting it to current year statistics.

#### Non-Functional Requirements

* Performance, Response Time, Throughput
* Serviceability
* Data Integrity
* Usability & Interoperability

#### Software Requirements

* Operating System: Windows 7,8,10 / Linux/ Mac
* Platform: Jupyter Notebook (optional)
* Programming Language: Python

##### Software Setup-

#### Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

A Jupyter Notebook can be converted to a number of [open standard](https://en.wikipedia.org/wiki/Open_standard) output formats ([HTML,](https://en.wikipedia.org/wiki/HTML) [presentation slides](https://en.wikipedia.org/wiki/Presentation_slide), [LaTeX,](https://en.wikipedia.org/wiki/LaTeX) [PDF,](https://en.wikipedia.org/wiki/PDF) [ReStructuredText](https://en.wikipedia.org/wiki/ReStructuredText), [Markdown](https://en.wikipedia.org/wiki/Markdown), [Python](https://en.wikipedia.org/wiki/Python_(programming_language))) through "Download As" in the web interface, via the [nbconvert](https://nbconvert.readthedocs.io/) library or "jupyternbconvert" command line interface in a shell.

To simplify visualization of Jupyter notebook documents on the web, the nbconvert library is provided as a service through [NbViewer](https://nbviewer.org/) which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

Jupyter Notebook provides a browser-base[d REPL](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop) built upon a number of popular [open-](https://en.wikipedia.org/wiki/Open-source_software) [source](https://en.wikipedia.org/wiki/Open-source_software) libraries:

* [IPython](https://en.wikipedia.org/wiki/IPython)
* [ØMQ](https://en.wikipedia.org/wiki/%C3%98MQ)
* [Tornado (web server)](https://en.wikipedia.org/wiki/Tornado_(web_server))
* [jQuery](https://en.wikipedia.org/wiki/JQuery)
* [Bootstrap (front-end framework)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
* [MathJax](https://en.wikipedia.org/wiki/MathJax)

#### Python Packages

##### Python packages required to run this application

* Pandas
* Sklearn
* Numpy
* Matplotlib

#### Numpy

* + **NumPy** is a library for the [Python programming language,](https://en.wikipedia.org/wiki/Python_(programming_language)) adding support for large, multi-dimensional [arrays](https://en.wikipedia.org/wiki/Array_data_structure) and [matrices,](https://en.wikipedia.org/wiki/Matrix_(math)) along with a large collection of [high-](https://en.wikipedia.org/wiki/High-level_programming_language)  [level](https://en.wikipedia.org/wiki/High-level_programming_language) [mathematical](https://en.wikipedia.org/wiki/Mathematics) [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by [Jim Hugunin](https://en.wikipedia.org/wiki/Jim_Hugunin) with contributions from several other developers. In 2005, [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) and has many contributors.

#### Pandas

* + In [computer programming,](https://en.wikipedia.org/wiki/Computer_programming) **pandas** is a [software library](https://en.wikipedia.org/wiki/Software_library) written for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and [time series.](https://en.wikipedia.org/wiki/Time_series) It is [free](https://en.wikipedia.org/wiki/Free_software)  [software](https://en.wikipedia.org/wiki/Free_software) released under the [three-clause BSD license](https://en.wikipedia.org/wiki/3-clause_BSD_license). The name is derived from the term "[panel data](https://en.wikipedia.org/wiki/Panel_data)", an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for data sets that include observations over multiple time periods for the same individuals

#### Matplotlib

* + **Matplotlib** is a [plotting](https://en.wikipedia.org/wiki/Plotter) [library](https://en.wikipedia.org/wiki/Library_(computer_science)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language and its numerical mathematics extension [NumPy.](https://en.wikipedia.org/wiki/NumPy) It provides an [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [API](https://en.wikipedia.org/wiki/API) for embedding plots into applications using general-purpose [GUI toolkits](https://en.wikipedia.org/wiki/GUI_toolkit) like [Tkinter](https://en.wikipedia.org/wiki/Tkinter), [wxPython,](https://en.wikipedia.org/wiki/WxPython) [Qt](https://en.wikipedia.org/wiki/Qt_(software)), or [GTK+.](https://en.wikipedia.org/wiki/GTK%2B) There is also a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) "pylab" interface based on a [state machine](https://en.wikipedia.org/wiki/State_machine) (like [OpenGL](https://en.wikipedia.org/wiki/OpenGL)), designed to closely resemble that of [MATLAB](https://en.wikipedia.org/wiki/MATLAB), though its use is discouraged. [SciPy](https://en.wikipedia.org/wiki/SciPy)makes use of Matplotlib.

