
Using Machine Learning Techniques for Finding Rising Stars in Team Sports

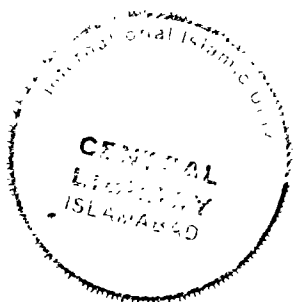


Ph.D. (Computer Science)

By

Zafar Mahmood

82-FBAS/PHDCS/F12



Supervisor

Dr. Ali Daud

Associate Professor

Co-Supervisor

Dr. Rabeeh Ayaz Abbasi

Associate Professor

Department of Computer Science & Software Engineering

Faculty of Basic & Applied Sciences

International Islamic University, Islamabad

(2022)

PHD
006-31
ZP

Accession No TH-25938 ^{VII}

Machine in the
control section
of the plant

INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD
FACULTY OF BASIC & APPLIED SCIENCES
DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING

Date: 25-08-2022


Final Approval

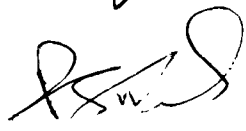
It is certified that we have read this thesis, entitled "Using Machine Learning Techniques for Finding Rising Stars in Team Sports" submitted by Mr. Zafar Mahmood, Registration No. 82-FBAS/PHDCS/F12. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for the award of the degree of PhD in Computer Science.

Committee

External Examiners:

Prof. Dr. Basit Raza,
COMSATS University,
Islamabad.






Dr. Sher Afgan Usmani,
PropSure Digital Solution Pvt. Ltd, NSTP, NUST,
Islamabad.

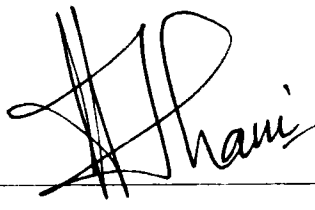
Internal Examiner:

Dr. Asim Munir,
Chairman/Assistant professor
Department of Computer Science & Software Engineering
FBAS, IIUI



Supervisor:

Dr. Ali Daud,
Associate Professor
Department of Computer Science & Software Engineering
FBAS, IIUI

for 

Co-Supervisor:

Dr. Rabeeh Ayaz Abbasi,
Associate Professor
Department of Computer Science QAU,
Islamabad



Declaration

I hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that no portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Zafar Mahmood

Dedication

Dedicated to My Teachers, Parents, Family and Friends.

Zafar Mahmood

Acknowledgments

I am very grateful to *ALLAH* the *ALMIGHTY*, without His grace and blessing this study would not have been possible.

Foremost, I would like to express my sincere gratitude to my supervisor *Dr. Ali Daud* for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

I would like to extend immeasurable appreciation and deepest gratitude for the help and support of my Co-Supervisor *Dr. Rabeeh Ayaz Abbasi*, who gave all his knowledge, guidance and support to boost my confidence and learning. His mentoring and encouragement have been specially valuable, and his early insights launched the greater part of this dissertation.

It is my pleasure to thank *Dr. Ali Daud* for introducing and motivating me to work on the topic of Rising Star Prediction. I am extremely grateful to Higher Education Commission (HEC) of Pakistan for the financial support throughout my PhD degree. This work is completed under the 5000-Indigenous PhD fellowship program.

Last but not least I wish to avail this opportunity, to express a sense of gratitude and love to my beloved parents and family for their moral support, strength, help and for everything.

I would also like to thank my wife who supported me patiently and firmly during completion of this task.

Abstract

Rising stars in any field are the persons that have the potential to become popular in near future. Exploring rising stars in any organization will help the organization in its decisions making. The concept of the rising star has been applied for finding rising authors in the research community, rising business managers in telecommunication industry and rising players in the game of cricket. In this thesis, we presented the rising star prediction in basketball as a machine learning problem. We presented three types of co-players: co-players of same team in same game, co-players of opponent team in same game and co-players of both same and opponent team in same game. Co-players statistics are used as features for machine learning models.

