**Cricket Match Prediction**

**Capstone Project Report**

**Submitted by:**

**(101410029)Jatin Singhal**

**(101410037)Mokshit Kumar**

**(101403209)Yatin Dang**

**BE Fourth Year, Computer Engineering**

**Lab Group: CAG-1, Project Team No. 7**

Under the Mentorship of

Dr. Rupali Bhardwaj

Assistant Professor

Thapar University



**Computer Science and Engineering Department**

**Thapar Institute of Engineering and Technology, Patiala**

**December 2017**

**TABLE OF CONTENTS**

1. ABSTRACT
2. DECLARATION
3. ACKNOWLEDGEMENTS
4. CONTENTS
5. LIST OF TABLES
6. LIST OF FIGURES

**ABSTRACT**

With the advent of statistical modeling in sports, predicting the outcome of a game has been established as a fundamental problem. Cricket is one of the most popular team games in the world. Cricket is the second most watched sport in the world after soccer, and enjoys a multi-million dollar industry. There is remarkable interest in simulating cricket and more importantly in predicting the outcome of cricket match which is played in three formats namely test match, one day international and T20 match. The complex rules prevailing in the game, along with the various natural parameters affecting the outcome of a cricket match present significant challenges for accurate prediction. Several diverse parameters, including but not limited to cricketing skills and performances, match venues and even weather conditions can significantly affect the outcome of a game. We embark on predicting the outcome of a One Day International (ODI) cricket match using a supervised learning approach from a team composition perspective. We first preprocessed the data by filling the missing entries with appropriate values and then by changing the format of the specific features to make it suitable for model generation. We then generated newer features like home team advantage, strength of each team and its performance in past few matches. Our project suggests that the relative team strength between the competing teams forms a distinctive feature for predicting the winner. Modeling the team strength boils down to modeling individual player’s batting and bowling performances, forming the basis of our approach. We use career statistics as well as the recent performances of each player to model him. Player independent factors like bating and bowling averages have also been considered in order to predict the outcome of a match. We then used various machine learning classifiers like decision tree, random forest, logistic regression and support vector machines to model our data. We used K Fold cross validation technique while modelling the data so that the whole dataset can be used for training as well as testing. We then calculated the accuracy of each model by dividing the number of right predictions out of total predictions made. Out of all the models used we found out that the logistic regression gives us the better accuracy. We finally came to conclusion that along with factors like toss decision, venue of the match, prediction depends on their player’s statistics like batting average, bowling average, their performance in past few matches.

**DECLARATION**

We hereby declare that the project entitled Cricket Match Prediction is an authentic record of our own work carried out in the Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, under the guidance of Dr. Rupali Bhardwaj during 6th and 7th semester (2017).

Date:

|  |  |  |
| --- | --- | --- |
| **Roll No** | **Name** | **Signature** |
| 101410029 | Jatin Singhal | ------------ |
| 101410037 | Mokshit Kumar | ------------ |
| 101403209 | Yatin Dang | ------------ |
|  |  |  |

Counter Signed by

Faculty Mentor:

Dr. Rupali Bhardwaj

Assistant Professor

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology, Patiala

**ACKNOWLEDGEMENTS**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to our mentor Dr. Rupali Bhardwaj for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the project.

We would like to express our gratitude towards programme coordinator Dr. Inderveer Chana & the whole team for their kind co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

**CONTENTS**

1 Introduction 10

1.1 Project Overview 10

1.2 Need Analysis 11

1.3 Problem Definition and Scope of the Project 12

1.4 Approved Objectives 13

1.5 Methodology Used 13

1.6 Project Outcomes 14

1.7 Novelty of Work 15

2 Requirement Analysis 15

2.1 Literature Survey 15

2.2 Analysis 18

2.3 SRS 21

2.3.1 Introduction 21

2.3.2 Overall Description 22

2.3.3 System Features 23

2.3.4 External Interface Requirements 24

2.3.5 Other Non-Functional requirements 25

2.3.6 Other Requirements 26

2.4 Analysis Model 26

2.4.1 Activity Diagram 26

2.4.2 Class Diagram 27

2.4.3 Data Flow Diagram 28

2.4.4 Sequence Diagram 29

2.4.5 Use Case Diagram 30

2.5 Cost Analysis 31

2.6 Assumptions and Constraints 31

3. Design Specifications 32

3.1 Flow Chart of the System 32

3.2 Design Phases 33

3.3 User Interface Snapshots 35

3.4 System Components and Relevant Algorithms 36

4 Results and Evaluations 38

4.1 Testing Process 38

4.2 Inference Drawn 38

5 Conclusion and Future Directions 39

5.1 Conclusions 39

5.2 Environmental/Economic/Social Benefits 39

5.3 Reflections or Self Learning 39

5.4 Future Direction 40

6 Project Metrics 40

6.1 Challenges Faced 40

6.2 Relevant Subjects 40

6.3 Interdisciplinary Knowledge Sharing 41

6.4 Peer Assesment Matrix 41

6.5 Role Playing and Project Schedule 42

6.6 Student Outcome (A-K) Mapping 42

6.7 Brief Analytical Assesment 43

7. Refernces 44

**LIST OF TABLES**

[Table 6.1 Subject Code and Subject Name](#_Toc500684599) 41

[Table 6.2 Peer assessment within the team](#_Toc500684600) 42

**LIST OF FIGURES**

[Figure 2.1 User Interface 18](#_Toc500684653)

[Figure 2.2 System Design 19](#_Toc500684654)

[Figure 2.3 Model Evaluation 21](#_Toc500684655)

[Figure 2.4 Activity Diagram 26](#_Toc500684656)

[Figure 2.5. Class digram 27](#_Toc500684657)

[Figure 2.6. Data Flow Diagram (level 0) 28](#_Toc500684658)

[Figure 2.7. Data Flow Diagram (level 1) 28](#_Toc500684659)

[Figure 2.8. Sequence Diagram 29](#_Toc500684660)

[Figure 2.9. Use Case Diagram 30](#_Toc500684661)

[Figure 3.1. Flow Chart 32](#_Toc500684662)

[Figure 3.2. User Interface with Results 35](#_Toc500684663)

[Figure 3.3. Command Line Results 36](#_Toc500684663)

**1. Introduction**

**1.1 Project Overview**

Statistical modeling has been used in sports since decades and has contributed significantly to the success on field. Cricket is one of the most popular sports in the world. Various natural factors affecting the game, enormous media coverage, and a huge betting market have given strong incentives to model the game from various perspectives. However, the complex rules governing the game, the ability of players and their performances on a given day, and various other natural parameters play an integral role in affecting the final outcome of a cricket match. This presents significant challenges in predicting the accurate results of a game. The game of cricket is played in three formats - Test Matches, ODIs and T20s. We focus our research on ODIs, the most popular format of the game. To predict the outcome of ODI cricket matches, we propose an approach where we first estimate the batting and bowling potentials of the 22 players playing the match using their career statistics and active participation in recent games. We then use these player potentials to render the relative dominance one team has over the other. Taking two other base features into account, namely, toss decision and the venue of the match, along with the relative team strength, we adopt supervised learning algorithms to predict the winner of the match.

We initiate our project by fetching data from sites like ESPN, Kaggle, and Cricksheet. The data we obtained is in the form of csv file where each file describes each match details. Then we generated python scripts to club the data entries of each and every match. We then applied feature selection to select features like teams involved, match venue, toss decision and winner of the match. We then used feature generation to create more important features like strength of each team, performance of each team in past few matches, probability of team batting first and winning probability of particular team at a specific location based on all the previous matches played between those two teams at the specific venue. The strength and performances of each match are calculated in terms of relative strength and relative performances. The strength of each team in a particular match is calculated based on the batting and bowling averages of each player in that match. The strength of each team is calculated by subtracting the batting average of the team with the bowling average. Further the team batting average is calculated by taking the mean of batting averages of all the players of the team and the bowling average is calculated by taking the mean of bowling averages of all the players of the team. After getting the strength of each team we subtract the strength of team A with team B to get relative strength of overall match. If relative strength is positive it means team A is stronger than team B. If negative it means team B is stronger than team A. Else both are equally strong. Along with strength we have also considered the past few match performances of each team as it tell us whether the team is in good form or not. The performances are calculated by taking mean batting average of past ten matches of each team. After getting individual team performances we then calculate the relative performance of the team by subtracting team A performance with team B performance. If the relative performance is positive it means team A is in better form than team B. If negative then it means team B is in better form. Otherwise both are in equal form. The magnitude tells us the amount with which one team is better than other.

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

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

Batting Average of Team = ∑ (Bat average of each player)/No. of players

Bowling Average of Team= ∑ (Bowl average of each player)/No. of players

After preprocessing and feature generating we used the machine learning to generate models which can be used to predict the results. We used different models and calculated the accuracy from each model and finally selected the one with best accuracy. We used decision tree, random forest, logistic regression and SVM. Out of all these we used logistic regression as it gave us the better accuracy of 65.5% among all other models. While generating the models we used K Fold cross validation technique so as to make use of whole dataset as training as well as testing. After generating models we developed a GUI in python using tkinter module so that user can easily interact with our system rather than going to command prompt and typing all the fields.