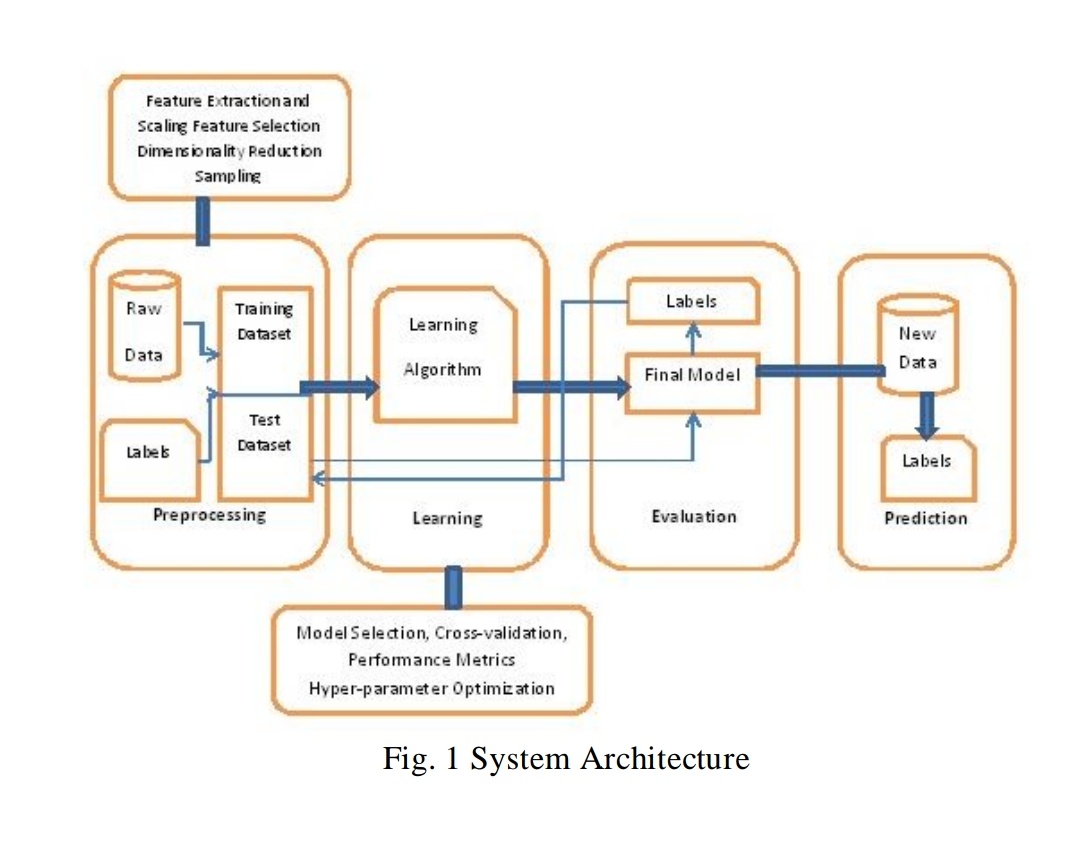
#### Scikit learn

* + **Scikit-learn** (formerly **scikits.learn**) is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programminglanguage.[[3]](https://en.wikipedia.org/wiki/Scikit-learn#cite_note-jmlr-3) Itfeaturesvarious [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clust ering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [*k*- means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN,](https://en.wikipedia.org/wiki/DBSCAN) and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy)

#### Hardware Requirements

* Processor: Intel i5
* Hard Disk: 40GB HDD
* RAM: 2 GB

#### Architecture Diagram



**CHAPTER-3**

1. **LITERARY SURVEY**

Machine learning is a branch of artificial intelligence that aims at solving real life engineering problems. It provides the opportunity to learn without being explicitly programmed and it is based on the concept of learning from data. It is so much ubiquitously used dozen a times a day that we may not even know it. The advantage of machine learning (ML) methods is that it uses mathematical models, heuristic learning, knowledge acquisitions and decision trees for decision making. Thus, it provides controllability, observability and stability.

The common feature selection techniques include wrapper subset evaluation, filtering and embedded models. Embedded models use classifiers to construct ensembles, the wrapper subset evaluation method provides ranks to features based on their importance and filter methods rank the features based on statistical measurements.

Most of the work as discussed above is contextual and related to the forum. We believe that this approach is far too restrictive, as only 5% to 10% of the students enrolled in a MOOC use the forum. We hence look at count based features such as page access, page navigation, wiki access, problems accessed etc. MOOCs have challenged the conventional education system in every way, however, themselves face an embarrassing and currently inevitable issue of very high Dropout Rates. We hence delve into the matter, considering user activity data. It is important to predict dropouts. Our goal is to understand shared practices in the user click data, and how can hidden patterns in user study behavior be used to develop a healthier and conducive learning environment.

The literature survey report consists of detail work done to predict MOOC dropouts by gaining insights into forum posts and forum click stream data. The paper considered posting patterns and number of up votes, down votes among forum interaction. The researchers also delved into linguistic features such as polarity and tone of posts on the forum to assess how the user was going about the course modules. These features were finally combined using Probabilistic Soft Logic (PSL) rules, to conclude if the user completed a course or not.

Yang et al, considered the number of threads and sub-threads that a user has started or has been a part of. The work including looking at the average number of words used on the forum, post density - number of posts in a week. The motivation behind this was to evaluate whether forum starters are survivors. Or is posting on the forum even related to finishing a course.

Finally, Sinha tried to predict dropouts by analyzing the clickstream on video modules, such as pausing, playing, forwarding, skipping video lectures.

#### Software reliability

Software reliability is defined as the probability that a system will not have a failure over a specified period of time under specific conditions. The knowledge of software reliability is very vital in critical systems because it indicates the design perfection. In this work, the primary aim is to enhance the software reliability of the computer aided diagnosis systems using machine learning algorithms.

To provide quality treatment and prevent misdiagnosis are the prime motivations for developing a medical diagnosis system. Diagnosing a disease of a patient accurately is a great challenge in medical field. A huge amount is spent on advanced primary health care devices based on software reliability search as they are considered as critical systems.

There are several software reliability models available in the literature; however, none of the models are perfect. An important research issue is choosing a suitable estimation model based on a specific application. One advantage of software reliability over hardware reliability is that a mechanical part surely undergoes ageing; suffer from wear and tear problem over time and usage; however, software does not rust or wear out. Software reliability is a vital parameter for software quality, functionality and performance. Some common software reliability models are prediction and estimation models like bathtub curve, exponential, Putnam etc.

#### Learning Types

* + 1. Supervised learning

Supervised learning is the most common form of machine learning scheme used in solving the engineering problems. It can be thought as the most appropriate way of mapping a set of input variables with a set of output variables. The system learns to infer a function from a collection of labelled training data. The training dataset contains a set of input features and several instance values for respective features. The predictive performance accuracy of a machine learning algorithm depends on the supervised learning scheme. The aim of the inferred function may be to solve a regression or classification problem. There are several metrics used in the measurement of the learning task like accuracy, sensitivity, specificity, kappa value, area under the curve etc. In this work, the aim is to classify the patients as healthy or ill based on the past medical records. Before solving any engineering problem, it is vital that it is necessary to choose a suitable algorithm for the training purpose based on the type of the data. The selection of a method depends primarily on the type of the data as the field of machine learning is data driven. The next important aspect is the optimization of the chosen machine learning algorithms

* + 1. Un-Supervised learning

Unsupervised learning is the training of an artificial intelligence ([AI](https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence)) [algorithm](https://whatis.techtarget.com/definition/algorithm) using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance.

In unsupervised learning, an AI system may group unsorted information according to similarities and differences even though there are no categories provided. AI systems capable of unsupervised learning are often associated with generative learning models, although they may also use a retrieval-based approach (which is most often associated with [supervised](https://searchenterpriseai.techtarget.com/definition/supervised-learning) [learning](https://searchenterpriseai.techtarget.com/definition/supervised-learning)). [Chatbots,](https://searchcrm.techtarget.com/definition/chatbot) [self-driving cars,](https://searchenterpriseai.techtarget.com/definition/driverless-car) [facial recognition](https://searchenterpriseai.techtarget.com/definition/facial-recognition) programs, [expert systems](https://searchenterpriseai.techtarget.com/definition/expert-system) and [robots](https://searchenterpriseai.techtarget.com/definition/robot) are among the systems that may use either supervised or unsupervised learning approaches.