For basketball game, co-player features are classified by feature size and type, which are further divided into different categories. Derived features along with their mathematical formulation are presented, that are derived from players statistics. The impact of co-players on prediction of rising star is measured through various machine learning models. Experimental results show that derived features are dominant on different datasets in terms of F-measure score. The highest F-measure score achieved by derived features is 96%. Comparison of different machine learning models shows that Maximum Entropy Markov Model is dominant on all datasets in terms of F-measure score. The highest F-measure score achieved by Maximum Entropy Markov Model is 96%. Ranking comparison shows that most of the labeled rising stars are ranked in the top 100 in the subsequent six seasons. Comparison of rising stars with NBA (National Basketball Association) most improved players shows that rising stars have better efficiency in those seasons for which NBA most improved players were selected.

For prediction of rising stars in baseball, we aimed to identify best features. We used baseball game statistics as attributes for machine learning models. To identify the best features we used random forest classifier for feature selection. Four categorize of features were obtained through feature selection. The first category consists of five features, second consist of ten features, third consist of fifteen features and fourth category consists of twenty features. The results show that first category of features with minimum number of features achieved highest F-measure score than other category of features. We also carried out analysis that shows that those players who appeared in games with expert players have improved their performance in future as compared to players who never appeared in any game with expert players.

Contents

1	Introduction	1
1	Introduction	2
1.1	Machine Learning	3
1.2	Motivation	7
1.3	Problem Statment	7
1.4	Problem Formulation	7
1.5	Research Questions	8
1.6	Research Objectives	9
1.7	Contribution	9
1.8	Performance Measures	10
1.9	Thesis Structure	12
2	Literature Review	13
2	Literature Review	14
2.1	Ranking Players in Team Sports	14
2.2	Performance Analysis of Players	15
2.3	Match Outcome Prediction in Sports	18
2.4	Rising Star Prediction in Academic Network	22
2.5	Rising Star Prediction in Sports	24
2.6	Chapter Summary	28
3	Rising Star Prediction in Basketball	29
3	Rising Star Prediction in Basketball	30
3.1	Basic Concepts of Basketball	30
3.2	Player Statistics	31
3.3	Basketball Datasets	34
3.4	Proposed Steps For Rising Star Prediction	37
3.5	Machine Learning Techniques	39
3.6	Co-player Selection Criteria	40
3.7	Features for Rising Star Prediction	45

3.8	Mathematical Formulation of Derived Features	50
3.9	Experiments	57
3.10	Results and Discussion	63
3.11	Season-wise Rankig Comparison Of Top 20 Labeled Rising Stars .	75
3.12	Rising Stars VS NBA Most Improved Player	77
3.13	Chapter Summary	78
4	Exploring attributes for Rising Stars in Baseball	79
4	Exploring Attributes for Rising Stars in Baseball	80
4.1	Basic Concepts of Baseball	80
4.2	Baseball Dataset	81
4.3	Features for Rising Star Prediction	83
4.4	Features Categorization	84
4.5	Experiments	84
4.6	ML models For Rising Star Prediction in other Domains	87
4.7	Chapter Summary	88
5	Impact of Expert Players on Performance of Junior Players	89
5	Impact of Expert Players on Performance of Junior Players	90
5.1	Dataset Description and Experiment Setup	90
5.2	Candidates Comparison of Group A and Group B	91
5.3	Performance Analysis of Group A and Group B	92
5.4	Chapter Summary	95
6	Conclusion and Future Work	96
6	Conclusion and Future Work	97
6.1	Conclusion	97
6.2	Future Work	98