**1.2 Need Analysis**

Cricket was one of the first sports to use statistics as a tool for illustration and comparison. Although compared to other sports, there has not been much statistical modeling work done for cricket. For baseball, Ganeshapillai and Guttag (2013) developed a prediction model that decides when to change the starting pitcher as the game progresses. It is very much similar to our work-flow, where they used the combination of previous data and in game data to predict a pitchers performance. Tulabandhula and Rudin (2014) were designed a real time prediction and decision system for professional car racing. Model makes the decision of when is the best time for tire change and how many of them. Wood (1945) used the geometric distribution to model the total score, while Kimber and Hansford (1993) proposed a nonparametric approach based on runs scored for assessing batting performance. The most common and popular form of cricket is the One Day International (ODI), where over 50- overs per side are played. As is typical in games of sport, winning is the ultimate goal. Some studies, (De Silva, 2001), analyze the magnitude of the victory, but most consider the factors affecting winning. Kaluarachchi and et al (2010) takes into account various factors affecting the game including home team advantage, day/night effect and toss, etc., and uses the Bayesian classifier to predict the outcome of the match. Sankaranarayanan et al (2014) used machine learning approach to predict the result of a one day match depending on the previous data and in game data. Sohail Akhtar and Philip Scarf (2012) have forecasts match outcomes in test cricket in play, session by session. Match outcome probabilities at the start of each session are forecast using a sequence of multinomial logistic regression models. These probabilities can facilitate a team captain or management to consider an aggressive or defensive batting strategy for the coming session. But these probabilities can be increased by taking into consideration factors like team composition and their performance in past few matches, the batting averages and the bowling averages of each player in the team and the winning probability of team at a specific venue against a specific team.

**1.3 Problem Definition**

Cricket is the most popular sport in India. Various natural factors affecting the game, enormous media coverage, and a huge betting market have given strong incentives to model the game from various perspectives. However, the complex rules governing the game, the ability of players and their performances on a given day, and various other natural parameters have not been taken into consideration while designing the prediction softwares which are already existing in the market. As a result their accuracy fails and prediction goes wrong most of the times.

**1.4 Approved Objectives**

1. To predict the results of the cricket match by not only taking factors like toss, venue, day-night but also by considering factors like team composition, the batting and bowling averages of the each player in the team, the performance of the team in their past few matches, the probability of winning by batting first at a specific venue against a specific team.
2. To assist the cricket board of the team for selecting the players for the specific tournament.

**1.5 Methodology Used**

The project has been divided into sub tasks.

* Data Preprocessing
* Feature Generation
* Model Generation and its testing
* GUI Development for better user interaction.

We initiate our project by fetching data from sites like ESPN, Kaggle, and Cricksheet. The data we obtained is in the form of csv file where each file describes each match details. Then we generated python scripts to club the data entries of each and every match. We then applied feature selection to select features like teams involved, match venue, toss decision and winner of the match. We then used feature generation to create more important features like strength of each team, performance of each team in past few matches, probability of team batting first and winning probability of particular team at a specific location based on all the previous matches played between those two teams at the specific venue. The strength and performances of each match are calculated in terms of relative strength and relative performances. The strength of each team in a particular match is calculated based on the batting and bowling averages of each player in that match. The strength of each team is calculated by subtracting the batting average of the team with the bowling average. Further the team batting average is calculated by taking the mean of batting averages of all the players of the team and the bowling average is calculated by taking the mean

of bowling averages of all the players of the team. After getting the strength of each team we subtract the strength of team A with team B to get relative strength of overall match. If relative strength is positive it means team A is stronger than team B. If negative it means team B is stronger than team A. Else both are equally strong. Along with strength we have also considered the past few match performances of each team as it tell us whether the team is in good form or not. The performances are calculated by taking mean batting average of past ten matches of each team. After getting individual team performances we then calculate the relative performance of the team by subtracting team A performance with team B performance. If the relative performance is positive it means team A is in better form than team B. If negative then it means team B is in better form. Otherwise both are in equal form. The magnitude tells us the amount with which one team is better than other.

After preprocessing and feature generating we used the machine learning to generate models which can be used to predict the results. We used different models and calculated the accuracy from each model and finally selected the one with best accuracy. We used decision tree, random forest, logistic regression and SVM. Out of all these we used logistic regression as it gave us the better accuracy among all other models. While generating the models we used K Fold cross validation technique so as to make use of whole dataset as training as well as testing. After generating models we developed a GUI in python using tkinter module so that user can easily interact with our system rather than going to command prompt and typing all the fields.

**1.6 Project Outcomes/Deliverables**

* Machine learning based model which will predict the ODI match results in advance by taking into consideration the factors like team composition, players batting and bowling averages and their performances in past ten matches along with other parameters like toss decision and venue and home team advantage.
* Tkinter based GUI for making the user friendly interaction with the model.
* Bar chart of accuracy attained with different models.

**1.7 Novelty**

* The project builds the model by also considering team composition, players batting and bowling averages and their performances in past ten matches along with other parameters like toss decision and venue and home team advantage.
* The project can be easily used by layman because of the friendly intractable user interface.

**2. Requirement Analysis**

**2.1 Literature Survey**

According to [2] factors contributing to winning games are imperative, as the ultimate objective in a game is victory. The aim of this study was to identify the factors that characterize the game of cricket, and to investigate the factors that truly influence the result of a game using the data collected from the Champions Trophy cricket tournament. According to the results, this cricket tournament can be characterized using the factors of batting, bowling, and decision-making. Further investigation suggests that the rank of the team and the number of runs they score have the most significant influence on the result of games.

[4] embarks upon a very critical aspect that the team composition changes over time. It propose novel methods to model batsmen, bowlers and teams, using various career statistics and recent performances of the players. It propose a novel dynamic approach to reflect the changes in player combinations.

[5] proposes a model to predict the winner at the end of each over in the second innings of an IPL cricket match. Our methodology not only incorporates the dynamically updating game context as the game progresses, but also includes the relative strength between the two teams playing the match. Estimating the relative strength between two teams involves modeling the individual participating players’ potentials. To model a player, we use his career as well as recent performance statistics. Using the various dynamic features, we evaluate several supervised learning algorithms to predict the winner of the match. Finally, using the Random Forest Classifier (RFC), we have achieved an accuracy of 65.79% - 84.15% over the course of second innings, with an overall accuracy of 75.68%.

[6] focuses on the prediction of likelihood of India winning or losing in One Day International (ODI) cricket match against Australia by fitting the logistic regression model. According to ICC ODI championship rating, dated 7th August 2015, India holds 2nd position with 5875 points and 115 rating by playing 51 matches. Data from actual recent matches with five independent variables and one dependent binary logistic variable are used throughout to illustrate the implementation of this successful use of mathematical and statistical principles to the solution of a practical problem in one-day international cricket match.

In [8], a machine learning model has been developed that predicts match result on every ball played. Using Duckworth- Lewis formula match outcome will be predicted for live match. For every ball bowled a probability is calculated and probability figure is plotted. For betting industry this model and the probability figure will be very useful for bettor in deciding which team to on and how much to bet.

In [12] a model has been proposed that has two methods, first predicts the score of first innings not only on the basis of current run rate but also considers number of wickets fallen, venue of the match and batting team. The second method predicts the outcome of the match in the second innings considering the same attributes as of the former method along with the target given to the batting team. These two methods have been implemented using Linear Regression Classifier and Naive Bayes Classifier for first innings and second innings respectively. In both methods, 5 over intervals have been made from 50 overs of the match and at each interval above mentioned attributes have been recorded of all non-curtailed matches played between 2002 and 2014 of every team independently. It has been found in the results that error in Linear Regression classifier is less than Current Run Rate method in estimating the final score and also accuracy of Naive Bayes in predicting match outcome has been 68%.

[13] deals with the evaluation of the Duckworth Lewis method, identifying its limitation, and devising a modification to address these limitations. The Duckworth Lewis method, or D/L method, was created by Frank Duckworth and Tony Lewis. The International Cricket Council (ICC) adopted D/L method in 1999 to address the issue of delayed one-day cricket matches due to interruptions such as inclement weather conditions, poor light and floodlight failures, and crowd problems. They have attempted to identify the shortcomings in the existing Duckworth Lewis method using data mining algorithms such as Random Forests and C4.5. They have also shown that the p-values and other data mining techniques serve a dual purpose of not only evaluating whether systems such as D/L method have been exploited by taking advantage of their properties such as simplicity, but also devising alternate and robust approaches (or models). In the first part of their project, they have analyzed fifty one-day international (ODI) cricket matches, in which the Duckworth Lewis method has been applied, using tools such as WEKA and Microsoft Excel. They have observed that the Duckworth Lewis method has some limitations. As a result, using data mining methods they have shown that the Duckworth Lewis system has proven over time to be biased towards the team batting first and the team winning the toss -- a toss refers to the coin-flip at the beginning of the match used to decide who bats or fields. Bias in the context of the report is defined as taking advantage of the properties of systems such as the Duckworth Lewis method. They also showed that such an "exploitation" of the system permits prediction of the match winner with outcomes that are better than just chance. Using the analyses described above, they propose a modification to the existing Duckworth Lewis method to reduce this bias by considering the "venue" of the game as an additional resource along with the two existing resources-overs and wickets - to predict the target score. They have done a basic evaluation of the reduction in bias due to the proposed changes. The modification has helped not only to reduce the bias but also to alleviate the impact of factors such as toss and team batting first in predicting the target scores in limited-overs cricket matches.

[14] identifies rising stars in cricket domain by employing machine learning techniques. More precisely, it predict rising stars from batting as well as from bowling realms. For this intent, the concepts of co-players, team, and opposite teams are incorporated and distinct features along with their mathematical formulations are presented. For classification purpose, generative and discriminative machine learning algorithms are employed, and two models from each category are evaluated. As a proof of applicability, the proposed approach is validated experimentally while analyzing the impact of individual features. Besides, model and category wise assessment is also performed. Employing cross validation, it demonstrate high accuracy for rising star prediction that is both robust and statistically significant. Finally, ranking lists of top ten rising cricketers based on weighted average, performance evolution, and rising star scores are compared with the international cricket council rankings.

**2.2 Analysis/Working Principles**

The first step of our project involves taking input from the user. The inputs are teamA, teamB, city, toss winner, toss decision. The image below shows a GUI which was created using python module called Tkinter. These inputs will be the prediction parameters that will predict the outcome of the match.