In unsupervised learning, an AI system is presented with unlabeled, uncategorized data and the system’s algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI. Unsupervised learning algorithms can perform more complex processing tasks than supervised learning systems. However, unsupervised learning can be more unpredictable than the alternate model. While an unsupervised learning AI system might, for example, figure out on its own how to sort cats from dogs, it might also add unforeseen and undesired categories to deal with unusual breeds, creating clutter instead of order.

* + 1. Semi-Supervised learning

Semi-supervised machine learning is a combination of [supervised](https://www.datarobot.com/wiki/supervised-machine-learning/) and [unsupervised machine](https://www.datarobot.com/wiki/unsupervised-machine-learning/) [learning](https://www.datarobot.com/wiki/unsupervised-machine-learning/) methods.

With more common [supervised machine learning](https://www.datarobot.com/wiki/supervised-machine-learning/) methods, you [train](https://www.datarobot.com/wiki/training-validation-holdout/) a learning algorithm on a “labeled” dataset in which each record includes the outcome information. This allows the algorithm to deduce patterns and identify relationships between your [target](https://www.datarobot.com/wiki/target/) variable and the rest of the dataset based on information it already has. In contrast, [unsupervised machine](https://www.datarobot.com/wiki/unsupervised-machine-learning/) [learning](https://www.datarobot.com/wiki/unsupervised-machine-learning/) algorithms learn from a dataset without the outcome variable. In semi-supervised learning, an algorithm learns from a dataset that includes both labeled and unlabeled data, usually mostly unlabeled.

#### Technologies Used

The following are the base technologies used to develop this application:

#### Python

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. It provides constructs that enable clear programming on both small and large scales. In July 2018, the creator Guido Rossum stepped down as the leader in the language community after 30 years.

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 interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python’s other implementations. The non-profit Python Software Foundation manages Python and CPython.

**Python is interpreted -** It means that it is processed at runtime by the interpreter and you do not need to compile your program before executing it.

**Python is Interactive -** It means that you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

**Python is Object-Oriented -** It means that Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

**Python is Beginner's Language -** Python is a great language for the beginner programmers and supports the development of a wide range of applications from simple text processing to www browsers to games.

#### Web Scraping

Web scraping, web harvesting, or web data extraction is data scraping used for extracting data from websites. Web scraping software may access the World Wide Web directly using the Hypertext Transfer Protocol, or through a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying, in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

Web scraping a web page involves fetching it and extracting from it. Fetching is the downloading of a page (which a browser does when you view the page). Therefore, web crawling is a main component of web scraping, to fetch pages for later processing. Once fetched, then extraction can take place. The content of a page may be parsed, searched, reformatted, its data copied into a spreadsheet, and so on. Web scrapers typically take something out of a page,

to make use of it for another purpose somewhere else. An example would be to find and copy names and phone numbers, or companies and their URLs, to a list (contact scraping).

Web scraping is used for contact scraping, and as a component of applications used for web indexing, web mining and data mining, online price change monitoring and price comparison, product review scraping (to watch the competition), gathering real estate listings, weather data monitoring, website change detection, research, tracking online presence and reputation, web mashup and, web data integration.

Web pages are built using text-based mark-up languages (HTML and XHTML), and frequently contain a wealth of useful data in text form. However, most web pages are designed for human end-users and not for ease of automated use. Because of this, tool kits that scrape web content were created. A web scraper is an Application Programming Interface (API) to extract data from a web site. Companies like Amazon AWS and Google provide web-scraping tools, services and public data available free of cost to end users.

Newer forms of web scraping involve listening to data feeds from web servers. For example, JSON is commonly used as a transport storage mechanism between the client and the web server.

There are methods that some websites use to prevent web scraping, such as detecting and disallowing bots from crawling (viewing) their pages. In response, there are web-scraping systems that rely on using techniques in DOM parsing, computer vision and natural language processing to simulate human browsing to enable gathering web page content for offline parsing.

#### Linear Regression Modelling

In statistics, linear regression is a linear approach to modelling the relationship between a scalar response or more independent variables. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear

models. Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

Linear regression has many practical uses. Most applications fall into one of the following two broad categories:

* If the goal is prediction, or forecasting, or error reduction, linear regression can be used to fit a predictive model to an observed data set of values of the response and explanatory variables.
* If the goal is to explain variation in the response variable that can be attributed to variation in the explanatory variables, linear regression analysis can be applied to quantify the strength of the relationship between the response and the explanatory variables, and in particular to determine whether some explanatory variables may have no linear relationship with the response at all, or to identify which subsets of explanatory variables may contain redundant information about the response.

Linear regression models are often fitted using the least squares approach, but they may also be fitted in other ways, such as by minimizing the "lack of fit" in some other norm, or by minimizing a penalized version of the least squares cost function as in ridge regression (*L*-norm penalty) and lasso (*L*-norm penalty). Conversely, the least squares approach can be used to fit models that are not linear models. Thus, although the terms "least squares" and "linear model" are closely linked, they are not synonymous.

#### Classification task

Classification task is a classical problem in the field of data mining which deals with assigning a pre-specified class to an unknown data. A learning model is built based on the relationship between the predictor attribute values and the value of the target. The challenge is to correctly predict the class based on learning of past data. In machine learning, this kind of classification problems are referred to as supervised learning. Hence, we need to provide a data set containing instances with known classes and a test data set for which the class has to be determined. The success of the classification ability largely depends on the quality of data provided for learning and also the type of machine learning algorithm used. For example, the classification techniques can be used to predict the fraud customers in a bank who apply for a loan or classify mangoes whether they are good or bad and lots of other real time applications. The most common type of classification problem is binary classification, where the target has two possible values like good or bad, yes or no etc. There are several methods for measuring the classification performance like confusion matrix, lift curve, receiver operator characteristics etc.

#### Optimization

Every machine learning algorithm has a specific technique of learning and is based on the values of their parameters. When an algorithm is applied to solve a classification problem with a different set of parameters, the classification accuracy also differs abruptly in each case. The challenge in machine learning to find the most suitable parameter values of the algorithms that solves an engineering problem to the best possible way in terms of performance metrics. Therefore, one has to fine tune the algorithm parameters that best suits the problem. There are several optimization techniques like genetic algorithm, particle swarm optimization, Taboo search methods etc. The focus of the study is to calibrate the algorithm parameters using design of experiment method.

