VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum – 590 018

A project report on

"Predicting Results of IPL Matches using Machine Learning"

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

by

M. RITHU (1CR16IS049)
SACHIN BASNET THAPA (1CR16IS087)

Under the guidance of

Mr. Prasad B.S
Assistant Professor
Dept. of ISE, CMRIT, Bengaluru

CMR INSTITUTE OF TECHNOLOGY
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
#132, AECS Layout, IT Park Road, Bengaluru-560037

2019-2020

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum – 590 018

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Certificate

This is to certify that the project entitled, **"Predicting Results of IPL Matches using Machine Learning"**, is a bonafide work carried out by **M. RITHU(1CR16IS049) and SACHIN BASNET THAPA(1CR16IS087)** in partial fulfillment of the award of the degree of Bachelor of Engineering in Information Science & Engineering of Visvesvaraya Technological University, Belgaum, during the year 2019-20. It is certified that all corrections/suggestions indicated during reviews have been incorporated in the report. The project report satisfies the academic requirements in respect of the Phase I project work prescribed for the said Degree.

Name & Signature of Guide	Name & Signature of HOD
Mr. Prasad B.S Asst. Professor Dept. of ISE, CMRIT	Dr. M. Farida Begam Professor,HoD Dept. of ISE, CMRIT

External Viva

Name of the Examiners	Signature with date
1.	
2.	

CMR INSTITUTE OF TECHNOLOGY
BANGALORE-560037

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Declaration

We, **M. RITHU(1CR16IS049) and SACHIN BASNET THAPA(1CR16IS087),** bonafide students of CMR Institute of Technology, Bangalore, hereby declare that the dissertation entitled, **"Predicting Results of IPL Matches using Machine Learning"** has been carried out by us under the guidance of Mr. Prasad B S, Associate Professor, CMRIT, Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Information Science Engineering, of the Visvesvaraya Technological University, Belgaum during the academic year 2019-2020. The work done in this dissertation report is original and it has not been submitted for any other degree in any university.

M. Rithu(1CR16IS049)
Sachin Basnet Thapa(1CR16IS087)

ABSTRACT

In today's date data analysis is need for every data analytics to examine the sets of data to extract the useful information from it and to draw conclusion according to the information. Data analytics techniques and algorithms are more used by the commercial industries which enables them to take precise business decisions. It is also used by the analysts and the experts to authenticate or negate experimental layouts, assumptions and conclusions. In recent years the analytics is being used in the field of sports to predict and draw various insights. Due to the involvement of money, team spirit, city loyalty and a massive fan following, the outcome of matches is very important for all stake holders. In this paper, the past seven year's data of IPL containing the player's details, match venue details, teams, ball to ball details, is taken and analyzed to draw various conclusions which help in the improvement of a player's performance. Various other features like how the venue or toss decision has influenced the winning of the match in last seven years are also predicted. Various machine learning and data extraction models are considered for prediction are Linear regression, Decision tree, K-means, Logistic Regression etc. The cross validation score and the accuracy are also calculated using various machine learning algorithms. Before prediction we have to explore and visualize the data because data exploration and visualization is an important stage of predictive modeling.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So, it is with gratitude that we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We would like to thank **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for providing an excellent academic environment in the college and his never-ending support for the B.E program.

We would like to express our gratitude towards **Dr. Farida Begam,** Assoc. Professor and HOD, Department of Information Science and Engineering CMRIT, Bangalore, who provided guidance and gave valuable suggestions regarding the project.

We consider it a privilege and honour to express our sincere gratitude to our internal guide **Mr. Prasad B.S,** Assoc. Professor, Department of Information Science & Engineering for their valuable guidance throughout the tenure of this project work.

We would also like to thank all the faculty members who have always been very Co-operative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during our project.

M.Rithu
Sachin Basnet Thapa

TABLE OF CONTENTS

TITLE	PAGE NO.
Abstract	**i**
Acknowledgement	**ii**
Chapter 1	**1-2**
PREAMBLE	**1**
1.1 Introduction	1
1.2 Plan of Implementation	2
1.3 Problem Statement	2
1.4 Objective of the Program	2
Chapter 2	**3-4**
LITERATURE SURVEY	**3**
Chapter 3	**5-7**
APPROACH AND DESIGN	**5**
3.1 Data Collection	5
3.2 Data Processing	5
3.3 Data Visualization	6
3.4 Model Development and Evaluation	7
Chapter 4	**8-10**
SYSTEM REQUIREMENTS SPECIFICATION	**8**
4.1 Functional Requirements	8
4.2 Non- Functional Requirements	9

4.3 System Configuration	10
4.4 Hardware Requirements	10
4.5 Software Requirements	10
Chapter 5	**11-12**
SYSTEM DESIGN	**11**
5.1 System Development Methodology	11
Chapter 6	**13-18**
IMPLEMENTATION	**13**
Chapter 7	**19-24**
RESULTS	**19**
Chapter 8	**25-25**
FUTURE SCOPE AND CONCLUSION	**25**
REFERENCES	**26-26**

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
Fig-1	Process of predicting IPL Team	5
Fig-2	Relation between match winning and toss winning	6
Fig-3	System Architecture	12

Chapter 1

PREAMBLE

1.1 Introduction

Machine Learning is a branch of Artificial Intelligence that aims at solving real-life engineering problems. This technique requires no programming, whereas it depends on only data learning where the machine learns from pre-existing data and predicts the result accordingly. Machine Learning methods have benefit of using decision trees, heuristic learning, knowledge acquisition, and mathematical models. It thus provides controllability, observability, stability and effectiveness.

Cricket is being played in many countries around the world. There are a lot of domestic and international cricket tournaments being held in many countries. The cricket game has various forms such as Test Matches, Twenty20 Internationals, Internationals one day, etc. IPL is also one of them, and has great popularity among them. It's a twenty-20 cricket game league played to inspire young and talented players in India. The league was conducted annually in March, April or May and has a huge fan base among India. There are eight teams which represent eight cities which are chosen from an auction. These teams compete against each other for the trophy. The whole match depends on the luck for the team, player's performance and lot more parameters that will be taken in to the consideration. The match that is played before the day is also will make a change in the prediction. The stakeholders are much more benefited due to the huge popularity and the huge presence of people at the venue. The accuracy of a data depends on the size of the data we take for analysing and the records that are taken for predicting the outcome.