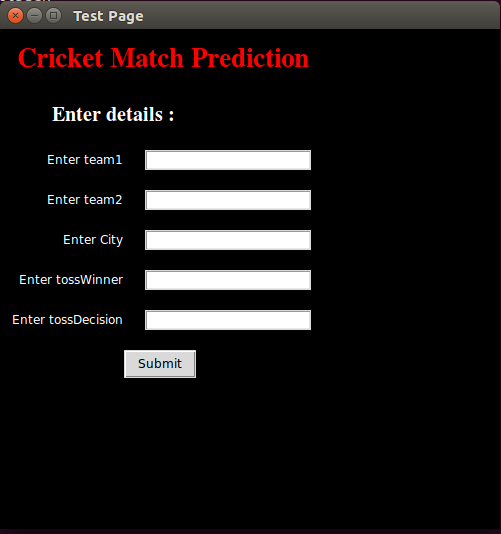


Figure 2.1 User Interface

The overall working of our project can be explained through the following abstract diagram

****

Figure 2.2 System Design

**Data Preprocessing-** The initial step is to perform the data preprocessing of the initial data. Basically our initial data consists of the following columns -

* Match ID
* Toss
* Toss Decision
* Team A
* Team B
* Date of the match
* Venue
* Winner

Of these data there are few missing entries which are solved by taking mean of all the values of that specific attribute. Then we apply feature generation techniques to add features which tells the relative strength and relative performance of team A. There are 4 features which are added.

**TeamA\_win\_prob –** This quantity is obtained by considering the 10 recent matches of the 2 teams in the original row and finding how many of these matches have team A won. After that, the matches won by team A is divided by the total 10 latest matches played by both the teams (probability of winning).

**Relative Performance–** The performance of a match is calculated by subtracting the average form of the individual teams over the last 10 recent matches. The quantity form basically represents the batting average of the whole match. The formula basically becomes

Relative Performance = Team A Batting Avg - Team B Batting Avg

**Strength** – It basically tells the winning probability based on team composition in that particular match. It takes into account the batting averages of all the 11 players and bowling averages of all the 11 players and then subtract batting average with bowling average to give the strength of that team. Then we subtract team A strength with team B strength to get relative strength of team A.

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

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

**Training the prediction model -** After out dataset in ready and input has been taken from the user, we apply a machine learning models and determine the outcome. The additional 4 features are also calculated for the input match in the same way we calculated for each row in the match during data preprocessing. Then we form a dictionary of all the inputs and finally convert that dictionary to pandas dataframe so that it can be processed by predict function. Now out of all the models we have used logistic regression because as you can see in the following diagram how it gives the best result among all the other classifiers. We pass the test dataframe to logistic regression model with parametric value of random state set to one. We then get the winner of the match as a single entity. We display the other parameters in command line interface while we pass the winner value to our front end file where it store its value in a label container and then display that container. The user can change parametric values and know the results without restarting the application.

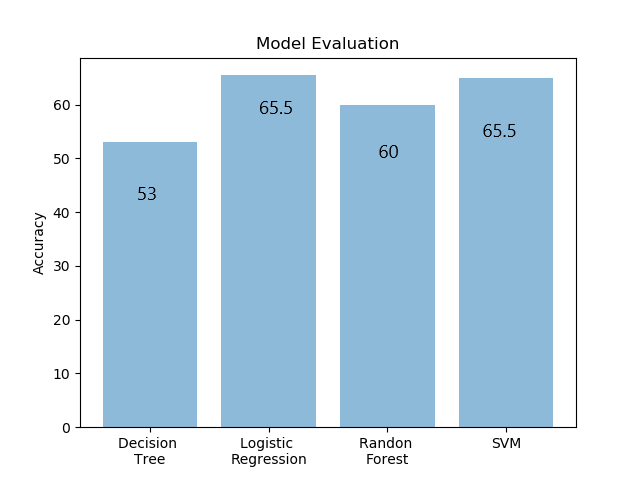
****

Figure 2.3 Model Evaluation

**2.3 SRS Details**

**2.3.1. Introduction**

**Purpose**

The purpose of this project is to provide software having version no. 1.0 that takes cricket match data as input in the form of team names, the toss decision, the toss winner, venue and output the predicted value as the winner of the match.

**Document Conventions**

The document follows a pre-specified Modern Language Association (MLA) format. Bold faced text has been used to emphasize section and sub-section headings. Every requirement statement written hereby is to have its own priority and no prior inheritance is assumed anywhere.

**Intended Audience and Reading Suggestions**

The document is intended for all the developers involved in the software and all the users who will be using the software, testers and documentation writers. The rest of the SRS specifies thefunctioning the match predictor software in detail.

Section 2 presents the overall description about the software, its functions and assumptions taken to make the software. Section 3 takes us through the system features. Section 4 tells us about the external interface requirements which are needed while section 5 and 6 guides us about other requirements which are needed.

**Project Scope**

The goal of the project is to predict the result of the match beforehand by taking into consideration the data and the results of the previous matches played between the two teams. The software is quite simple to use and can be used by any person.

**2.3.2 Overall Description**

**Product Perspective**

The perspective of the product is to predict the result of ODI matches with the help of machine learning technology. The prediction will be of who will win the match.

**Product Features**

The major features of the product are summarized as follows:

* User will be able to enter predicting match details through a GUI made in Tkinter module of python
* The product will predict which team is going to win.
* The product is based on supervised learning model. It will train previous data and will store the results in the database.
* It will take a dataset in the form of csv containing the history of all the ODI matches which include the teams’ names, toss decision, toss winner, venue.

**Operating Environment**

Operating System – Windows XP and above, Linux

Hardware Requirements – Pentium IV or more, RAM – 512 or more (Though core i3 or above is recommended if results are required fast)

Storage Requirements – A hard drive containing the dataset and the software code itself.

**Design and Implementation Constraints**

Some of the constraints that can limit the options available are:

* Accuracy: all the fields of the input form which is used to take predicting match details are required else prediction may go wrong.
* Performance: the hardware and software requirements mentioned in the operating environment should be present for proper functioning of the software.
* Language: English is kept as the only displayed language.

**2.3.3 System Features**

**Input Predicting Match Details**

**Description and Priority**

User will be able to enter match data a GUI interface built using Tkinter library of python.

**Stimulus/Response Sequences**

* User first interact with the Home screes
* Home page will have input form which requires match details which are team names, toss decision, toss winner, venue.
* Further processing will take place only after all the details have been entered else an error message will pop up.

**Functional Requirements**

REQ-1: Input Form.

REQ-2: A function that will check the completeness of the details entered by the user.

**Predicting the Result of the Match**

**Description and Priority**

The product will predict which team will win the game.

**Stimulus/Response Sequences**

* The data collected from the form is stored in a vector.
* Before predicting the match first the machine learning model is created based on previous data.
* Now the input details are sent to the predict function of our machine learning model class.
* According to that prediction will be done.

**Functional Requirements**

REQ-1: Optimized Machine Learning Algorithm.

REQ-2: Valid dataset with which testing has to be done.

**2.3.4. External Interface Requirements**

**User Interfaces**

Python based library called Tkinter is used for taking the input from the user which are (teams names, toss winner, toss decision, venue). Several exception are handled for creating this form, like if the user inputs incomplete information, or the user inputs some other field, the screen will display a message “Details are incomplete, please fill the complete details” and “The input of following field is invalid” respectively.

**Hardware Interfaces**

Display hardware device for user interface and displaying the result, mouse and a keyboard to interact with the tool are all required to be connected. Data will be maintained by a System administrator. The data will be in a csv format.

**Software Interfaces**

Tools used are SVM, python and Tkinter. Scikit is the library used.

**2.3.5. Other Nonfunctional Requirements**

**Performance Requirements**

To generate the result with optimized efficiency and process it faster, a fast processor will be required so as to train the machine learning model quickly.

**Safety Requirements**

The software is an attempt to predict the result with maximum accuracy possible. Since the result is based on the current player performances, how the teams will play and probabilities from the test and train data, so 100% accuracy is impossible to achieve. Since the possibility of winning a game turns with every moment, which is what this software aims to predict accurately, the results can sometimes be not as the true outcome. The software developing company hold no responsibility if any harm or loss occurs due to the use of software results.

**Security Requirements**

The System Administrator will be responsible for keeping the database secured for access by only certain individuals. Any malicious link, or advertisements will not be shown and if it does, users are requested to click them at their own risk as they are a part of user’s client app.

**Software Quality Attributes**

a) Availability: The software will be available as a package.

b) Maintainability: The software is easy to maintain and operate.

c) Testability: The software is easy to use and test as previous records can be interpreted and they can be backtracked and tested.

d) Correctness: The probabilities which are the output are accurate to certain level.

**2.3.6. Other Requirements**

All other requirement details have been specified separately in sub section 4. There are no additional requirements.

**2.4 Analysis Model**

**2.4.1 Activity diagram**

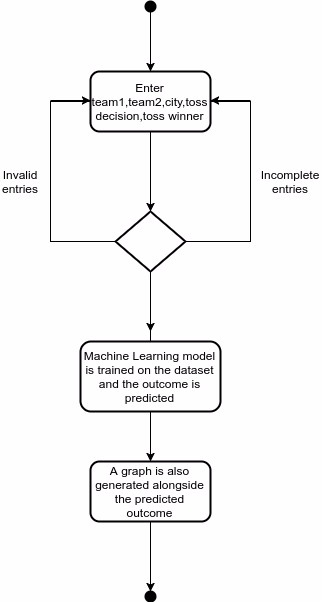


Figure 2.4 Activity Diagram

**2.4.2 Class Diagram**

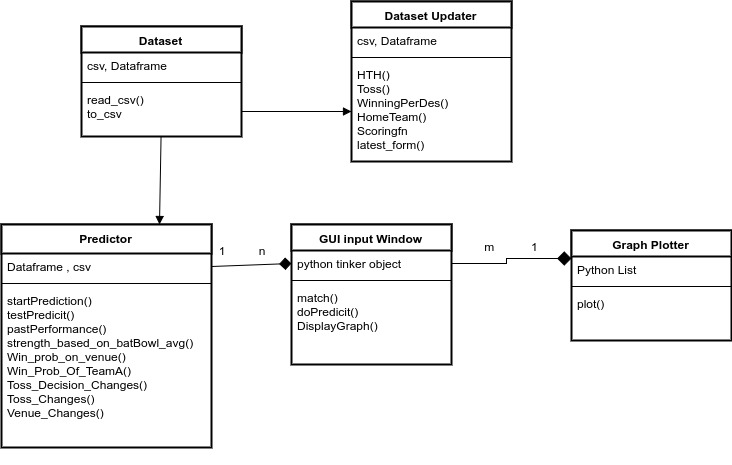


Figure 2.5 Class Diagram

**2.4.3 Data Flow Diagram (level 0)**

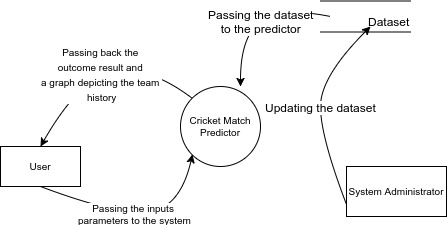


Figure 2.6 Data Flow Diagram (level-0).