List of Figures

1.1	Confusion Matrix.	10
3.1	Flow Chart showing dataset acquisition process for basketball data.	36
3.2	Rising Star Prediction Model	38
3.3	Players of same team and same game.	41
3.4	Players of opposite team and same game.	42
3.5	Players of same and opposite team in same game.	42
3.6	Relationship between co-players(same team) and rising star players. (x-axis represents average efficiency of co-players. y-axis represents average-efficiency of players labeled as rising and not-rising stars)	43
3.7	Relationship between co-players(opponent team) and rising star players. (x-axis represents average efficiency of co-players. y-axis represents average-efficiency of players labeled as rising and not-rising stars)	44
3.8	Relationship between co-players(both same and opponent team) and rising star players. (x-axis represents average efficiency of co-players. y-axis represents average-efficiency of players labeled as rising and not-rising stars)	45
3.9	Features Categorization.	45
3.10	Features Distribution of Selected Features for Dataset A. (Red color shows rising star and blue color shows not-rising star. X-axis shows distribution of values and Y-axis shows frequency of values for an attribute.)	59
3.11	Features Distribution of Selected Features for Dataset B.	60
3.12	Features Distribution of Selected Features for Dataset C.	60
3.13	K-fold cross validation with k=10, E is Evaluation metric.	62
3.14	Individual Feature Analysis of Dataset A.	64
3.15	Individual Feature Analysis of Dataset B.	65
3.16	Individual Feature Analysis of Dataset C.	66
3.17	Comparison of F-measure score of different feature categories.	68
3.18	Comparison of F-measure score of different classifiers on three datasets.	69
3.19	Season Wise Ranking of Top 20 Labeled Rising Stars.	77
3.20	Comparison of Rising Stars Vs NBA Most Improved Players (Bar labeled with text "MIP" shows NBA most improved player of releavent season)	78

4.1	Flow Chart showing dataset acquisition process for baseball data.	82
4.2	Comparison of different Feature Sets.	86
4.3	Comparison of different Machine Learning Classifiers.	87
5.1	Candidates comparison of Group A and Group B.	92
5.2	Performance analysis of Group A and Group B.	94
5.3	Performance analysis of Group A and Group B (Percent Wise).	95

List of Symbols/Abbreviations

Symbol	Detail
CART	Classification and Regression Trees
SVM	Support Vector Machines
MEMM	Maximum Entropy Markov Model
BN	Bayesian Network
NB	Naive Bayes
NBA	National Basketball Association
MVP	Most Valuable Player
EFF	Efficiency
PTS	Points
REB	Rebounds
AST	Assists
STL	Steals
BLK	Blocks
FGA	Field Goal Attempts
FT	Free Throw
FTA	Free Throw Attempts
TOV	Turn Over
HGS	Hollinger Score

List of Publications

Publication:

Mahmood, Zafar, Ali Daud, and Rabeeh Ayaz Abbasi. "Using machine learning techniques for rising star prediction in basketball". Knowledge-Based Systems 211 (2021): 106506. (Impact Factor: 8.139).

Chapter 1

Introduction

1 Introduction

Rising Stars are the persons who have low performance at the start of their career but have the potential to become experts in their field shortly [1]. Rising star prediction is very useful to know the impact of a recently joined member on the future performance of an organization. In an organization, their members can be ranked by analyzing their past statistics but in the case when a member has only spent few years in an organization and there is not much statistics to rank him, in such case concept of the rising star is very useful because in rising star prediction a member is predicted as a rising star or not rising star by finding whether a member's performance increase or decrease while working with senior and expert teammates of the same organization.

Initially concept of rising star was used to find the experts in author networks. PubRank algorithm proposed by Li et al. [1] explored rising stars in author's network. StarRank algorithm is presented by Daud et al. [2] which is better than PubRank because StarRank considers the author's contribution based on mutual influence and dynamic publication venue scores, whereas PubRank only considers author's mutual influence and static ranking of journals or conferences. The evolution of authors over time is presented by Tsatsaronis et al. [3], they defined four types of author evolution as rising stars, declining authors, authors with stable publication rate and well-established authors. Machine learning models were used by Daud et al. [4] for rising star prediction in co-author network. They proposed several types of features and fed these features into machine learning algorithms. They ranked the authors based on the feature scores. Weighted Mutual Influence Rank (WMIRank) method is proposed by Daud et al. [5] for finding rising stars in co-author networks. WMIRank method is based on co-author's citations based on mutual influence, co-author's order based mutual influence and co-author venue's citations based mutual influence. The impact of senior scholars on junior ones is studied by Amjad et al. [6], their study revealed that a junior author can become an expert if had a chance to work with a senior and expert author. Their study also revealed that if an author does not get a chance to work with the expert and senior authors, still the junior author can become an expert soon by his hard work. Bibliometric and collaborative information of scholars were used by Panagopoulos et al. [7] for extracting and analyzing profiles of scholars. Inner factors were used by Ding et al. [8] to evaluate rising stars in a heterogeneous social network. The influence of co-authors and the quality cited papers have been used by Nie et al. [9] to predict academic rising stars. Detailed discussion on rising stars in bibliometric networks can be found in [10]. Recent