Cricket is a game played between two teams comprising of 11 players in each team. The result is either a win, loss or a tie. However, sometimes due to bad weather conditions the game is also washed out as Cricket is a game which cannot be played in rain. Moreover, this game is also extremely unpredictable because at every stage of the game the momentum shifts to one of the teams between the two. A lot of times the result gets decided on the last ball of the match where the game gets really close. Considering all these unpredictable scenarios of this unpredictable game, there is a huge interest among the spectators to do some prediction either at the start of the game or during the game. Many spectators also play betting games to win money.

1.2 Plan of Implementation

The project can be broken down into 7 main steps which are as follows:

1. Understand the dataset.
2. Clean the data.
3. Analyse the candidate columns to be Features.
4. Process the features as required by the model/algorithm.
5. Train the model/algorithm on training data.
6. Test the model/algorithm on testing data.
7. Tune the model/algorithm for higher accuracy.

1.3 Problem Statement

To predict the results of an IPL match using machine learning techniques or algorithms such as Logistic Regression, Gaussian Naive Bayes, K Nearest Neighbours, SVM, Gradient boost algorithm, Decision tree and Random forest.

We have used 17 features which are as follows: season, city, date, team1, team2, toss_winner, toss_decision, result, dl_applied, winner, win_by_runs, win_by_wickets, player_of_match, venue, umpire1, umpire2 and umpire3.

1.4 Objective of the Project

The main objective of this project is to give the team players information about how each venue makes a difference to the game. And give feedback of how the players can improve their own performance in each game. And also give have a better planning of how the match should be played overall by the whole team regardless of the toss decision.

Chapter 2

LITERATURE SURVEY

In order to get required knowledge about various concepts related to the present application, existing literature were studied. Some of the important conclusions were made through those are listed below.

1. **Kalpdrum Passi and Niravkumar Pandey** discussed about the prediction accuracy in terms of runs scored by batsman and the no. of wickets taken by the bowler in each team [1].

2. **P. Wickramasinghe** proposed a methodology to predict the performance of batsman for the previous five years using hierarchial linear model [2].

3. **R.P.Schumaker** et. al, discussed about different statistical simulations used in predictive modeling for different sports [3].

4. **John McCullagh** implemented neural networks and datamining techniques to identify the talent and also for the selection of players based on the talent in Australian Football League[4].

5. **Bunker et. al**, proposed a novel sport prediction framework to solve specific challenges and predict sports results [5].

6. **Ramon Diaz-Uriarte et. al**, investigated the use of random forest for classification of microarray data and proposed a new method of gene selection in classification problem based on random forest [6].

7. **Rabindra Lamsal and Ayesha Choudhary**, proposed a solution to calculate the weightage of a team based on the player's past performance of IPL using linear regression [7].

8. **Akhil Nimmagadda et. Al**, proposed a model using Multiple Variable Linear Regression and Logistic regression to predict the score in different innings and also the winner of the match using Random Forest algorithm [8].

9. **Ujwal U J et. Al**, predicted the outcome of the given cricket match by analyzing previous cricket matches using Google Prediction API [9].

10. **Rameshwari Lokhande and P.M.Chawan** came up with live cricket score predicton using linear regression and Naïve Bayes classifier [10].

11. **Abhishek Naik et. Al**, proposed a new model used matrix factorization technique to analyze and predict the winner in ODI cricket match [11].

12. **Esha Goel and Er. Abhilasha** discussed the improvements in Random Forest Algorithmand described the usage in various fields like agriculture, astronomy, medicine, etc. [12].

13. **Amit Dhurandhar and Alin Dobra** proposed a new methodology for analysing the error of classifiers and model selection measures to analyse the decision tree algorithm [13].

14. **H. Yusuff et. Al**, performed logistic regression using mammograms to find the accuracy with valid samples [14].

Chapter 3

APPROACH AND DESIGN

The below figure explains the approach we have taken into building the predictive model using machine learning algorithms.

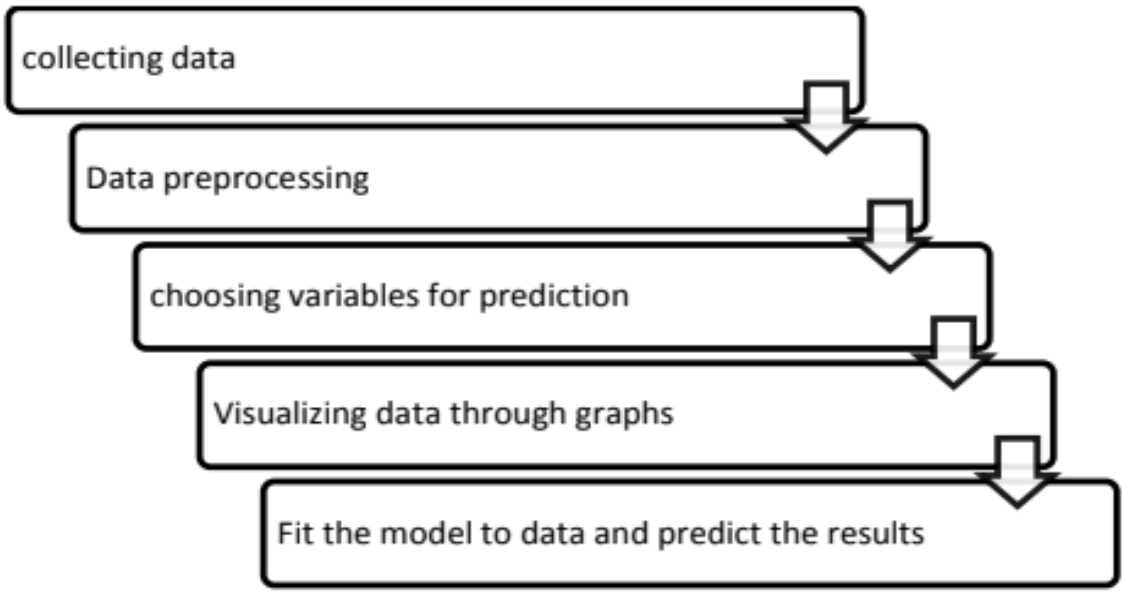


Fig 1: process of predicting IPL team

3.1 Data Collection

Data collection is the process of gathering and measuring information from countless different sources. In order to use the data, we collect to develop practical machine learning solutions.