**Data Flow Diagram (level 1)**

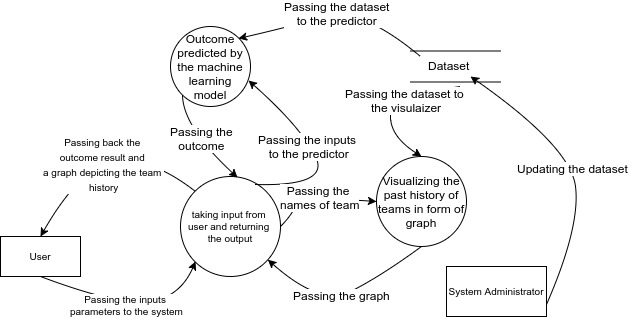


Figure 2.7 Data Flow Diagram (level-1).

**2.4.4 Sequence Diagram**

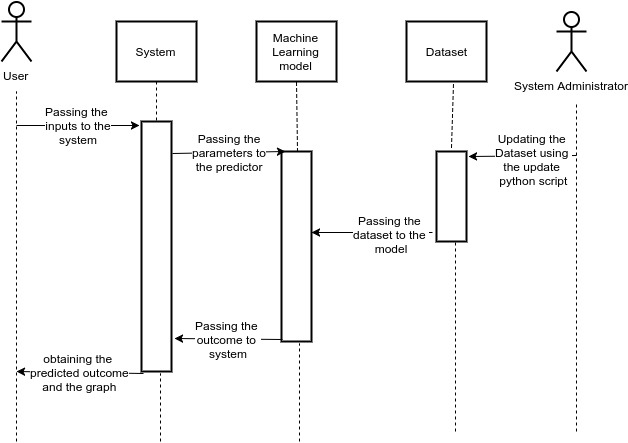


Figure 2.8 Sequence Diagram.

**2.4.5 Use case Diagram**

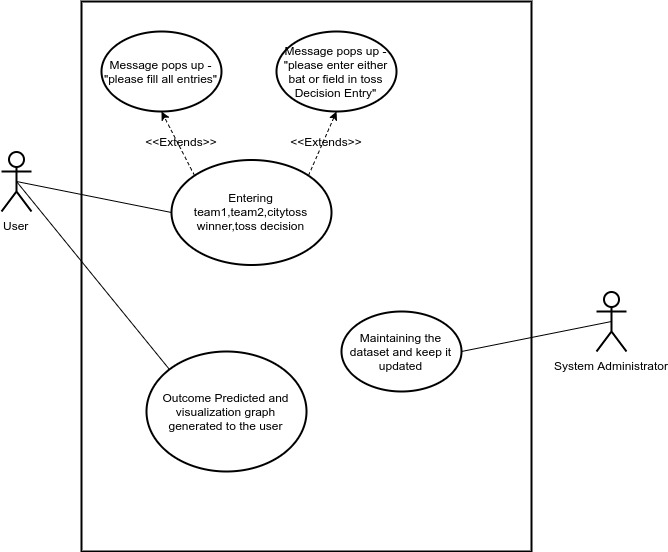


Figure 2.9 Use Case Diagram.

**2.5 Cost Analysis**

Apart from the laptop or the personal computer running the prediction software, there is no other hardware we have used in our project. All the python libraries used and python itself are freely available too on the internet.

For the choice of the laptop or any machine we recommend having a good processor (at least core i3). A processor below that may take much more time as in i3 itself, a neural network may take 30 minutes to train and 15 minutes for logistic regression.

**2.6 Assumptions and Constraints**

1. It is assumed that the python (3.0) is installed in the system alongside the following libraries -

* Pandas
* sci-kit learn
* Tkinter
* mathplotlib
* csv

2. There should be a System administrator whose job is to keep the dataset updated before a user runs the program.

3. The system should have sufficient processor speed so as to train the huge amount of dataset on different machine learning algorithms.

**3. Design Specifications**

**3.1 Flow Chart**

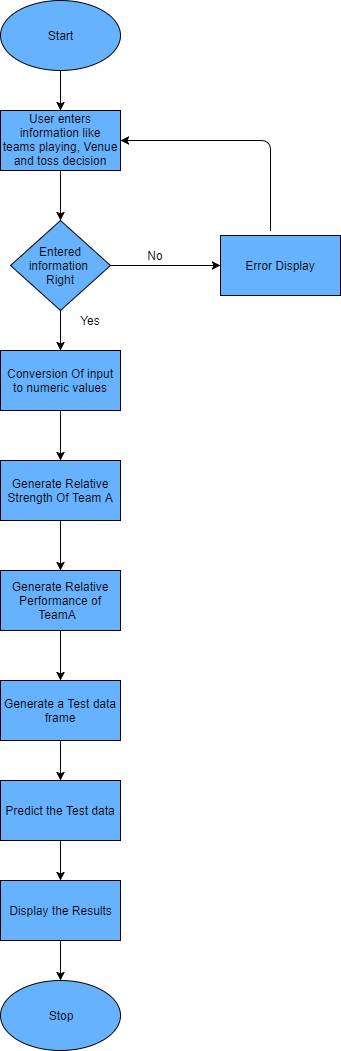


Figure 3.1 Flow Chart

**3.2 Design Phases**

The whole project is divided into 4 phases.

* Data Preprocessing
* Feature Generation
* Model Generation and its testing
* GUI Development for better user interaction.

**Data Preprocessing**

This stage involves fetching data from web and then processing it to make it suitable for modelling. The data we obtained is in the form of csv files where each file describes each match details. The details that single csv file constitutes are match id, team names, toss winner, toss decision taken after winning , venue of the match, date of the match, scores of each team after every over. This data also contains few missing entries. With the help of python we first parsed all the files and stored the data in a single file. While parsing we have fetched only required attributes from each file rather than storing whole contents of file. Now the resultant file each row corresponds to details of each match. After parsing we handled missing values by taking mean of that particular feature for which the value is missing.

**Feature Generation**

In this phase we created more important features like strength of each team, performance of each team in past few matches, probability of team batting first and winning probability of particular team at a specific location based on all the previous matches played between those two teams at the specific venue. The strength and performances of each match are calculated in terms of relative strength and relative performances. The strength of each team in a particular match is calculated based on the batting and bowling averages of each player in that match. The strength of each team is calculated by subtracting the batting average of the team with the bowling average. Further the team batting average is calculated by taking the mean of batting averages of all the players of the team and the bowling average is calculated by taking the mean of bowling averages of all the players of the team. After getting the strength of each team we subtract the strength of team A with team B to get relative strength of overall match. If relative strength is positive it means team A is stronger than team B. If negative it means team B is stronger than team A. Else both are equally strong. Along with strength we have also considered the past few match performances of each team as it tell us whether the team is in good form or not. The performances are calculated by taking mean batting average of past ten matches of each team. After getting individual team performances we then calculate the relative performance of the team by subtracting team A performance with team B performance. If the relative performance is positive it means team A is in better form than team B. If negative then it means team B is in better form. Otherwise both are in equal form. The magnitude tells us the amount with which one team is better than other.

**Model Generation and its testing**

After preprocessing and feature generating we used the machine learning to generate models which can be used to predict the results. We used different machine learning models and calculated the accuracy from each model and finally selected the one with best accuracy. We used decision tree, random forest, logistic regression and SVM. Out of all these we used logistic regression as it gave us the better accuracy among all other models. While generating the models we used K Fold cross validation technique so as to make use of whole dataset as training as well as testing. For future match prediction we first take the inputs from user via GUI which is created using tkinter. Then we first convert those inputs in numeric form like 1 for team A and 0 for team B and generate other important features using the modules we have used in phase 2. Then we convert these whole inputs into pandas data frame and pass it to model for prediction. The model does the processing and gives us the winner of the match.

**GUI Development for better user interaction**

In this phase we developed a simple user interface so that a user need not to go to command prompt for entering all the required inputs. The interface has been developed using python module tkinter. The interface contains a single form where user can enter some match details. The form developed also validates if the provided input is valid or not. If the entered information is invalid it returns an error message. Also if the user doesn’t give all the required inputs then it displays an error message to fill all the required entries. Thus it is easy and better way of getting all the valid inputs.

**3.3 User Interface Snapshots**

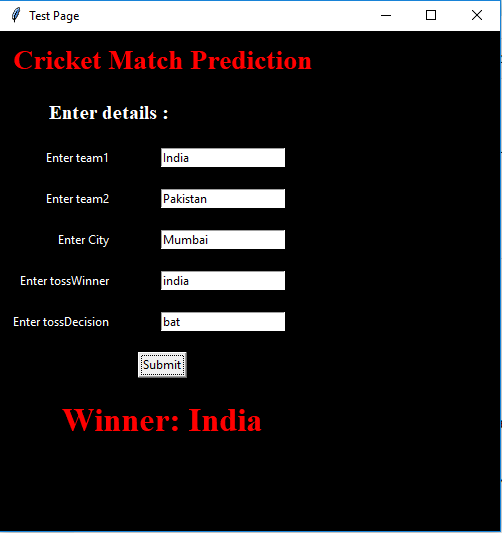


Figure 3.2 User Interface with Results.