#### Main challenges of Machine Learning

In short, since the main task is to select a learning algorithm and train it on some data, the two things that can go wrong are "bad algorithm" and "bad data."

1. Insufficient Quantity of Training Data
2. Non-representative Training Data
3. Poor-Quality Data
4. Irrelevant Features
5. Over fitting the Training Data
6. Under fitting the Training Data

**CHAPTER-4**

* 1. **SYSTEM DESIGN**

#### Introduction to UML

The Unified Modeling Language (UML) is a standard visual modeling language intended to be used for modeling business and similar processes and analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

UML can be applied to diverse application domains (e.g., banking, finance, internet, aerospace, healthcare, etc.) It can be used with all major object and component software development methods and for various implementation platforms (e.g., J2EE, .NET). It is a standard modeling language, not a software development process. UML Specification explained that process:

* provides guidance as to the order of a team’s activities,
* specifies what artifacts should be developed,
* directs the tasks of individual developers and the team as a whole, and
* Offers criteria for monitoring and measuring a project’s products and activities. UML is intentionally process independent and could be applied in the context of different processes. Still, it is most suitable for use case driven, iterative and incremental development processes. An example of such process is Rational Unified Process (RUP).

UML is not complete and it is not visual. Given some UML diagram, we cannot be sure to understand depicted part or behavior of the system from the diagram alone. Some information could be intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams.

## UML Diagrams

#### Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify different types of users of a system and the different use cases. It is often be accompanied by other types of diagrams as well. While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that "Use case diagrams are the blueprints for your system". They provide the simplified and graphical representation of what the system must actually do.

Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed. Siau and Lee conducted research to determine if there was a valid situation for use case diagrams at all or if they were unnecessary. What was found was that the use case diagrams conveyed the intent of the system in a more simplified manner to stakeholders and that they were "interpreted more completely than class diagrams".

The purpose of the use case diagrams is simply to provide the high-level view of the system and convey the requirements in layman's terms for the stakeholders. Additional diagrams and documentation can be used to provide a complete functional and technical view of the system.

To model a system, the most important aspect is to capture the dynamic behavior. To clarify bit in details, dynamic behavior means the behavior of the system when it is running /operating. The diagram is used to model the system/subsystem of an application.

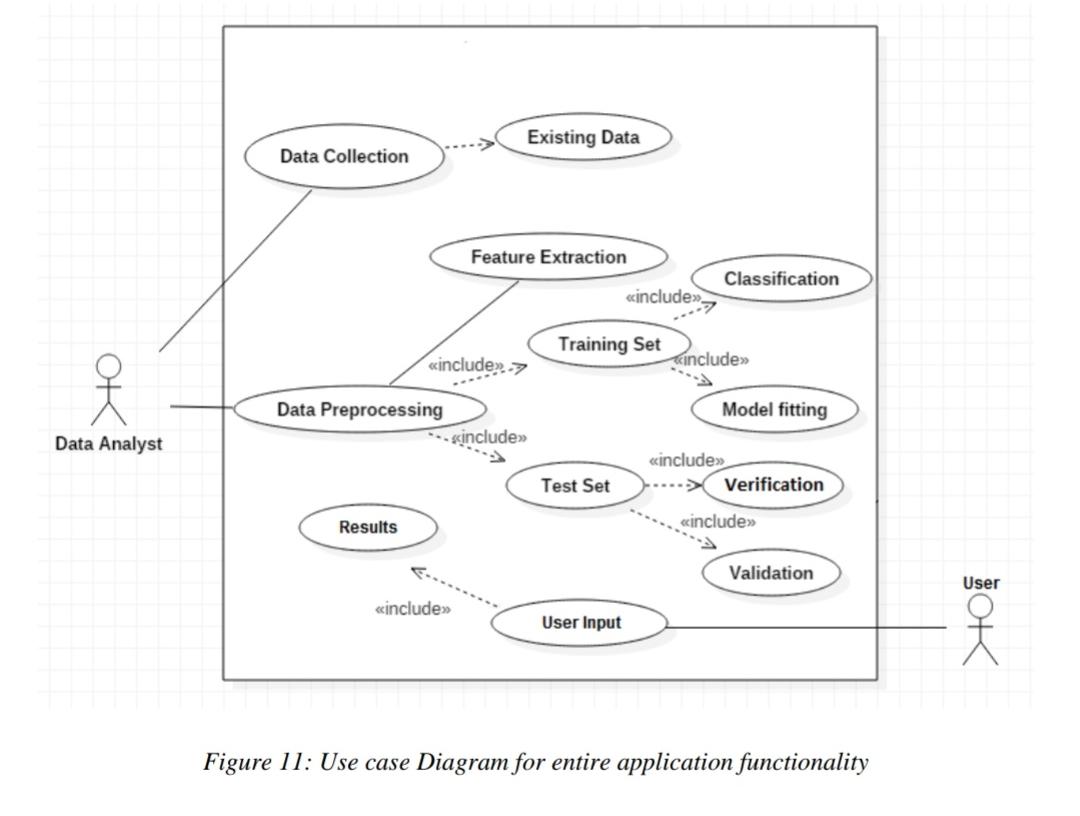
So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model dynamic

nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. These internal and external agents are known as actors. Therefore, use case diagrams consist of factors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

So to model the entire system numbers of use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

*Figure 1: U* *se Case Diagram*



#### Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

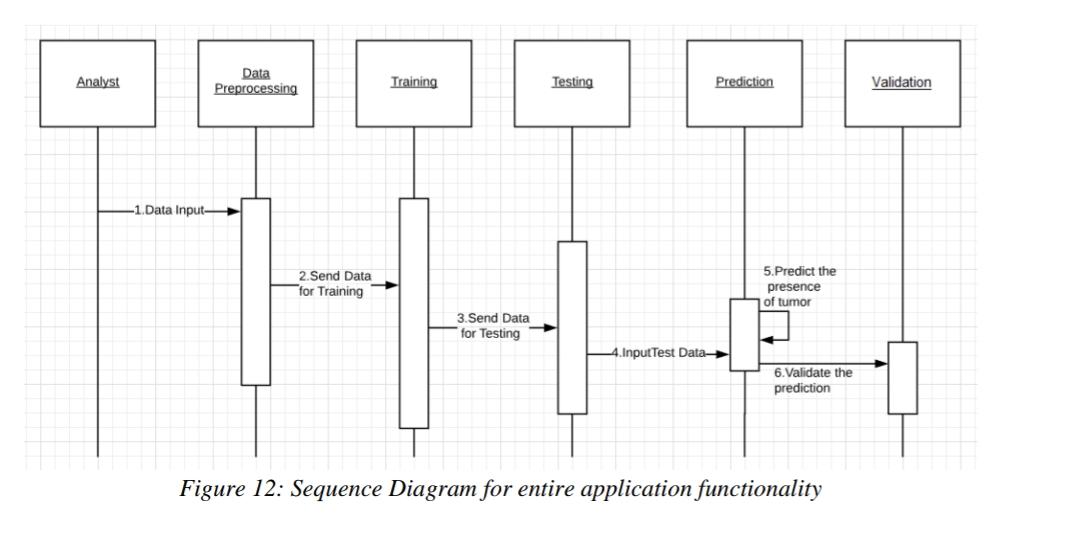
If the lifeline is that of an object, it demonstrates a role. Leaving the instance name blank can represent anonymous and unnamed instances.