Collecting data allows you to capture a record of past events so that we can use data analysis to find recurring patterns. From those patterns, you build predictive models using machine learning algorithms that look for trends and predict future changes.

The Indian Premier League's official website is the principal basis of data for this project. The data was web scrapped from the website and kept in the appropriate format using a python library called beautiful soup. The dataset has the columns regarding match-number, IPL season year, the place where match has been held and the stadium name, the match winner details, participating

teams, the margin of winning and the umpire details, player of the match etc. Indian Premier League was only 11 years old, which is why, after the pre-processing, only 577 matches were available. Here, some of the columns may contain null values and some of the attributes may not be required for match winner prediction which is discussed in data preprocessing.

3.2 Data Preprocessing

3.2.1 Data cleaning

There are some null values in the dataset in the columns such as winner, city, venue etc. Due to the presence of these null values, the classification cannot be done accurately. So, we tried to replace the null values in different columns with dummy values.

3.2.2 Choosing Required Attributes

This step is the main part where we can eliminate some columns of the dataset that are not useful for the estimation of match winning team. This is estimated using feature importance. The considered attributes have the following feature importance.

3.3 Data Visualization

- The data which has been collected is used for visualizing for the better understanding of the information.

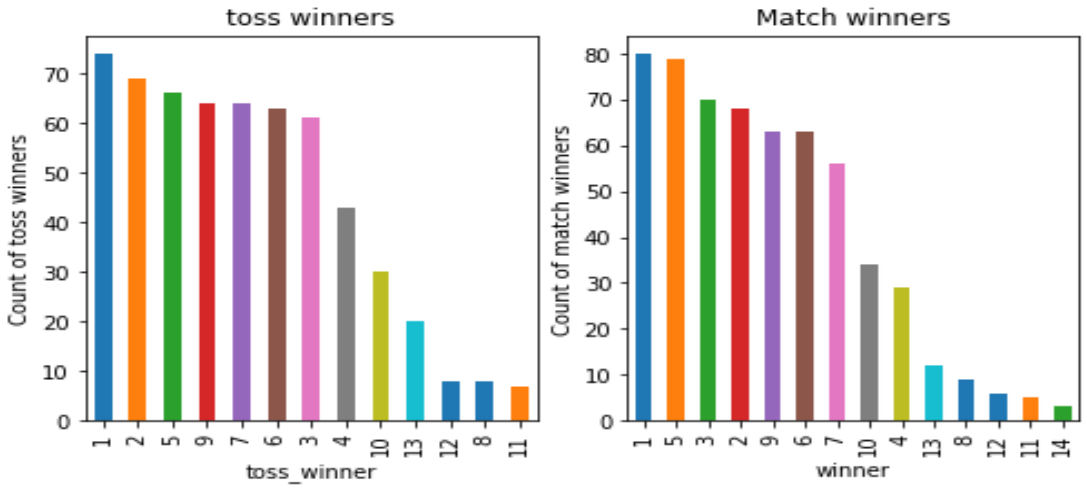


Fig 2 - Relation between toss winning and match winning

- Matplotlib Library is used here for visualizing the graphs
- The data visualization is necessary to understand the solution in a better way. The below graphs were drawn based up on the previous seasons of the IPL matches.

3.4 Model Development and Evaluation

Here, we have developed a generic model and applied all classification methods. The data is split into training data and test data, we train the model using certain features and use it to predict the testing data, then we calculate the performance of the system. The various classification models used are: Logistic Regression, Gaussian Naïve Bayes Classifier, KNN (K Nearest Neighbor) algorithm, Support Vector Machines, Gradient Boost Algorithm, Decision Trees and Random Forest Classifier. Among these methods the Random Forest and Decision tree has given good results.

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

A System Requirement Specification (SRS) is basically an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives the brief description of the services that the system should provide and also the constraints under which, the system should operate. Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will do it. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an ecommerce website and so on) must provide, as well as states any required constraints by which the system must abide. SRS also functions as a blueprint for completing a project with as little cost growth as possible. SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

4.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality.

Following are the functional requirements on the system:

1. The whole process can be handled at minimal human interaction with android and web both.
2. The application automatically receives the captured data from server.
3. The user can call emergency, map location and ECG graph on demand
4. The system gives a warning message.

4.2 Non Functional Requirement

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as :-

- Performance Requirements
- Design Requirements
- Security Constraints
- Basic Operational Requirements

4.2.1 Product Requirements

- **Platform independency:** A progressive web app will be developed and deployed so that users with a smartphone or a computer can access the voting site to cast their vote.
- **Ease of use:** The progressive web app provides an interface which is easy to use and eliminates the need for the voter to go to a voting booth.
- **Modularity:** The complete product is broken up into modules and well-defined interfaces are developed to explore the benefit of flexibility of the product.
- **Robustness:** This software is being developed in such a way that the overall performance is optimized, and the user can expect the results within a limited time with utmost relevancy and correctness.

4.3 System Configuration

4.3.1 H/W System Configuration:

- Processor - Pentium – IV
- Speed - 1.1 GHz
- RAM - 256 MB (min)
- Hard Disk - 20 GB

4.3.2 S/W System Configuration:

- Operating System - XP/7/8/8.1/10
- Coding Language - Python

4.4 Hardware Requirements

- Processors - Pentium IV Processor
- Speed - 3.00 GHZ
- RAM - 2 GB
- Storage - 20 GB

4.5 Software Requirements

- Operating system - Windows 10 Professional
- IDE used - Visual Studio Code

Chapter 5

SYSTEM DESIGN

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customer's requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

5.1 System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

5.1.1 Model phases

The waterfall model is a successive programming improvement process, in which advance is seen as streaming relentlessly downwards (like a waterfall) through the periods of Requirement start, Analysis, Design, Implementation, Testing and upkeep.