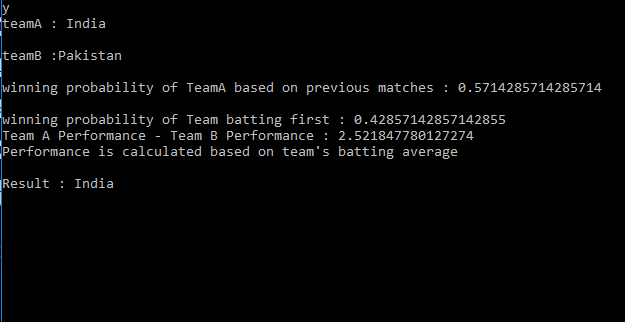


Figure 3.3 Command Line Values.

**3.4 Algorithms and Pseudo Codes**

**Pseudo Code for Relative Strength of Team A**

For index, row in iterrows(dataset):

teamA = row ['TeamA']

teamB = row ['TeamB']

matchId = row ['matchID']

df = pd.read\_csv("Dataset/PlayerInfo/"+matchId)

teamA\_list = df [(df.Country==teamA)]

teamB\_list = df [(df.Country==teamB)]

total\_A=0.0

total\_B=0.0

for index, row in teamA\_list.iterrows():

total\_A=total\_A+float(row['Bat\_Avg'])

batting\_avg\_A=total\_A/11

for index, row in teamB\_list.iterrows():

total\_B=total\_B+float(row['Bat\_Avg'])

batting\_avg\_B=total\_B/11

teamA\_list=teamA\_list.sort\_values(by = 'Wkts\_Taken', ascending=0)

top\_bowl\_A=teamA\_list.head(6)

teamB\_list=teamB\_list.sort\_values(by = 'Wkts\_Taken', ascending=0)

top\_bowl\_B=teamB\_list.head(6)

top\_A=0.0

top\_B=0.0

for index, row in top\_bowl\_A.iterrows():

top\_A=top\_A+float(row['Bowl\_Avg'])

bowling\_avg\_A=top\_A/6

for index, row in top\_bowl\_B.iterrows():

top\_B=top\_B+float(row['Bowl\_Avg'])

bowling\_avg\_B=top\_B/6

teamA\_strength = batting\_avg\_A-bowling\_avg\_A

teamB\_strength = batting\_avg\_B-bowling\_avg\_B

relative\_strength = teamA\_strength - teamB\_strength

**Pseudo Code for classifier**

predictors = ['Toss', 'Toss\_Decision', 'HTH', 'Venue', 'WinningPerDes','Strength','latest\_form']

alg = LogisticRegression(random\_state=1)

df = df1[predictors.append(‘Winner’)]

kf = KFold(df1.shape[0], random\_state=1)

predictions = []

for train, test in kf.split(df):

train\_predictors = (df[predictors].iloc[train,:])

train\_target = df["Winner"].iloc[train]

alg.fit(train\_predictors, train\_target)

test\_predictions = alg.predict(df[predictors].iloc[test,:])

predictions.append(test\_predictions)

predictions = np.concatenate(predictions, axis=0)

predictions = predictions.astype(int)

cnt = 0

for index, row in df.iterrows():

if predictions[index] == row["Winner"]:

cnt = cnt + 1

accuracy = cnt/len(predictions)

print accuracy

# 4. Results and Evaluations

**4.1 Testing Process**

* In order to calculate the accuracy of the model we have used K Fold algorithm with K value equal to 3. With this technique of cross validation we have used entire dataset for training as well as entire dataset for testing.

Test Results: 65% accuracy observed

* After that, real time testing is carried out by first taking inputs from the user via graphical user interface then generating the other features using the modules which were used to generate for actual dataset and then passing this newly created test data frame to predict function.

Test Results: Software is able to successfully predict the winner of the match.

**4.2 Inferences Drawn**

* The winning probability of a particular team greatly depends on its batting average, bowling average and its performance in past few matches.
* Logistic Regression algorithm gave the best accuracy among all the models.
* The only drawback with SVM in this case is the training time. It has provided the same accuracy as of logistic regression but took a large time in modelling.

**5. Conclusions and Future Directions**

**5.1 Conclusions**

With growing interest in the sport of cricket over the past few years, a need for tool which can predict the results of the match in advance has to be developed. Though there are lots of tools available in the market, there accuracy lacks in the way they take factors into consideration. To give an edge to those existing classifiers, our project aims to also take into consideration some important factors like team composition, performance of players in past few matches, batting and bowling averages of the players in each team and winning probability of team batting first at a specific venue against a specific opponent. All these important factors along with toss and venue has taken into account and a classifier has been generated to give better results. Along with these a user interface has also been generated so that even a layman can interact with our system with ease.

**5.2 Social Benefits**

* Cricket enthusiasts can know who has the higher chances of winning the match even before the match starts.
* Cricket Board can select specific players for the upcoming tournament by checking which team composition has higher chances of winning.
* It also assists team captains in deciding whether to choose bat or field at a specific stadium if they win the toss.

**5.3 Self Learnings**

* Learned various machine learning models like decision tree, random forest, logistic regression and SVM.
* Learned the approach used in solving machine learning problems.
* Learned tkinter module in python to develop GUI for desktop applications.
* Learned matplotlib library in python for data visualization.

**5.4 Future Direction**

The project currently takes into account the important factors from previous matches and then predicts the result before the match starts. But we can also take the ongoing match details to predict the result. So the future work will be to combine both the previous data with ongoing match data to have much more better results. Also the prediction can be extended to not only predict the winner of the match but also the expected runs to be scored by both teams.

**6. Project Metrics**

**6.1 Challenges Faced**

* In deciding the best parametric value for different classifiers while generating the model.
* The second was a minor problem of designing and coloring the GUI window
* Unknowingly Rearranging of index values while sorting the dataset on a particular attribute using pandas dataframe.
* Change of Kfold module from cross validation to model selection.

**6.2 Relevant Subjects**

Table 6.1 Subject Code and Subject name

|  |  |
| --- | --- |
| **Subject Code** | **Subject Name** |
| UTA003 | Computer Programming |
| UML501 | Machine Learning |
| UCS 503 | Software Enginnering |
| UCS406 | Data Structure and Algorithm |
| UMA001 | Mathematics - 1 |
| UCS633 | Data Analytics and Visualization |
| UCS305 | Programming Language Concepts |

**6.3 Interdisciplinary Knowledge Sharing**

In this project, we have used the principles of machine learning which include preprocessing the data, applying machine learning models to predict the outcome and data visualization using graph. The programming language used is python in which we have used Data structures like lists, dictionaries and Dataframe. The following python libraries have been put to use in project -

* scikit learn (Applying machine learning models)
* Tkinter ( making the GUI )
* mathplotlib ( For data visualization)
* Pandas (For data analysis)
* csv (for reading the dataset in form of csv)

Apart from that, principles of Software enginnering have also been applied to make various UML diagrams like DFD, sequence diagram, class diagram, activity diagram, use case diagram.

**6.4 Peer Assessment Matrix**

Table 6.2 Peer Assessment Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Evaluation of** | | |
|  | **Name of team members** | Mokshit Kumar | Jatin Singhal | Yatin Dang |
| **Evaluation by** | Mokshit Kumar | 4 | 4.5 | 4 |
| Jatin Singhal | 4.5 | 4 | 4.5 |
| Yatin Dang | 4 | 4.5 | 4 |

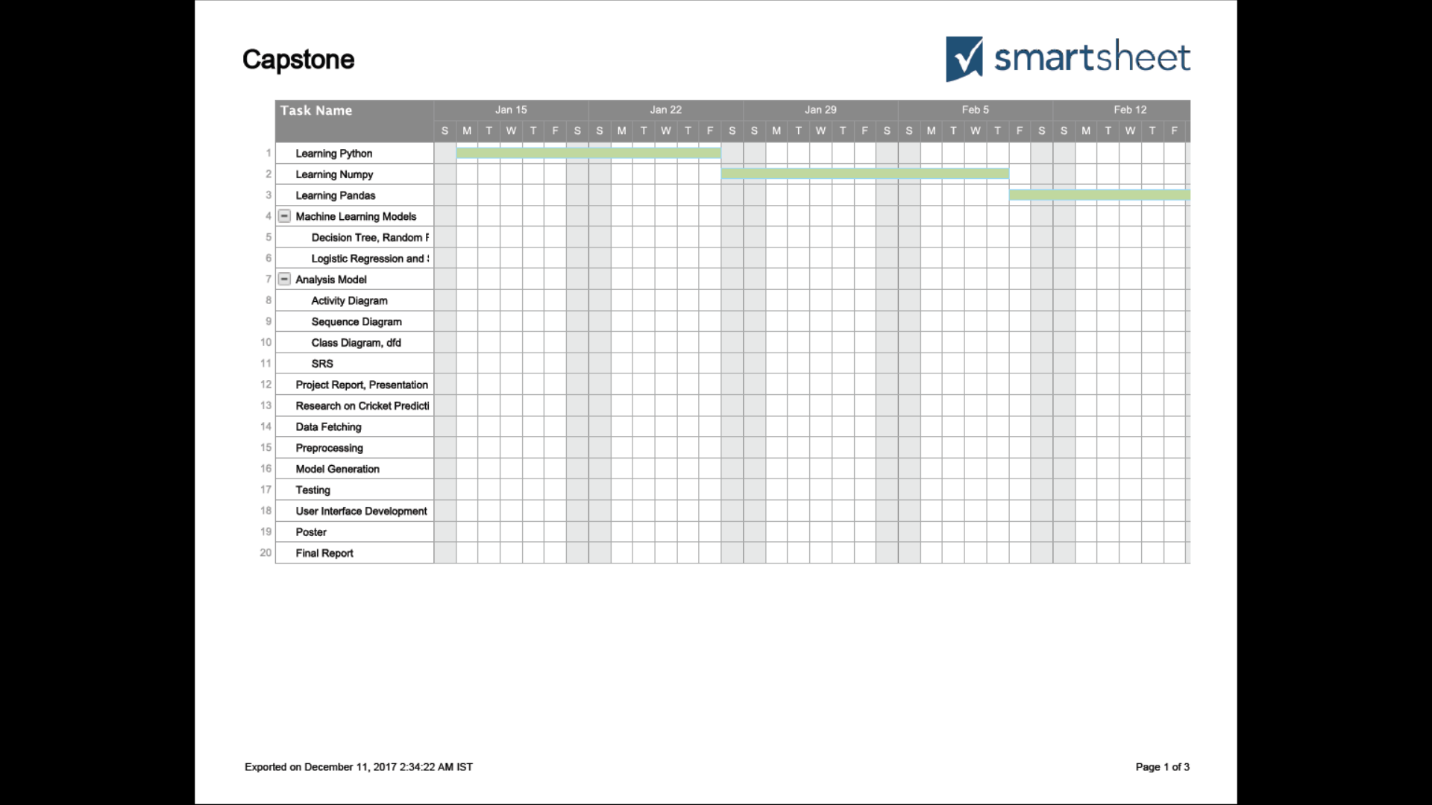
**6.5 Role Playing and Project Schedule**