Messages, written with horizontal arrows with the message name written above them, display interaction. Solid arrowheads represent synchronous calls, open arrowheads represent asynchronous messages, and dashed lines represent reply messages. If a caller sends a synchronous message, it must wait until the message is done, such as invoking a subroutine.

If a caller sends an asynchronous message, it can continue processing and does not have to wait for a response. Asynchronous calls are present in multithreaded applications, event-driven applications and in message-oriented middleware. Activation boxes, or method -call boxes, is opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message

Objects calling methods on themselves use messages and add new activation boxes on top of any others to indicate a further level of processing. If an object is destroyed (removed from memory), an X is drawn on bottom of the lifeline, and the dashed line ceases to be drawn below it. It should be the result of a message, either from the object itself, or another. A message sent from outside the diagram can be represented by a message originating from a filled-in circle (found message in UML) or from a border of the sequence diagram (gate in

UML).



#### DFD Diagram

*Figure 2: Sequence Diagram*

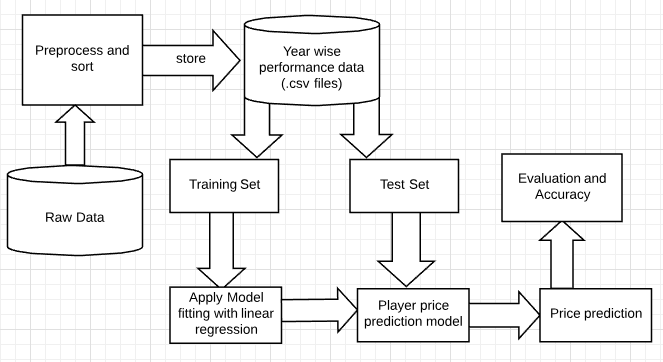
A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system. How the data will advance through the system, and where the data will be stored. It does not show information about process timing. Whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart, which focuses on control flow, or a UML activity workflow diagram, which presents both control and data, flows as a unified model.

Data flow diagrams can be used in both the Analysis and Design phases of the SDLC.

There are different notations to draw data flow diagrams (Yourdon & Coad and Gane & Sarson), defining different visual representations for processes, data stores, data flow, and external entities.

The DFD serves two purposes:

* 1. To provide an indication of how data are transformed as they move through the system.
* 2. To depict the function and sub-functions that transforms the data.



*Figure 3: Data Flow Diagram*

* + 1. **Class Diagram**

Class diagrams are the main building blocks of every object oriented methods. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in its context. It describes various kinds of objects and the static relationship in between them. The main purpose to use class diagrams are:

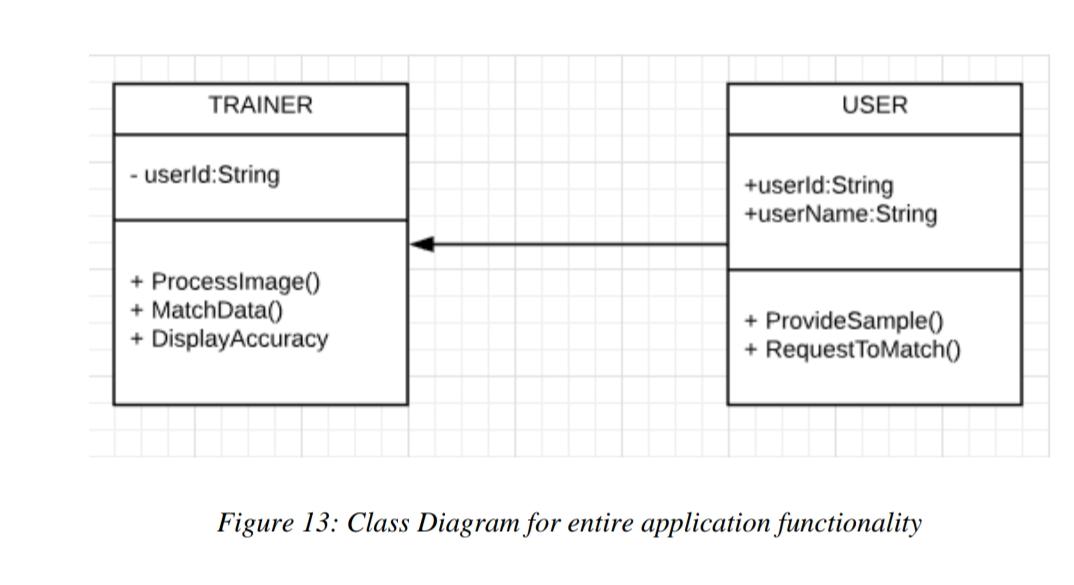
• This is the only UML which can appropriately depict various aspects of OOPs concept.

• Proper design and analysis of application can be faster and efficient.

• It is base for deployment and component diagram. Each class is represented by a rectangle having a subdivision of three compartments name, attributes and operation. Histopathological Cancer Detection using Image Recognition

•There are three types of modifiers which are used to decide the visibility of attributes and operations.

• + is used for public visibility (for everyone)



**CHAPTER-5**

* 1. **IMPLEMENTATION**

#### Pseudo code

Pseudocode is an informal high-level description of the operating principle of a computer program or other algorithm.

Step 1:

Select the functionality for which the script must be written.

Step 2:

Import the necessary packages.

Step 3:

Collect data from the internet via Web Scraping.

Step 4:

Pre-Process the data.

If (Data Label == “2016”) Insert into DF\_2016 If (Data Label == “2017”) Insert into DF\_2017 If (Data Label == “2018”) Insert into DF\_2018

Step 5:

Create the model and train it.