research on the application of rising stars to predict rising business manager is carried out by Daud et al.[11]. Ranking of cricket teams based on ranking algorithms is presented by [12]. Daud et al. [13] discuss about falsely predicted rising stars. In sports, the concept of the rising star has only been applied to the game of cricket by Ahmad et al. [14], they used machine learning techniques for the rising star prediction in the domain of batting and bowling. They used co-players, team and opposite team features. The predicted rising star players are ranked with respect to feature scores.

1.1 Machine Learning

Machine learning is part of artificial intelligence. Machine learning enables the systems to take decisions on their own without being programmed. Machine learning nowadays are widely used for face recognition, natural language processing, speech recognition, self driving cars and in various industries for the purpose of automation. There are different types of machine learning algorithms. One common thing in all machine learning algorithms is that they need some data from which they can make decisions. Machine learning algorithms work in the following steps:

1. Provide past data as an input.
2. ML algorithm learn from the input data.
3. Build a logical model that can take actions on new/unseen data.
4. On the basis of logical model built, output or actions on new data can be made.

Performance of machine learning models depends both on working of algorithms and quality of data fed into the ML model. Machine learning is broadly categorized into three categories:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

1.1.1 Supervised Learning

Supervised learning is the type of machine learning where the input data is labeled. Prior to use machine learning algorithm the raw data is needed to be converted into such format that is compatible with the machine learning algorithms. The data provided to the machine

learning algorithm is also called training data. Training data can be represented as a matrix. Each row of the matrix is called instance or test case whereas each column in the matrix is called attribute or feature. In supervised Learning the last column is called the class or label of the instance.

When training data is fed into machine learning algorithm, the algorithms try to learn from the data and construct a logical model which is then used to find the class or label of unseen instance. The unseen data is also called the test data. Supervised learning is further divided into two types:

1. Classification
2. Prediction

When the class or label is discrete value, then such type of learning is called classification. For example face recognition training data contains class as names of the persons which is non-numeric. Some of the well known classification algorithms are: Naive Bayes[15], Logistic Regression[16], Decision Tree [17], K-Nearest Neighbors[18] and Support Vector Machines[19].

When the class or label, then such type of learning is called prediction. For example students marks prediction data contains label as marks of the students which are numeric values. Some of the well known regression methods are: Linear Regression, Multiple Linear Regression and Polynomial Regression.

1.1.2 Unsupervised Learning

In unsupervised learning the input data is not labeled. The aim of unsupervised learning is to divide data into different groups or categories on basis of their similarity. Unsupervised learning can be used to discover hidden patterns inside the data. Some of the well known unsupervised learning methods are:

1. K-means[20]
2. Agglomerative[21]
3. Fuzzy C-Means[22]

K-means clustering method group the data into different clusters. First k number of required clusters are defined and initial centers of the clusters are randomly chosen. In first iteration

distance of each data point is measured with randomly selected center of each cluster. The data point is assigned to the center with minimum distance. In second iteration the centers of clusters are recomputed and again distance of each point is measured with updated centers of each cluster. The process of recomputing the centers continues till the centers remain unchanged or certain number of iterations are completed.

Agglomerative clustering is a type of hierarchical clustering. At start each data point is a separate cluster. Clusters that are close to each other are merged into a single cluster. The process of merging clusters continues until all clusters are merged into a single cluster. History of grouping clusters is visualized through dendrograms to find required number of clusters.