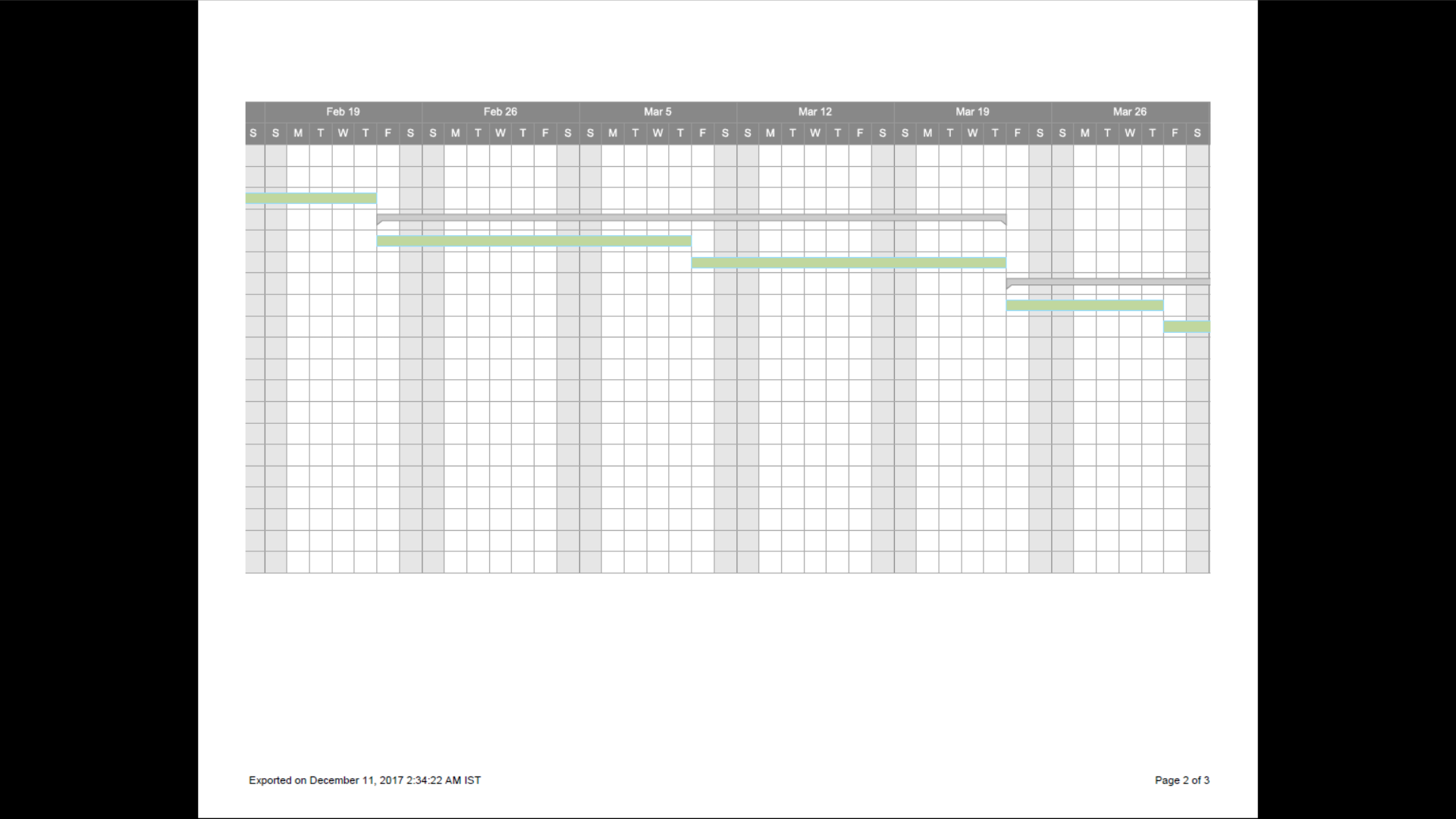
1. Jatin Singhal: Model Generation and user interface

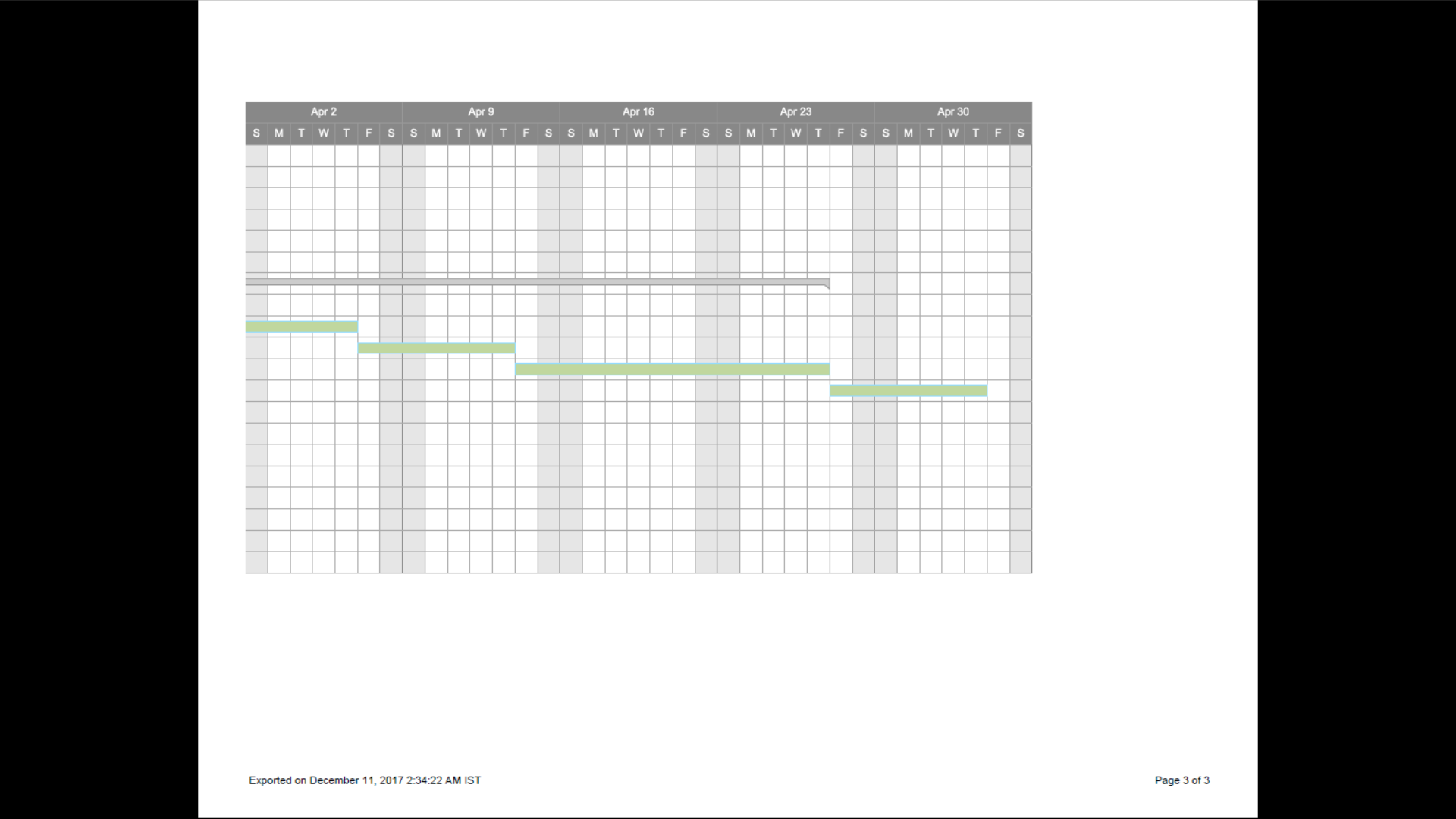
2. Mokshit Kumar: Data preprocessing and visualization