Step 6:

Run the console.

Step 7:

Check the result

Step 8:

Stop

#### Code Snippets

* + 1. Web Scraping:

Web scraping is the process of extracting data from websites. It is useful when we have to extract a large amount of data that would take a long time for manual extraction. We used web scraping to create the dataset containing the points and base prices of players since the dataset was not already available.

import requests

page = requests.get("http[s://www.iplt20.com/st](http://www.iplt20.com/stats/2018/player-points)a[ts/2018/player-points"](http://www.iplt20.com/stats/2018/player-points)) page

from bs4 import BeautifulSoup

soup = BeautifulSoup(page.content, 'html.parser') sb=[]

#sb1=[]

#name=row.findAll('a')lname=row.findAll('span',class\_='top-players last-name') fours=[]

sixes=[]

for i in range(0,149):

sb.append(soup.find\_all('td',class\_='top-players pts top-players padded is- active')[i].get\_text().lstrip().rstrip())

#print(sb)

for i in range(0,149):

sixes.append(soup.find\_all('td',class\_='top-players 6s')[i].get\_text().lstrip().rstrip()) fours.append(soup.find\_all('td',class\_='top-players 4s')[i].get\_text().lstrip().rstrip())

table=soup.find('table',class\_='top-players') #print(table)

sb1=[]

for row in table.find\_all('tr'): name=row.findAll('a')

#lname=row.findAll('span',class\_='top-players last-name')

if len(name)==1:

sb1.append((name[0].get\_text().strip().replace(' ','').replace('\n',' '))) #print(lname)

* + 1. Converting the extracted data into a data frame:

import pandas as pd df=pd.DataFrame(sb,columns=['ponits']) df['playername']=sb1

df['4s']=fours

df['6s']=sixes df

* + 1. Merging the datasets of the different years:

import pandas as pd ba=pd.DataFrame(b['Player Name']) ba['Base Price']=b['Base Price'] bt=pd.DataFrame(df3['Player Name']) bt['Base Price']=df3['Base Price']

target=pd.merge(ba,bt,on='Player Name',how='outer') target

df3=df3.rename(columns={'Base Price':'Base Price 17'}) df3

merge18=merge18.rename(columns={'Base Price':'Base Price 18'}) merge18

merge18=merge18.rename(columns={'Base Price':'Base Price 18'})

merge18

df16drop=df3

df16drop=df16drop.drop(df16drop.columns[[2,3,5,6,7]],axis=1)

df16drop=df16drop.rename(columns={'points':'points 17','Base Price':'Base Price 17'}) df16drop

mergedf16b=pd.merge(df16drop,bdrop,on='Player Name',how='outer') mergedf16b=mergedf16b.rename(columns={'Base Price':'Base Price 16','points':'points 16'}) mergedf16b.count()

mergeplayer=pd.merge(mergedf16b,merge18drop,on='Player Name',how='outer')

mergeplayer

* + 1. Removing the NaN values:

mergeplayer=mergeplayer.dropna(how='any') mergeplayer=mergeplayer.reset\_index(drop=True)

mergeplayer

* + 1. Removing outliers from the data:

x['Base Price 18']=y quant=x.quantile([0.00,0.97]) quant

remove=x.apply(lambda q:q[(q>quant.loc[0,q.name])&(q<quant.loc[.97,q.name])],axis=0) remove=remove.dropna().reset\_index(drop=True)

remove.boxplot()

plt.show()

* + 1. Splitting the data and fitting the linear regression model:

x=pd.DataFrame(mergeplayer['Base Price 16'])

x['points 16']=mergeplayer['points 16']

x['points 17']=mergeplayer['points 17']

x['Base Price 17']=mergeplayer['Base Price 17'] y=pd.DataFrame(mergeplayer['Base Price 18'])

lr=reg.fit(x,y)

**CHAPTER-6**

#### Introduction to Testing

* 1. **TESTING**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test

* + - Meets the requirements that guided its design and development,
    - Responds correctly to all kinds of inputs,
    - Performs its functions within an acceptable time,
    - Is sufficiently usable,
    - Can be installed and run in its intended environments and
    - Achieves the general result its stakeholder’s desire.

#### Advantages

1. Software testing helps in **identifying and fixing bugs** before the software becomes operational, the risk of failure can be reduced considerably.
2. Any software does not necessarily work alone. Sometimes it has to **integrate and function with other existing legacy systems**, as need be. In such cases software testing gives a much needed assurance that it will work suitably and its performance won’t get affected due to the integration.
3. Software testing is a part of the software development process. It performs a **root cause analysis** for which helps in making it more efficient.
4. Software testing is carried out keeping the **end user in mind**. With a foresight for the scenarios that the end user will possibly face, tests are carried out and their accuracy and efficiency are duly noted for any discrepancies.
5. **Reliability of software** can be measured by certification or its conformation with any set technical standard. This helps in creating confidence and a healthy business relationship.

#### Validation

Data validation is intended to provide certain well-defined guarantees for fitness, accuracy, and consistency for any of various kinds of user input into an application or automated system. Data validation rules can be defined and designed using any of various methodologies, and be deployed in any of various contexts

Want to enable first-class analytics? Then ensure quality data or data which is clean, relevant and useful. How do you do that? To ensure that your data is clean, relevant and useful, you need to perform [data validation.](https://www.ze.com/wp-content/uploads/2015/04/DataValidation04-04-2015.pdf) Performing data validation before running any data analysis can help you get better results. Often, a business would use a set of data for years only to discover that the data that it had been using was wrong all along, causing the business to lose trust in the data warehouse environment.

The above demonstrates exactly why businesses need to take the quality of their data more seriously and perform data validation. If you don’t want to learn the importance of data validation the hard or costly way, then you need to take measures that ensure the quality of your data.

Often a topic of great importance, data validation is required to maintain the integrity of databases. Having valid data is the rule rather than the exception since information is constantly being updated, queried, deleted or shared around. With data validation, databases can be made more consistent and functional, allowing them to provide more value to users.

While validation checks may be part of your standard process, they may not be enough for you ensure the quality of your data. So, how can you effectively perform data validation to improve your data quality? Following are some recommended data validation techniques that you can follow.