Prerequisite Analysis: This stage is worried about gathering of necessity of the framework. This procedure includes producing record and necessity survey.

Framework Design: Keeping the prerequisites at the top of the priority list the framework details are made an interpretation of into a product representation. In this stage the fashioner underlines on calculation, information structure, programming design and so on.

Coding: In this stage developer begins his coding with a specific end goal to give a full portray of item. At the end of the day framework particulars are just changed over into machine coherent register code.

Usage: The execution stage includes the genuine coding or programming of the product. The yield of this stage is regularly the library, executables, client manuals and extra programming documentation.

Testing: In this stage all projects (models) are coordinated and tried to guarantee that the complete framework meets the product prerequisites. The testing is worried with check and approval.

Support: The upkeep stage is the longest stage in which the product is upgraded to satisfy the changing client need, adjust to suit change in the outside environment, right mistakes and oversights beforehand undetected in the testing stage, improve the proficiency of the product.

5.1.2 System Architecture

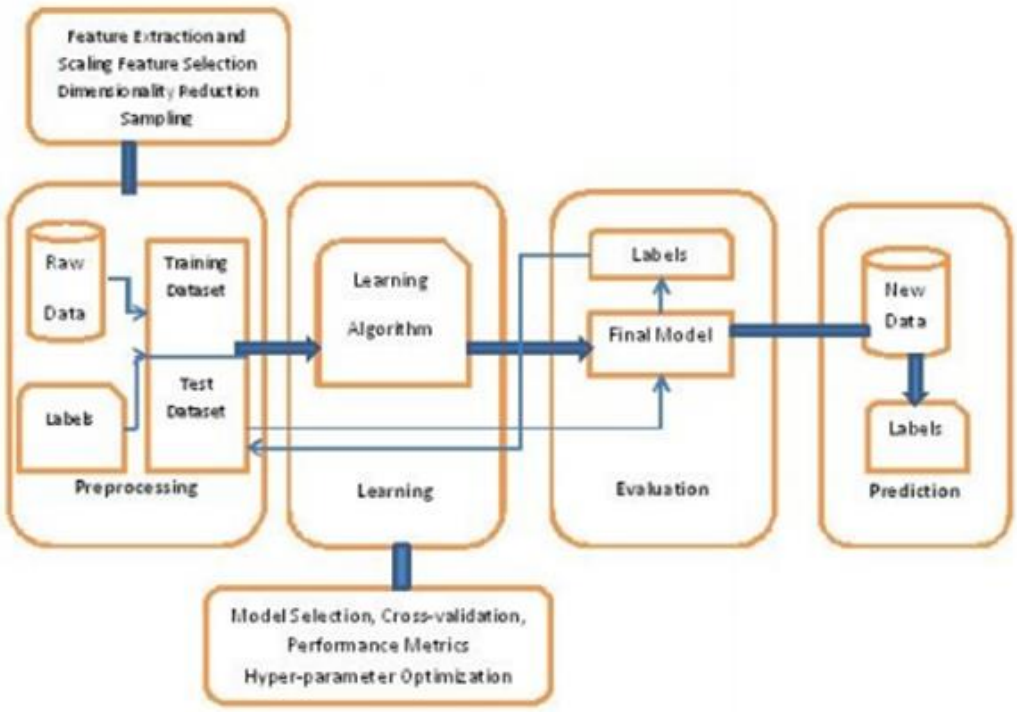


Fig 3- System Architecture

Chapter 6

IMPLEMENTATION

DATA COLLECTION

In [1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

DATA CLEANING

In [6]:
```
matches[pd.isnull(matches['winner'])]
#find all NaN values in winner column, so that we update this as draw
```

In [9]:
```
matches.replace(['Mumbai Indians','Kolkata Knight Riders','Royal Challengers B
angalore','Deccan Chargers','Chennai Super Kings',
                 'Rajasthan Royals','Delhi Daredevils','Gujarat Lions','Kings
 XI Punjab',
                 'Sunrisers Hyderabad','Rising Pune Supergiants','Kochi Tusker
s Kerala','Pune Warriors']
                ,['MI','KKR','RCB','DC','CSK','RR','DD','GL','KXIP','SRH','RP
S','KTK','PW'],inplace=True)


matches.head(2)
```

In [10]:
```
encode = {'team1': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8
,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13},
          'team2': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':8
,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13},
          'toss_winner': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,
'GL':8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13},
          'winner': {'MI':1,'KKR':2,'RCB':3,'DC':4,'CSK':5,'RR':6,'DD':7,'GL':
8,'KXIP':9,'SRH':10,'RPS':11,'KTK':12,'PW':13,'Draw':14}}
matches.replace(encode, inplace=True)
matches.head(2)
```

In [13]:
```
#remove any null values, winner has hence fill the null value in winner as dra
w
#City is also null, this is mainly for Dubai stadium. Hence update the City as
Dubai
#Make sure to impute the data(cleansing and finding missing data), there is al
so other process
#to verify expected value based on other resultants, for now by stadium, city
 is easily manually updated
matches['city'].fillna('Dubai',inplace=True)
matches.describe()
matches.info()
```