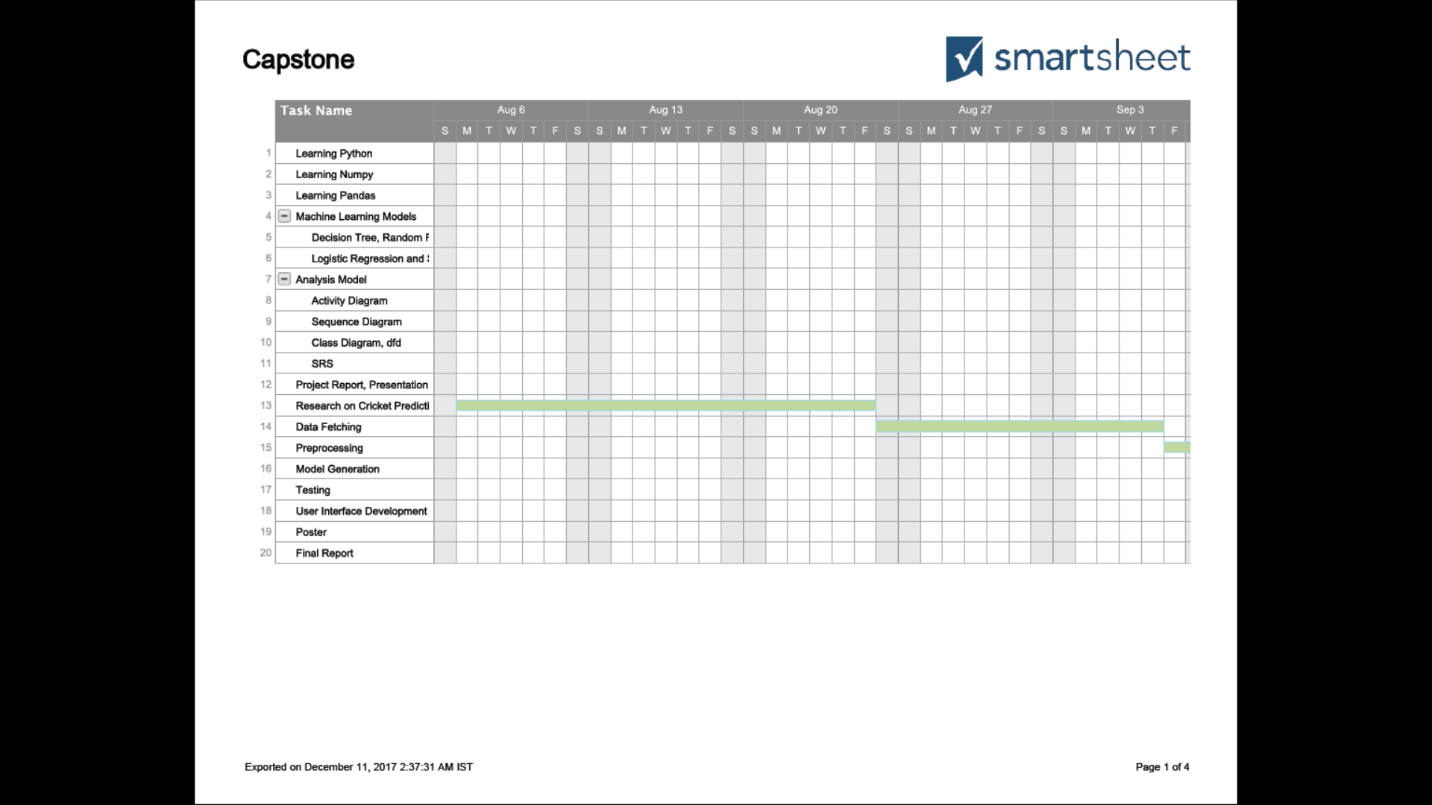
3. Yatin Dang: Data preprocessing

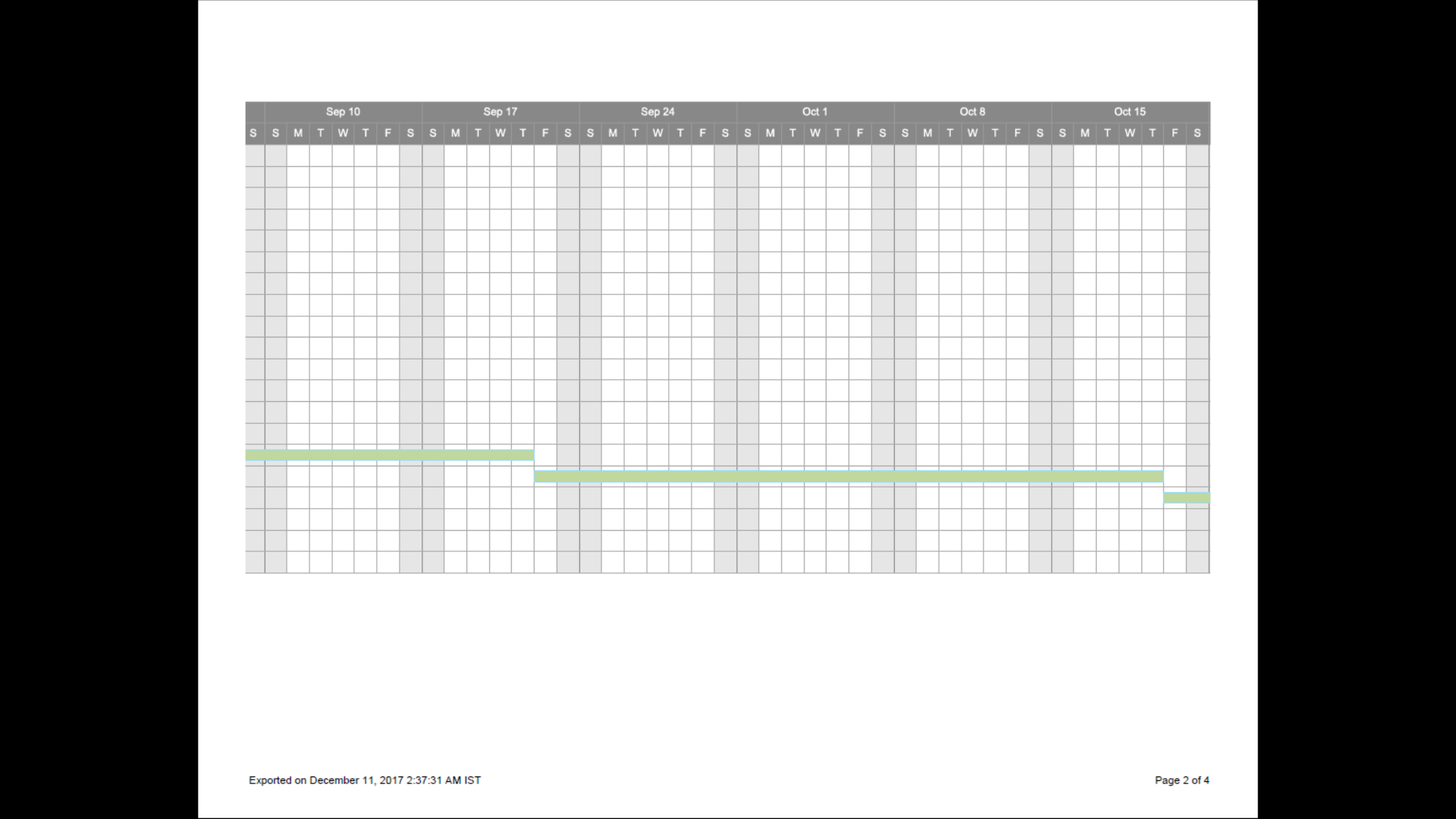
**6.6 Gantt Chart**

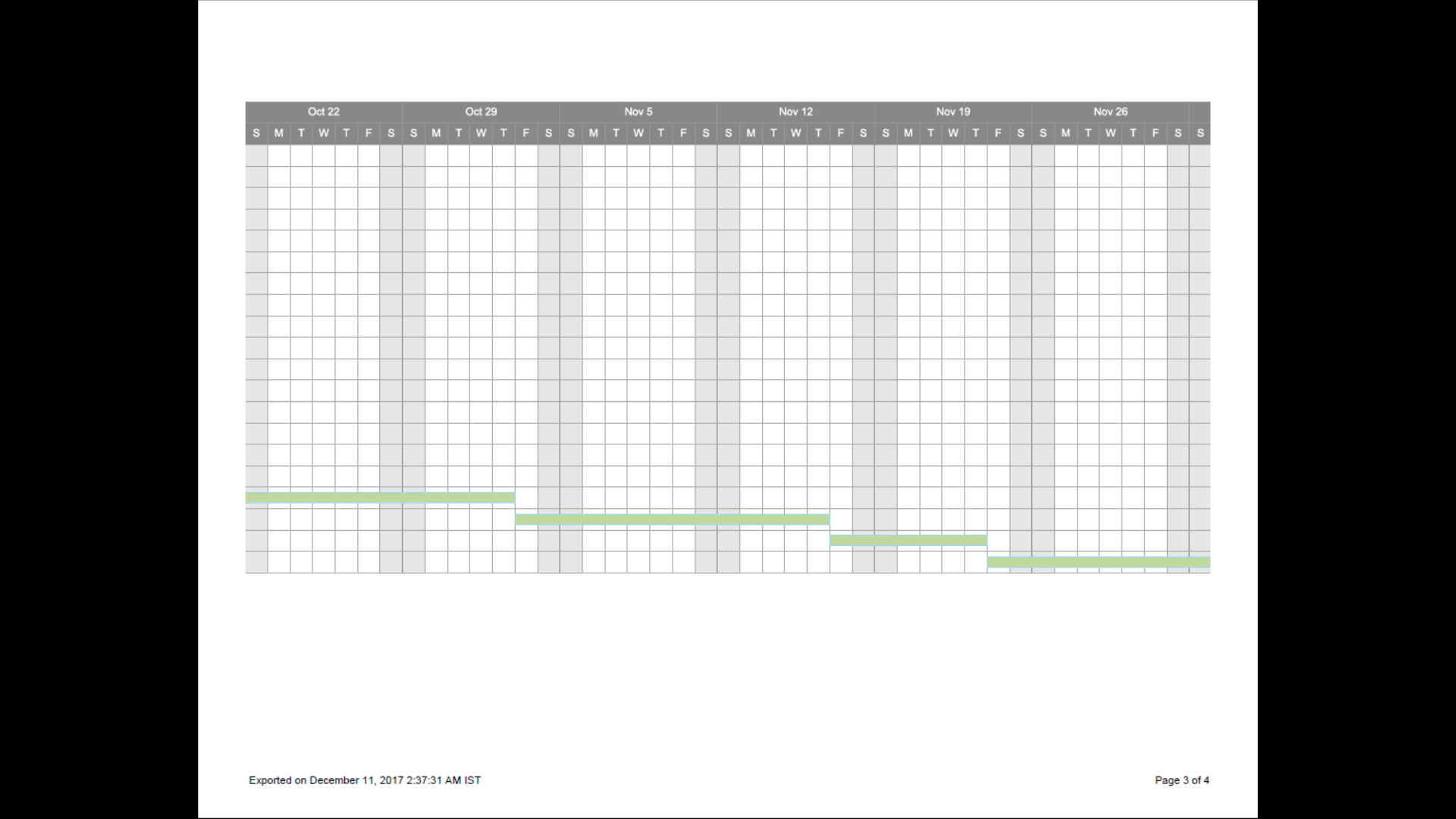


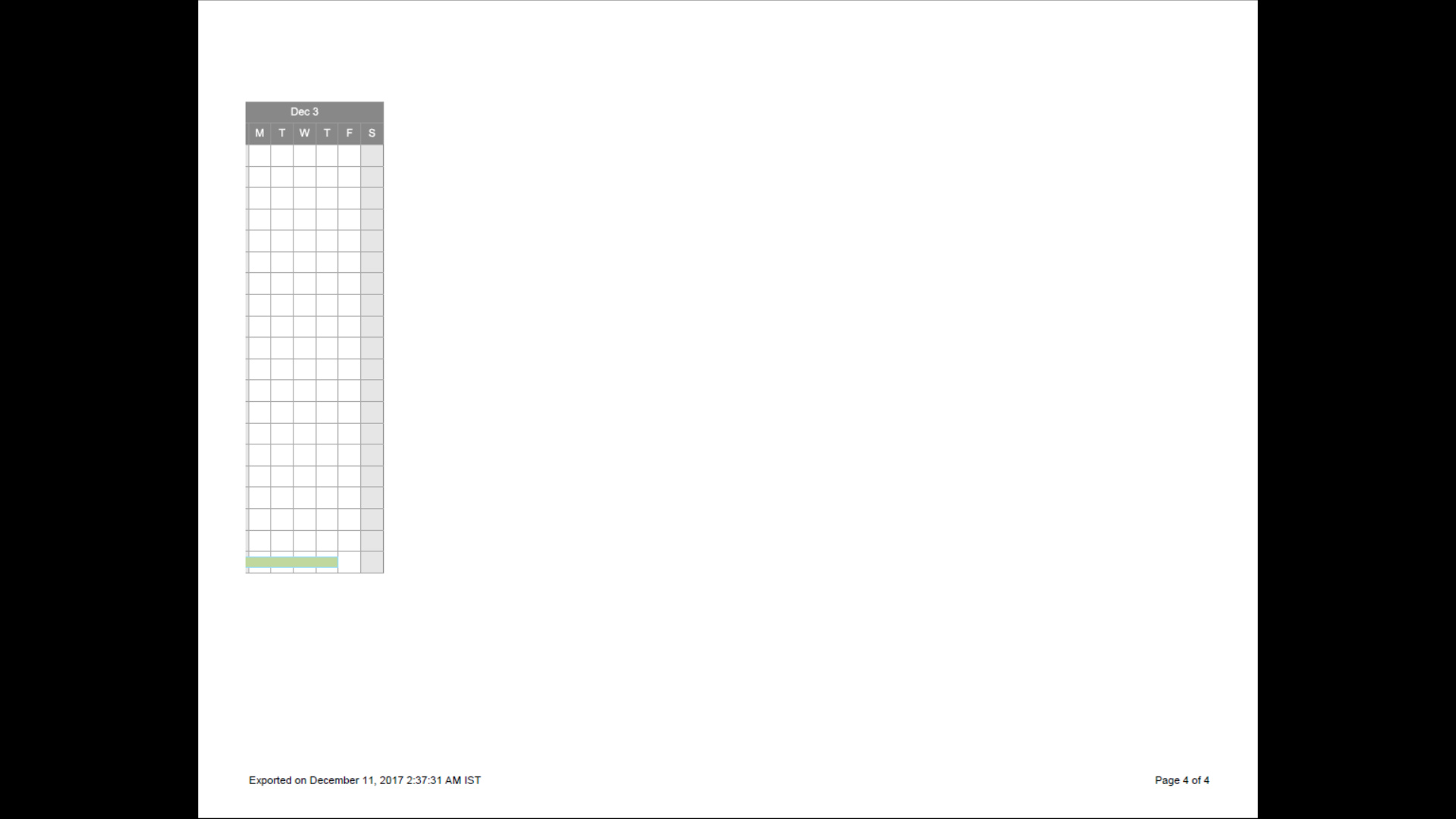












**6.7 Student Outcome (A-K)Mapping**

|  |  |  |
| --- | --- | --- |
| A1. | Applying basic principles of science  towards solving engineering problems | Used the principles of machine learning to analyze data and predict the outcome |
| B1. | Identify the constraints, assumptions and  models for the problems | Python environment installed with required libraries. The machine should have sufficient processing power. |
| C1. | Design software system to address desired  needs in different problem domains | The project addressed a better prediction model that took more factors into consideration as compared to the existing models |
| D1. | Fulfill assigned responsibility in  multidisciplinary teams | Each member of the team was assigned his task which was performed on time. |
| E2. | Develop appropriate models to formulate  solutions. | The machine learning model has been used for doing the match prediction |
| E3. | Use analytical and computational methods to obtain solutions. | Using the prediction model we have predicted the outcome of the match |
| G1. | Produce a variety of documents such as  laboratory or project reports using appropriate formats. | Our team has submitted number of documents like SRS(Software Requirement Specification), UML diagrams, Project Diary. |
| H1. | Examine economic tradeoffs in computing  systems | The project tries to optimize the computer resources like memory and CPU and corresponding costs |
| I1. | Able to explore and utilize resources to enhance self-learning. | Python language has been extensively used. |
| J2. | Comprehend the importance of  contemporary issues. | Prediction accuracy has been addressed by this project |
| K1. | Write code in different programming languages | Python has been used |
| K3. | Use Software tools necessary for computer engineering domain | Python, a csv file reader etc. |

**6.8 Brief Analytical Assessment**

**1. What sources of information did your group explore to arrive at the list of possible Project Problems?**

Answer: We studied online articles related to machine learning to get basic theoretical knowledge. Then we came across several projects related to prediction on MOOC and read technical journals and IEEE which have been mentioned in the reference.

**2. What analytical, computational and/or experimental methods did your project group use to obtain solutions to the problems in the project?**

Answer: As explained previously, we used a python csv reader and pandas dataframes to preprocess the data. For predicting the match outcome we used several machine learning models like decision tree, random forest, logistic regression and SVM.