Like K-means, Fuzzy C-Means clustering method also required number of clusters to be defined by the user. Unlike K-means where each data point belongs to a separate cluster, in Fuzzy C-Means clustering each data point belongs to every cluster but with different likelihood for each cluster.

1.1.3 Reinforcement Learning

Reinforcement is a type of machine learning where the learning of agents is based on interaction with the environment through trial and error. Learning of agents is improved with time through feedback from their experiences and actions. Reward and penalty is used by reinforcement learning for making right or wrong decision. Unsupervised learning finds similarity among data items whereas the goal of reinforcement learning is to find such type of actions which maximize the total reward of agent.

Some of the popular reinforcement algorithms are:

1. Q-learning[23]
2. SARSA (State Action Reward State Action)[24]
3. Deep Q-Networks (DQNs)[25]
4. Deep Deterministic Policy Gradient (DDPG)[25]

Q-learning and SARSA are similar in terms of the exploitation strategies they used but the exploration strategies used by both algorithms are different. Off policy method is used by Q-learning where the value learned by an agent is based on the action which is derived from another policy. Q-learning and SARSA can easily be implemented but they lack the ability

to predict values from unseen data. DQN's algorithm which is based on neural networks and find q-values can be used to overcome the limitation of SARSA and Q-learning. DQN's is not suitable for high dimensional space. DDPG which is an off policy algorithm has the ability to learn in high dimensional space.

Reinforcement learning has wide range of applications. Most of computer games that exhibit artificial intelligence widely used reinforcement learning. Backgammon and ATARI are examples of such games that are based on reinforcement learning. Reinforcement learning is also used extensively in automation of industry and robotics.

1.1.4 Machine Learning Applications

Machine learning models have wide range of applications. Here we give an overview of some of the application of machine learning techniques.

SVM [19] and Naive Bayes [15] techniques have been used by Wawre et al.[26] for classification of movie reviews. Text classification based on document embedding is used by Sinoara et al. [27]. One of the application of machine learning in the domain of legal documents is presented by Chalkidis et al. [28], where the authors applied various models for multi-label text classification on legislation documents. Words in pair neural networks is presented by Yujia et al. [29] for text classification that overcome the limitation of text classification based on single word with multiple meanings. Novel machine learning model SS3 proposed by Burdisso et al. [30] for text classification that have the ability of early risk detection on social media. siame capsule networks that are based on local and global features for text classification has been used by Wu et al. [31]

Machine learning has also been actively used for classification of spam messages. A review of soft techniques for classification of sms spam is presented by Abayomi et al. [32]. Discrete Hidden Markov Model is used by Xia et al. [33] for spam detection that has the capability to exploit the order of words and can handle the problem of low term frequency. Rule based algorithm with the ability of constant time complexity has been used for detection of spam by Xia et al. [34].

Classical machine learning technique are not much efficient in situations where the decisions are time-dependent, for such situations Chen et al. [35] presented a machine learning model that have the ability to work in time varying systems.

Machine learning techniques based on evolutionary framework has been used in medical

domain on clinical data by Castellanos et al. [36]. For prediction of breast cancer, Support Vector Machines and Artificial Neural Networks has been applied by Bayrak et al. [37] on Wisconsin Breast Cancer dataset.

1.2 Motivation

As we discussed in start of this chapter that rising stars are the persons who have low performance at start of their career but have the potential to become expert in near future. In sports players are ranked every year and players who are ranked high are considered as efficient players. The limitation of ranking players is that it purely rely on players individual performance and it does not tell whether the player will be efficient in coming years or not. Prediction of rising stars are based on co-players performance instead of player individual performance and rising stars are the players that have the tendency to be efficient players in next coming years.

1.3 Problem Statment

The main problem of this research is prediction of rising star in team sports. Rising star prediction problem can be presented as binary classification problem. First players are labeled as rising and not rising star players based on their performance and then releavent features are identified. This research consider features of co-players of rising star/not rising star players. Three types of co-players are considered in this research: co-players belong to same team, co-players belong to opponent team and co-players of both same and opponent team. The aim of using three type of co-players is to find the impact of each co-player type features on prediction of rising star.