DATA VISUALIZATION

In [18]:
```
#31 cities
df["city"].unique()
```

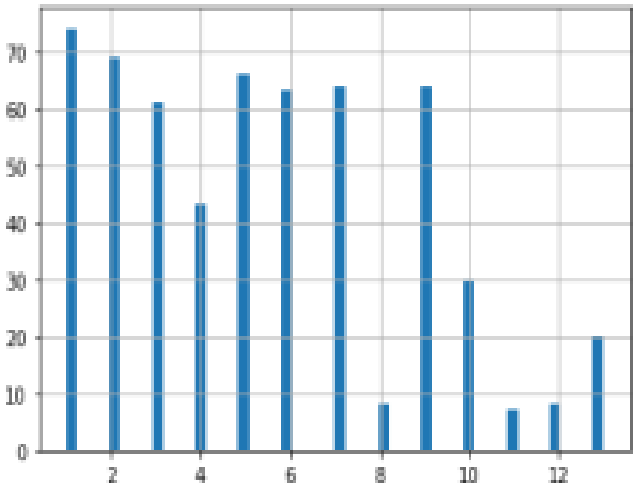
Out[18]:
```
array(['Bangalore', 'Chandigarh', 'Delhi', 'Mumbai', 'Kolkata', 'Jaipur',
       'Hyderabad', 'Chennai', 'Cape Town', 'Port Elizabeth', 'Durban',
       'Centurion', 'East London', 'Johannesburg', 'Kimberley',
       'Bloemfontein', 'Ahmedabad', 'Cuttack', 'Nagpur', 'Dharamsala',
       'Kochi', 'Indore', 'Visakhapatnam', 'Pune', 'Raipur', 'Ranchi',
       'Abu Dhabi', 'Sharjah', 'Dubai', 'Rajkot', 'Kanpur'], dtype=object)
```

In [23]:
```
#Find some stats on the match winners and toss winners
temp1=df['toss_winner'].value_counts(sort=True)
temp2=df['winner'].value_counts(sort=True)
#Mumbai won most toss and also most matches
print('No of toss winners by each team')
for idx, val in temp1.iteritems():
   print('{} -> {}'.format(list(dicVal.keys())[list(dicVal.values()).index(idx
)],val))
print('No of match winners by each team')
for idx, val in temp2.iteritems():
   print('{} -> {}'.format(list(dicVal.keys())[list(dicVal.values()).index(idx
)],val))
```

```
No of toss winners by each team
MI -> 74
KKR -> 69
CSK -> 66
KXIP -> 64
DD -> 64
RR -> 63
RCB -> 61
DC -> 43
SRH -> 30
PW -> 20
KTK -> 8
GL -> 8
RPS -> 7
No of match winners by each team
MI -> 80
CSK -> 79
RCB -> 70
KKR -> 68
KXIP -> 63
RR -> 63
DD -> 56
SRH -> 34
DC -> 29
PW -> 12
GL -> 9
KTK -> 6
RPS -> 5
Draw -> 3
```

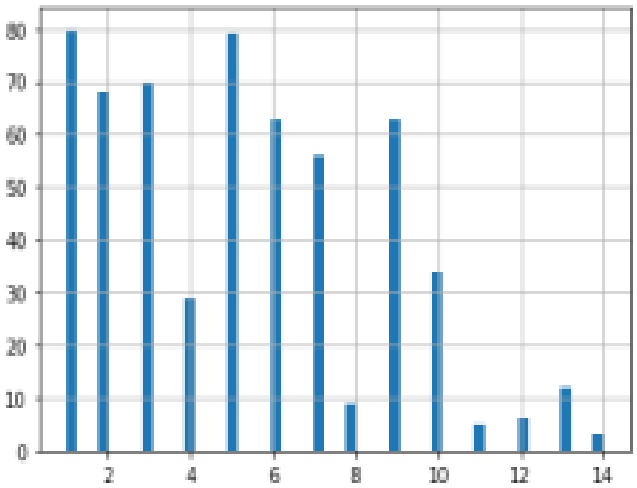
In [54]:
```
df['toss_winner'].hist(bins=50)
```

Out[54]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f1db4b2b88>
```



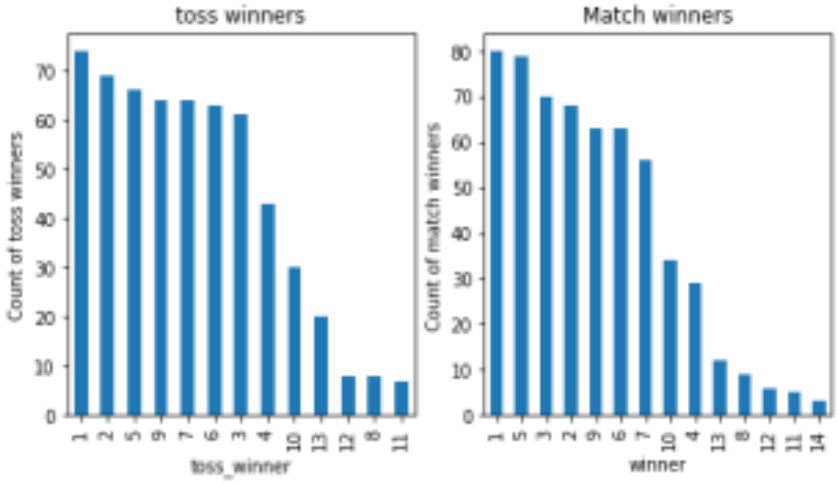
In [25]:
```
#shows that Mumbai won most matches followed by Chennai
df['winner'].hist(bins=50)
```

Out[25]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1f1d7bfdf08>
```



In [26]:
```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(8,4))
ax1 = fig.add_subplot(121)
ax1.set_xlabel('toss_winner')
ax1.set_ylabel('Count of toss winners')
ax1.set_title("toss winners")
temp1.plot(kind='bar')

ax2 = fig.add_subplot(122)
temp2.plot(kind = 'bar')
ax2.set_xlabel('winner')
ax2.set_ylabel('Count of match winners')
ax2.set_title("Match winners")
```

Out[26]:
```
Text(0.5, 1.0, 'Match winners')
```



In [27]:
```
df.apply(lambda x: sum(x.isnull()),axis=0)
    #find the null values in every column
```

Out[27]:
```
team1           0
team2           0
city            0
toss_decision   0
toss_winner     0
venue           0
winner          0
dtype: int64
```

MODEL DEVELOPMENT AND EVALUATION

In [30]:
```
#Import models from scikit learn module:
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import KFold   #For K-fold cross validation
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn import metrics

#Generic function for making a classification model and accessing performance:
def classification_model(model, data, predictors, outcome):
      model.fit(data[predictors],data[outcome])
      predictions = model.predict(data[predictors])
      print(predictions)
      accuracy = metrics.accuracy_score(predictions,data[outcome])
      print('Accuracy : %s' % '{0:.3%}'.format(accuracy))
```