**3. Did the project demand demonstration of knowledge of fundamentals, scientific and/or**

**Engineering principles? If yes, how did you apply?**

Answer: Fundamentals of Machine Learning, python language and principles of Data structures and algorithm and Software Enginnering were used.

**4. How did your group shares responsibility and communicate the information of schedule with others in team to coordinate design and manufacturing dependencies?**

Answer: We would meet regularly at fixed time to discuss the progress, future planning and divide different phases of project between ourselves. We would stick to a strict deadline.

**5. What resources did you use to learn new materials not taught in class for the course of the project?**

Answer: The basic data structures of python, libraries of python like csv, pandas etc. were not taught in the curriculum. For these we consulted the official online documentation.

**6. Does the project make you appreciate the need to solve problems in real life using engineering and could the project development make you proficient with software development tools and environments?**

Answer: The project helped me get a general idea of how a software is developed at a small scale, helped me cooperate with many team members and also helped learn a lot of software tools. All these benefits would definitely help us in the industry sector after college.

**7. References**

[1] Ananda Bandulasiri Manage (2015). Predicting the Winner in One Day International Cricket, Sam Houston State University.

[2] A Kaluarachchi, SV Aparna (2010).A classification based tool to predict the outcome in cricket.

[3] Bandulasiri, A. (2008). Predicting the winner in one day international cricket. Journal of Mathematical Sciences & Mathematics Education, 3 (1), 6-17.

[4] Madan Gopal Jhawar, Vikram Pudi (2016). Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach. Report No: IIIT/TR/2016/-1.

# [5] Sasank Viswanadha, Kaustubh Sivalenka, Madan Gopal Jhawar,Vikram Pudi(2006). Dynamic Winner Prediction in Twenty20 Cricket: Based on Relative Team Strengths.

# [6] Monalisha Pattnaik, Anima Bag (2015). Fitting of Logistic Regression Model for Prediction of Likelihood of India Winning or Losing in Cricket Match.

[7]. Munir, Fahad, Hasan, Md. Kamrul, Ahmed, Sakib, Md. Quraish, Sultan (2015). Predicting a T20 cricket match result while the match is in progress.

# [8] Parag Shah (2017). Predicting Outcome of Live Cricket Match Using Duckworth- Lewis Par Score.

### [9]Parag Shah, Mitesh Shah (2015).Predicting ODI Cricket Result.

[10] Stephan Grayand Tuan Anh Le (2002). How to fix a one day international cricket match.

[11] Swetha, Saravanan.KN (2017). Analysis on Attributes Deciding Cricket Winning.

[12] Tejinder Singh, Vishal Singla, Parteek Bhatia (2015).Score and Winning Prediction in Cricket through Data Mining.

# [13] Viraj Phanse, Sourabh Deorah (2011). Evaluation and Extension to the Duckworth Lewis Method: A Dual Application of Data Mining Techniques.

# [14] Haseeb Ahmed, Licheng Wang (2017). Prediction of Rising Stars in the Game of Cricket.