1.4 Problem Formulation

Let we have a dataset D that contains players P , corresponding feature set F and set of class label Y . The Dataset can be represented as a Matrix as following

$$D = \begin{bmatrix} Player_1 \\ Player_2 \\ Player_3 \\ . \\ . \\ . \\ Player_m \end{bmatrix} \begin{bmatrix} co_F_{11} & co_F_{12} & co_F_{13} & \dots & co_F_{1n} \\ co_F_{21} & co_F_{22} & co_F_{23} & \dots & co_F_{2n} \\ co_F_{31} & co_F_{32} & co_F_{33} & \dots & co_F_{3n} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ co_F_{m1} & co_F_{m2} & co_F_{m3} & \dots & co_F_{mn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ . \\ . \\ . \\ y_m \end{bmatrix}$$

Where $D = Dataset$.

The set of Players is defined by vector P.

$$P = \{Player_1, Player_2, Player_3, \dots, Player_m\}$$

The corresponding set of attributes or features is defined by vector F.

$$F = \{co_F_1, co_F_2, co_F_3, \dots, co_F_n\}$$

Set of labels are define by vector Y.

$$Y = \{y_1, y_2, y_3, \dots, y_m\}$$

where

$$y_1, y_2, y_3, \dots, y_m \in \{RisingStar, Not-RisingStar\}$$

For a give feature vector x:

$$x = \{x_1, x_2, x_3, \dots, x_n\}$$

where

$$x_1 \in co_F_1, x_2 \in co_F_2, x_3 \in co_F_3, \dots, x_n \in co_F_m$$

The core objective is to find a function, such that the function can assign class label to a feature set. The function can be defined as:

$$f: F \rightarrow P(Y) \quad (1.1)$$

1.5 Research Questions

1. How the concept of co-players can be utilized for the prediction of rising stars in team sports.?

2. How statistics of co-players can be used for rising star prediction?
3. Which type of features have better results for rising star prediction?
4. Which machine learning model have better results?
5. How Expert Players affect the performance of Junior Players?

1.6 Research Objectives

Main objective of this research are:

1. To find the impact of same team, opponent team and both teams co-players on the results of rising star prediction.
2. To use co-players game statistics as features for prediction and divide these features into various categories.
3. To carry out analysis of various types of features for prediction of rising stars in basketball and baseball.
4. To carry out analysis of various machine learning models for rising star prediction in basketball and baseball.
5. To find impact of expert players on performance of junior players.

1.7 Contribution

Main contributions of this thesis are:

1. Three types of co-players are introduced in this study. The first type of co-players are those players who appeared with a player in same game and belong to same team. The second type of co-players are those who appeared with a player in same game but belong to opposite team. The third type of co-players are those who appeared with a player in same game (include both same and opponent team players). The aim of presenting these three types of co-players is to examine how much each type of co-player is effective in predicting rising stars.
2. Derived features along with their mathematical formulation are presented. These features are derived from players traditional game statistics. The aim of derived features is to improve the results of machine learning models for rising star prediction.

Features are categorized by type and size and these are further divided into sub-types. Features categorized by type are further divided into “basic”, “shooting” and “derived” feature types. Features categorized by size are further divided into “all”, “selected” and “derived” feature types. The effectiveness of each category of feature is examined on different datasets.

3. The three datasets used in this study are constructed purely for rising star prediction in basketball. Each dataset consists of player name, different features of co-players and class as rising star or not rising star. The first dataset consists features of those co-players that belong to the player team. The second dataset consists of features of those co-players that belong to opponent team. The third type of dataset consists of features of co-players that belong to both same and opponent team.
4. Analysis of various types of features and machine learning classifiers showed that they can be utilized for the prediction of rising stars in team sports.