In [31]:
```
#Logistic Regression
outcome_var=['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decisio
n']
model =LogisticRegression()
classification_model(model, df,predictor_var,outcome_var)
```

In [32]:
```
#Gaussian NAive bayes algorithm
from sklearn.naive_bayes import GaussianNB
outcome_var=['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decisio
n']
model = GaussianNB()
classification_model(model, df,predictor_var,outcome_var)
```

In [33]:
```
#applying knn algorithm
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=3)
classification_model(model, df,predictor_var,outcome_var)
```

In [35]:
```
#Import Library
from sklearn import svm
#Assumed you have, X (predictor) and Y (target) for training data set and x_te
st(predictor) of test_dataset
# Create SVM classification object
model = svm.SVC(kernel='rbf', C=1, gamma=1)
outcome_var=['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decisio
n']
# there is various option associated with it, like changing kernel, gamma and
 C value. Will discuss more # about it in next section.Train the model using t
he training sets and check score
classification_model(model, df,predictor_var,outcome_var)
```

In [37]:
```
#Decision tree algorithm
from sklearn import tree
model = tree.DecisionTreeClassifier(criterion='gini')
outcome_var=['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decisio
n']
classification_model(model, df,predictor_var,outcome_var)
```

In [36]:
```
#Gradient boost algorithm
from sklearn.ensemble import GradientBoostingClassifier
model= GradientBoostingClassifier(n_estimators=1000, learning_rate=0.1, max_de
pth=3, random_state=0)
classification_model(model, df,predictor_var,outcome_var)
```

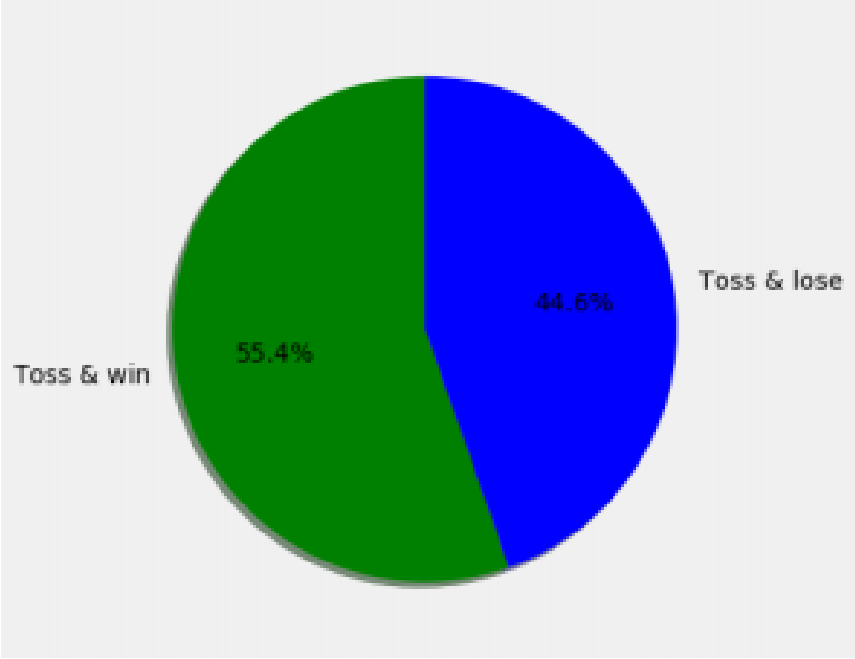
In [38]:
```
#Random forest classifier
model = RandomForestClassifier(n_estimators=100)
outcome_var = ['winner']
predictor_var = ['team1', 'team2', 'venue', 'toss_winner','city','toss_decisio
n']
classification_model(model, df,predictor_var,outcome_var)
```

Chapter 7

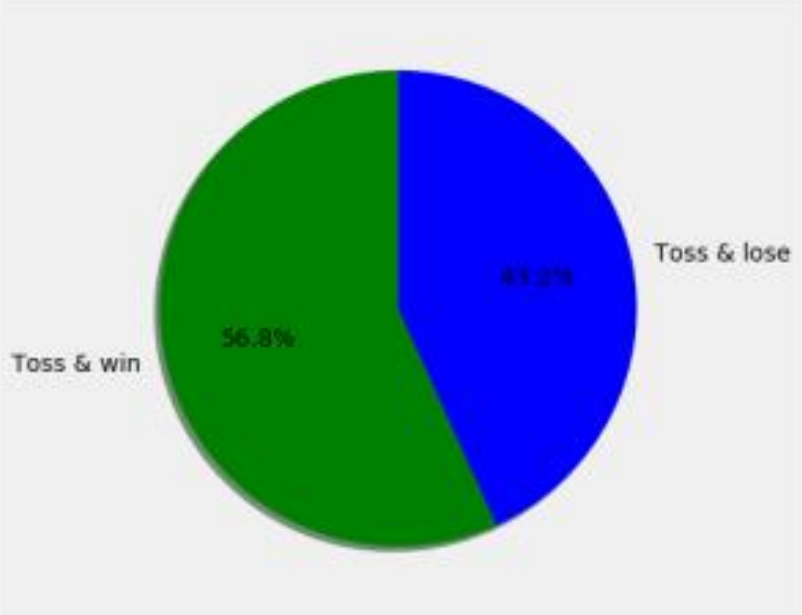
RESULTS

MODEL	ACCURACY
LOGISTIC REGRESSION	30.329%
GAUSSIAN NAÏVE BAYES ALGORITHM	20.264%
KNN ALGORITHM	62.565%
SUPPORT VECTOR MACHINE	89.081%
GRADIENT BOOST ALGORITHM	89.601%
DECISION TREE ALGORITHM	89.601%
RANDOM FOREST CLASSIFIER	89.601%