#### Kinds of Validation

* 1. Data type validation;
  2. Range and constraint validation;
  3. Code and Cross-reference validation; and
  4. Structured validation

#### Validation Methods

1. Allowed character checks
2. Batch totals
3. Cardinality check
4. Check digits
5. Data type checks
6. Logic check
7. Limit check

Data-type validation

Data type validation is customarily carried out on one or more simple data fields. The simplest kind of data type validation verifies that the individual characters provided through user input are consistent with the expected characters of one or more known primitive data types; as defined in

a programming language or data storage and retrieval mechanism as well as the specification of the following primitive data types:

1. integer;
2. float (decimal); or
3. String.

Simple range and constraint validation

Simple range and constraint validation may examine user input for consistency with a minimum/maximum range, or consistency with a test for evaluating a sequence of characters, such as one or more tests against regular expressions. For example, a US phone number should have 10 digits and no letters or special characters

Code and cross-reference validation

Code and cross-reference validation includes tests for data type validation, combined with one or more operations to verify that the user-supplied data is consistent with one or more external rules, requirements, or validity constraints relevant to a particular organization, context or set of underlying assumptions. These additional validity constraints may involve cross-referencing supplied data with a known look-up table or directory information service.

Structured validation

Structured validation allows for the combination of any of various basic data type validation steps, along with more complex processing. Such complex processing may include the Testing of conditional constraints for an entire complex data object or set of process operations within a system.

Allowed character checks

Checks to ascertain that only expected characters are present in a field. For example, a numeric field may only allow the digits 0–9, the decimal point and perhaps a minus sign or commas. A text field such as a personal name might disallow characters such a markup-based security attack. An e-mail address might require at least one @ sign and various other structural details.

Regular expressions are effective ways of implementing such checks. (See also data type checks below)

Batch totals

Checks for missing records. Numerical fields may be added together for all records in a batch. The batch total is entered and the computer checks that the total is correct, e.g., add the 'Total Cost' field of a number of transactions together.

Cardinality check

Checks that record has a valid number of related records. For example, if Contact record classified as a Customer it must have at least one associated Order (Cardinality > 0). If order does not exist for a "customer" record, then it must be either changed to "seed" or the order must be created. This type of rule can be complicated by additional conditions. For example, if contact record in Payroll database is marked as "former employee", then this record must not have any associated salary payments after the date on which employee left organization (Cardinality = 0).

Check digits

Used for numerical data. An extra digit is added to a number which is calculated from the digits. The computer checks this calculation when data are entered. For example, the last digit of an ISBN for a book is a check digit calculated modulus 10

Data type checks

Checks the data type of the input and give an error message if the input data does not match with the chosen data type, e.g., In an input box accepting numeric data, if the letter 'O' was typed instead of the number zero, an error message would appear.

Limit check

Unlike range checks, data are checked for one limit only, upper OR lower, e.g., data should not be greater than 2 (<=2).

Logic check

Checks that an input does not yield a logical error, e.g., an input value should not be 0 when it will divide some other number somewhere in a program.

#### TYPES OF TESTING

* + 1. Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs 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.

* + 1. Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

* + 1. Functional Testing

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 centred 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.

**System/Procedures**: interfacing systems or procedures must be invoked.

* + 1. System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of 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.

* + 1. 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.

* + 1. White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

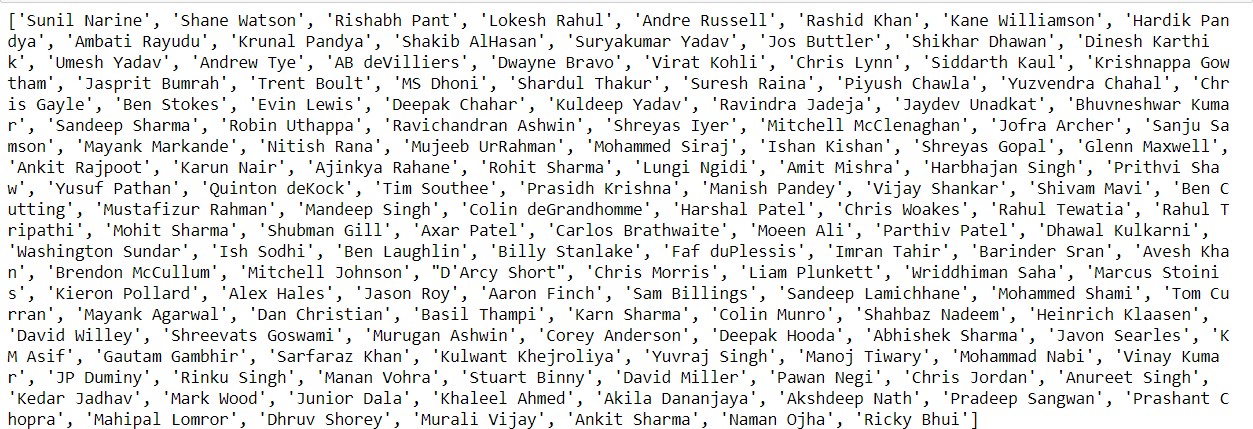
* + 1. Black Box Testing

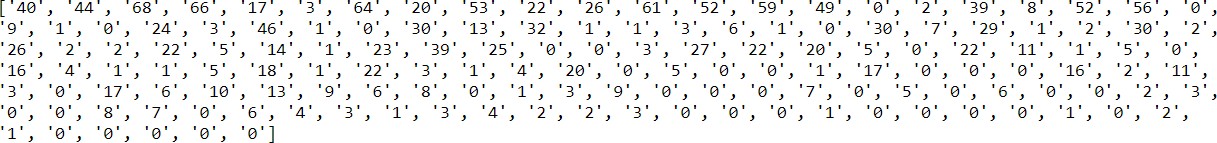
Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot ―see‖ into it. The test provides inputs and responds to outputs without considering how the software works.