1.8 Performance Measures

The purpose of machine learning classifier is to find the class label of test data. Confusion matrix, accuracy, precision, recall and F-measure are used for the evaluation of performance of machine learning classifiers.

1.8.1 Confusion Matrix

Confusion matrix is widely used to evaluate the performance of any machine learning classifier. Each cell in confusion matrix represents the number of predictions made. Pictorial representation of confusion matrix is shown in Fig.1.1

	<i>Positive (Predicted)</i>	<i>Negative (Predicted)</i>
<i>Positive (Actual)</i>	<i>True Positive</i>	<i>False Negative</i>
<i>Negative (Actual)</i>	<i>False Positive</i>	<i>True Negative</i>

Figure 1.1: Confusion Matrix.

Confusion matrix is composed of True Positive, True Negatives, False Positives and False

Negatives. Details of these terms are given below [38]:

1. TP (True Positive): When the actual class label is Positive and the classifier predicted class of is also Positive.
2. TN (True Negative): When the actual class label is Negative and the class label predicted by classifier is also Negative.
3. FP (False Positive): When the actual class label is Negative but the class label predicted by the classifier is Positive.
4. FN (False Negative): When the actual class label is Positive but it is predicted as Negative by the classifier.

1.8.2 Accuracy

Accuracy of a classifier can be obtained by dividing the number of correctly predicted class labels by total number of predictions made by the classifier.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1.2)$$

Accuracy is a good measure where the interest is to find the accuracy of a machine learning classifier.

1.8.3 Precision

Precision of a classifier can be obtained by dividing the TP (True positive) by the sum of TP (True positive) and FP (False positives).

$$Precision = \frac{TP}{TP + FP} \quad (1.3)$$

1.8.4 Recall

Recall of a classifier can be obtained by dividing the TP (True positives) by the sum of TP (True positive) and FN (False negatives).

$$Recall = \frac{TP}{TP + FN} \quad (1.4)$$

1.8.5 F-measure

Harmonic mean of precisoin and recall is called F-measure. F-measure is a good metric for comparing performance of machine learning models.

$$F\text{-measure} = \frac{2 * Precision * Recall}{Precision + Recall} \quad (1.5)$$

1.9 Thesis Structure

The rest of the thesis is organized as, Sec 2 discuss the related work, Sec 3.1 discusses basic concepts of basketball and Proposed method that comprise of machine learning techniques, co-player selection criteria, features for rising star prediction and mathematical formulation of dervied features are discussed in Sec 3.8. Details of experiments is discussed in Sec 3.9. Sec 4 discuss about features that are used for rising star prediction in baseball. Sec 5 discuss impact of expert plays on performance of junior players. Sec 6 discuss conclusion and future work.

Chapter 2

Literature Review

2 Literature Review

The related work can be divided into ranking and prediction in basketball. In ranking the performance of basketball players is evaluated by using various statistics whereas in prediction the outcome of the basketball game is predicted using machine learning classifiers.

2.1 Ranking Players in Team Sports

NBA game statistics like points, blocks, rebounds, field goals etc ¹ are widely used for rating the basketball players. John Hollinger introduced a formula that uses player box score statistics to measure the efficiency of the player. To know how much a player is efficient, the idea of on and off the court was proposed by Fearnhead et al. [39]. They observed whether the team performance increases or decreases when a specific player is on the court or off the court. Both offensive, defensive and combination of both were used to measure the strength of NBA players. Using data from the 2008-2009 season, LeBron James was considered the best player. The impact of an NBA team player is evaluated by Deshpande et al. [40], they used a bayesian linear regression model for finding an individual player impact on the team winning. The aforementioned research ranks the players with respect to their team and across the leagues.

Slack based measure method is used by Asghar et al. [41] to rank players in NBA games and compared their ranking with player impact measure approach. They conclude that even the players who are ranked top by slack based measure approach are ranked at bottom by player impact measure approach. The reason for getting different ranking on same data is because both methods work in different manner. Koster et al. [42] shows how network ties among players are affected through individual status and group performance. Relationship between game statistics and match outcome is explored by Zhang et al. [43]. They also consider how player's technical and physical performance is affected by interaction of opposition. Inconsistencies in box score statistics is discussed by Van et al. [44]. Spatio-temporal data was used for investigating inconsistencies.