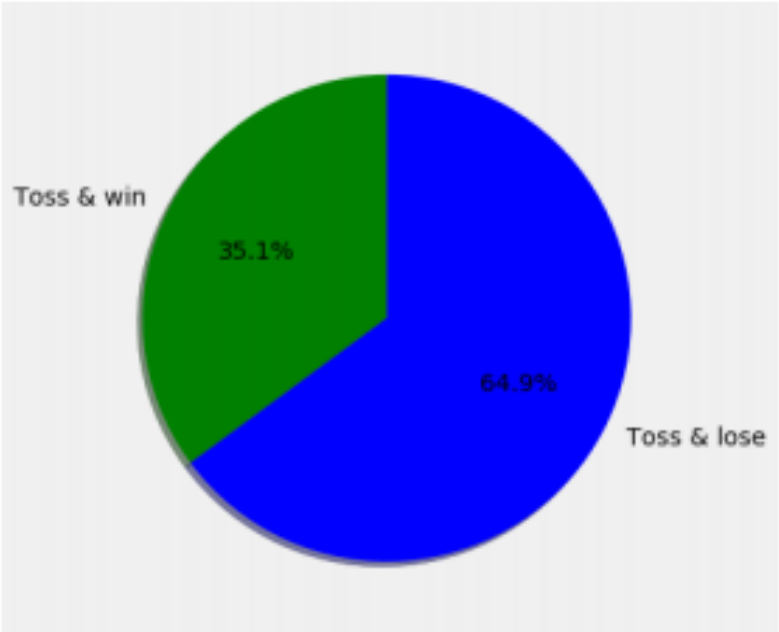
In [56]:
```
#probability of match winning by winning toss for MI
#df['toss_winner'].value_counts()
count =0
for i in range(577):
    if df["toss_winner"][i]==df["winner"][i]==1 :
        count=count+1
#okay from the above prediction on features, we notice toss winner has least c
hances of winning matches
#but does the current stats shows the same result
#df.count --> 577 rows
import matplotlib.pyplot as mlt
mlt.style.use('fivethirtyeight')
#df_fil=df[df['toss_winner']==df['winner']]
df_fil=df[df['toss_winner']==1]
#slices=[len(df_fil),(577-len(df_fil))]
slices=[count,(len(df_fil)-count)]
mlt.pie(slices,labels=['Toss & win','Toss & lose'],startangle=90,shadow=True,e
xplode=(0,0),autopct='%1.1f%%',colors=['g','b'])
fig = mlt.gcf()
fig.set_size_inches(6,6)
mlt.show()

# Toss winning does not gaurantee a match win from analysis of current stats a
nd thus
#prediction feature gives less weightage to that
```



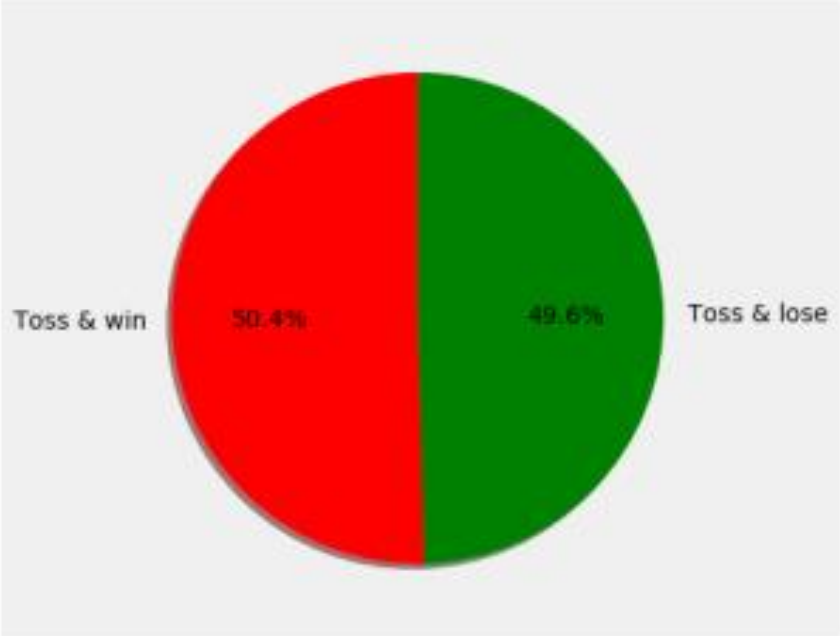
In [57]:
```
#probability of match winning by winning toss for Chennai Super Kings CSK
#df['toss_winner'].value_counts()
count =0
for i in range(577):
    if df["toss_winner"][i]==df["winner"][i]==5 :
        count=count+1
#okay from the above prediction on features, we notice toss winner has least c
hances of winning matches
#but does the current stats shows the same result
#df.count --> 577 rows
import matplotlib.pyplot as mlt
mlt.style.use('fivethirtyeight')
#df_fil=df[df['toss_winner']==df['winner']]
df_fil=df[df['toss_winner']==1]
#slices=[len(df_fil),(577-len(df_fil))]
slices=[count,(len(df_fil)-count)]
mlt.pie(slices,labels=['Toss & win','Toss & lose'],startangle=90,shadow=True,e
xplode=(0,0),autopct='%1.1f%%',colors=['g','b'])
fig = mlt.gcf()
fig.set_size_inches(6,6)
mlt.show()
# Toss winning does not gaurantee a match win from analysis of current stats a
nd thus
#prediction feature gives less weightage to that
```