**CHAPTER-7**

* 1. **SCREENSHOTS**

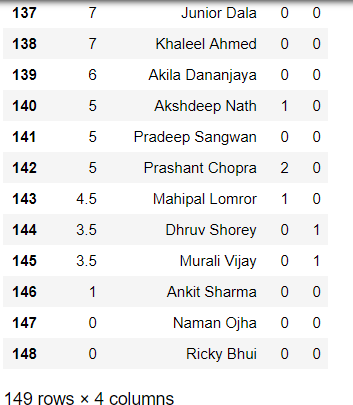
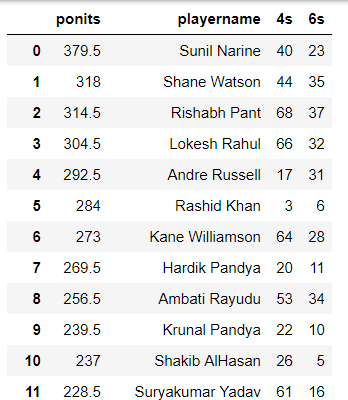
#### Web Scraping:





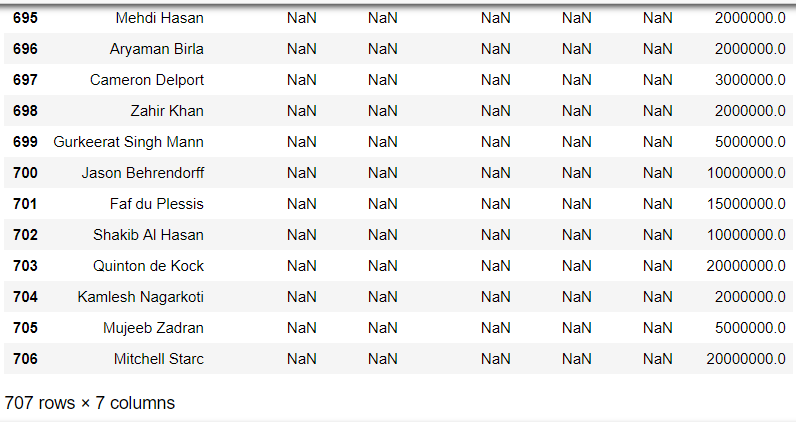
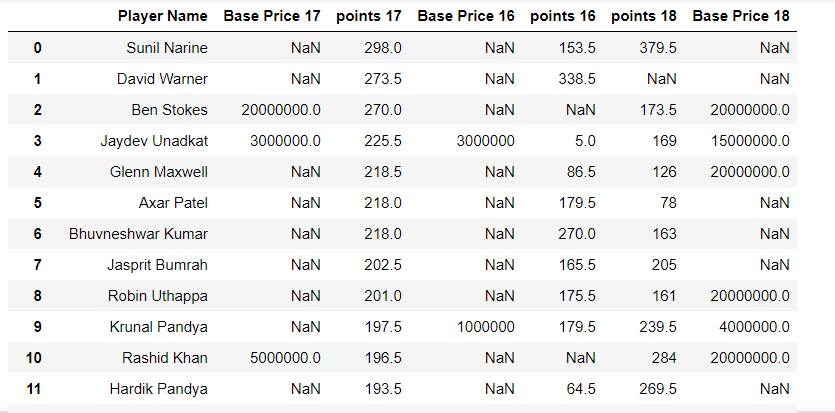
*Figure 4: Data Collected from the internet via Web Scraping*

#### Converting the extracted data into a data frame:



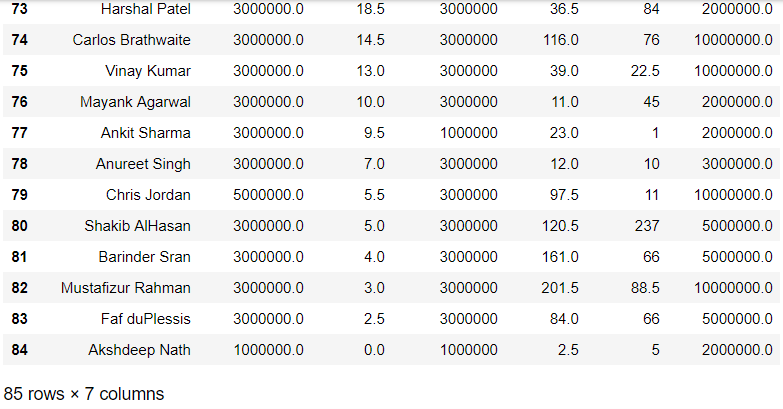
*Figure 5: Sorting collected data into data frames*

#### Merging the datasets of the different years:



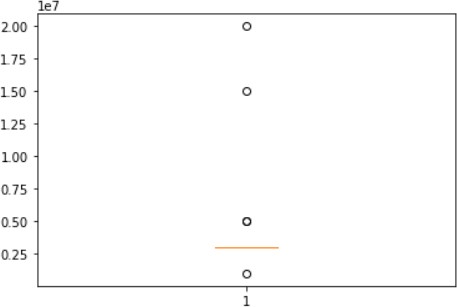
*Figure 6: Merging data frames of different years*

#### Removing the NaN values:



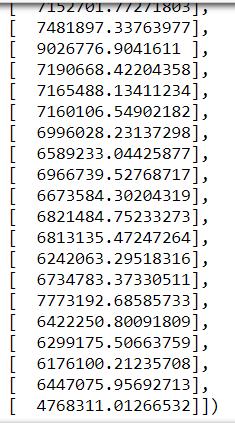
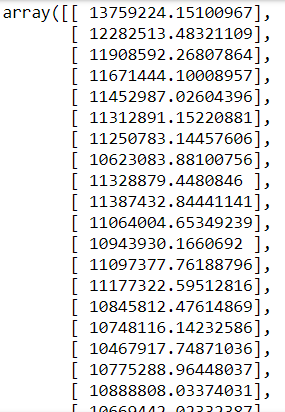
*Figure 7: Removing Null and invalid values*

#### Boxplot:



*Figure 8: Boxplot showing the outliers in the data*

#### Predictions using the fitted linear regression model:



*Figure 9: Data after prediction algorithm is applied*

# FUTURE ENHANCEMENTS

This System can be further developed to have an increased accuracy percentage. By factoring in more parameters and data, we can take make better predictions of the prices of players playing in IPL.

The downside of the system in inconsistent data, solutions can be found to tackle this issue and improve upon it.

As this is a basic approach to the statement of “Predicting future prices of IPL players”, it’s not perfect. Further improvements can be made by using a deeper learning algorithm than what is currently being used.

# CONCLUSION

IPL is a much beloved sport in India and there is always speculation about the teams that will be formed, arrival of new players and the prices at which players are sold to different teams.

Many experts are put to this task of prediction and analysis of data. With the system created for this project, it can be developed to replace aforementioned experts.

With the use of Linear Regression, a predicted dataset of expected prices of players in the oncoming IPL was created, based on past performance of players and various other form factors. Although the current accuracy is not high it can be improved upon very easily by factoring in more parameters.

# REFERENCES

* + 1. <https://www.iplt20.com/stats/>
    2. <https://machinelearningmastery.com/>
    3. Mastering Machine Learning with scikit-learn by GavinHackeling
    4. Hands-On Machine Learning with Scikit-Learn & TensorFlow by Aurelien Geron
    5. Data Preprocessing by Jun Du