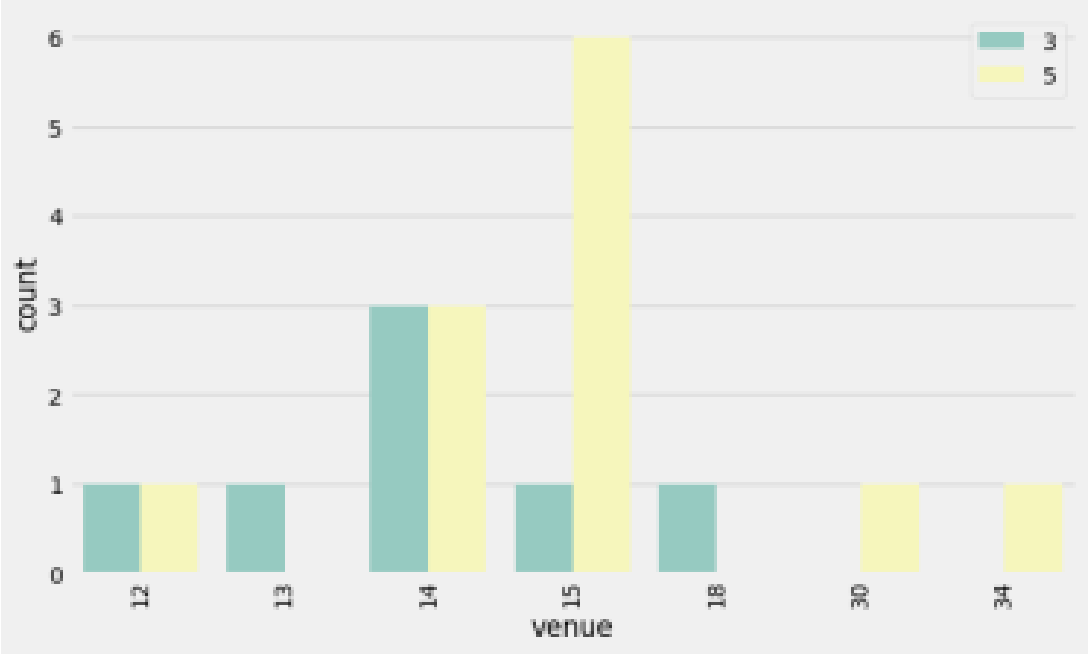
In [58]:
```
#probability of match winning by winning toss for KXIP
#df['toss_winner'].value_counts()
count =0
for i in range(577):
    if df["toss_winner"][i]==df["winner"][i]==9 :
        count=count+1
#okay from the above prediction on features, we notice toss winner has least c
hances of winning matches
#but does the current stats shows the same result
#df.count --> 577 rows
import matplotlib.pyplot as mlt
mlt.style.use('fivethirtyeight')
#df_fil=df[df['toss_winner']==df['winner']]
df_fil=df[df['toss_winner']==1]
#slices=[len(df_fil),(577-len(df_fil))]
slices=[count,(len(df_fil)-count)]
mlt.pie(slices,labels=['Toss & win','Toss & lose'],startangle=90,shadow=True,e
xplode=(0,0),autopct='%1.1f%%',colors=['g','b'])
fig = mlt.gcf()
fig.set_size_inches(6,6)
mlt.show()
# Toss winning does not gaurantee a match win from analysis of current stats a
nd thus
#prediction feature gives less weightage to that
```



In [59]:
```
#genralised probbaility for winning match by winning toss


import matplotlib.pyplot as mlt
mlt.style.use('fivethirtyeight')
df_fil=df[df['toss_winner']==df['winner']]

slices=[len(df_fil),(577-len(df_fil))]

mlt.pie(slices,labels=['Toss & win','Toss & lose'],startangle=90,shadow=True,e
xplode=(0,0),autopct='%1.1f%%',colors=['r','g'])
fig = mlt.gcf()
fig.set_size_inches(6,6)
mlt.show()
```



In [60]:
```
#top 2 team analysis based on number of matches won against each other and how
venue affects them?
#Previously we noticed that CSK won 79, RCB won 70 matches
#now Let us compare venue against a match between CSK and RCB
#we find that CSK has won most matches against RCB in MA Chidambaram Stadium,
 Chepauk, Chennai
#RCB has not won any match with CSK in stadiums St George's Park and Wankhede
 Stadium, but won matches
#with CSK in Kingsmead, New Wanderers Stadium.
#It does prove that chances of CSK winning is more in Chepauk stadium when pla
yed against RCB.
# Proves venue is important feature in predictability
import seaborn as sns
team1=dicVal['CSK']
team2=dicVal['RCB']
mtemp=matches[((matches['team1']==team1)|(matches['team2']==team1))&((matches[
'team1']==team2)|(matches['team2']==team2))]
sns.countplot(x='venue', hue='winner',data=mtemp,palette='Set3')
mlt.xticks(rotation='vertical')
leg = mlt.legend( loc = 'upper right')
fig=mlt.gcf()
fig.set_size_inches(10,6)
mlt.show()
le.classes_[15]
```



Out[60]: 'MA Chidambaram Stadium, Chepauk'

Chapter 8

FUTURE SCOPE AND CONCLUSION

Selection of the best team for a cricket match plays a significant role for the team's victory. The main goal of this paper is to analyse the IPL cricket data and predict the players' performance. Here, three classification algorithms are used and compared to find the best accurate algorithm. The implementation tools used are Anaconda navigator and Jupyter. Random Forest is observed to be the best accurate classifier with 89.15% to predict the best player performance. This knowledge will be used in future to predict the winning teams for the next series IPL matches. Hence using this prediction, the best team can be formed. This project opens scope for future work in the field of cricket and predicting other important things like best team of players, best venue, best city, best fielding decision to win a match.

REFERENCES

T. A. Severini, Analytic methods in sports: Using mathematics and statistics to understand data from baseball, football, basketball, and other sports. Chapman and Hall/CRC, 2014.

8. H. Ghasemzadeh and R. Jafari, "Coordination analysis of human movements with body sensor networks: A signal processing model to evaluate baseball swings," IEEE Sensors Journal, vol. 11, no. 3, pp. 603–610, 2010

9. R. Rein and D. Memmert, "Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science," SpringerPlus, vol. 5, no. 1, p. 1410, 2016

Veppur Sankaranarayanan, Vignesh and Sattar, Junaed and Lakshmanan,"Auto-play: A Data Mining Approach to ODI Cricket

Simulation and Prediction",SIAM Conference on Data Mining, 2014
K. A. A. D. Raj and P. Padma, "Application of Association Rule

Mining: A case study on team India", 2013 International Conference on Computer Communication and Informatics, 2013

Tim B. SWARTZ, Paramjit S Gill and S. Muthukumarana,"Modelling and simulation for one-day cricket", Canadian Journal of Statistics, 2009, Vol 37, No 2, pp-143-160