¹<http://www.espn.com/editors/nba/glossary.html>

2.2 Performance Analysis of Players

The relationship between time spent in college and performance in National Basketball Association is studied by Paulsen [53]. Analysis done by [53] revealed that there is no significant impact of time spent in college and the productivity of the player in NBA games. The said research concluded that additional college years have positive impact on performance at the beginning of career, negative impact on performance at the mid of career whereas the quality of peers does not have any significant impact on performance. The impact of the presence of peers on session rating is discussed by Minett et al. [54].

Experiment performed by Minett et al. [54] consists of 14 males having mean age of 22.4 years and the duration of cyclings was 24 minutes. After every 3 minutes, heart rate of the candidates was recorded. During cycling, rating of perceived exertion (RPE) was collected at 10 minutes interval of the cycling. Bayesian framework was used along with ordinal regression and heart rate by using linear mixed effects models to analyze RPE sessions. The analysis done by [54] concluded that session RPE are influenced by contextual psychosocial inputs.

Variety of factors related to peers are investigated by Molodchik et al. [55] that affects the performance of players in team sports. They [55] analysis consist of 5457 soccer players, 234 teams and the duration is 2010 to 2015. A metric (rating score from 1 to 100) provided by EA sports was used to measure the performance of individual players. The dataset used in [55] is based on FIFA video game simulator which is developed by EA sports. Experimental result of [55] shows that peers have significant impact on team quality. Football players improve his ratings if he play in stronger team as compare to playing in weaker team.

Player tracking data was used by Forcher et al. [56] for the analysis of defensive play in soccer. Studies that are related to defensive play using tracking data are discussed by [56] in detail. Their research focused on PubMed, Web of Science and SPORTDiscuss. Total of 23 studies were examined by utilizing standard quality assessment. Ball possession duration with respect to player position is studied by Coutinho et al. [57]. Players of age less than 15 years were grouped on the basis of their position (defenders, midfielders and attackers). Players were evaluated over three testing days. Digital videos camera was used for the recording of players movements. To evaluate tactical behaviours of team and players a notational system based on four categories was created. The said four categories are: team behavior, offensive actions of players, defensive actions and effectiveness of ball possession.

To categorise all ball possession sequences according to their duration variable, a two-step cluster analysis was performed using log-likelihood as the distance metric and Schwartz's Bayesian criterion. By building a cluster features tree, the original data were organised into pre-clusters. The conventional hierarchical clustering technique was then applied to the pre-clusters, producing a variety of solutions with various numbers of clusters. The cluster results were interpreted by calculating the average values of the continuous variable of the period of the ball possession sequence. An ideal number of three clusters were determined from a total of fifteen clustering options. The average silhouette coefficient was 0.7, suggesting that the cluster model was quite excellent. First cluster was comprised of 66 possession sequences (52.0%). Second cluster was made of 49 possession sequences (38.6%) and third cluster was formed of 12 possession sequences (9.4%).

Performance of Indian cricket team was analyzed by Tharoor et al. [58] in Test format. Various Python libraries were used to perform exploratory data analysis, visualization and statistical analysis. Machine learning models were used to evaluate the importance of various attributes. Data used by Tharoor et al. [58] was obtained from cricsheet.org. The dataset consists of test matches statistics that range from 2004 to 11 January 2022. Total number of matches contained in the dataset were 627. Number of games played by Indian cricket team were 178 and these games were selected for further analysis. Statistical analysis [58] consists of batter statistics, bowler statistics, captaincy statistics and nation wise statistics. The batter analysis showed that Kohli was the best batter who scored 7854 runs in 11 years. Ashwin who had debut in 2011 was a bowler who took maximum wickets. Captaincy analysis showed that in test cricket Virat Kohli was best captain. Nation wise analysis showed that England team was difficult to defeat. Random forest and support vector machine have been implemented by [58] for match outcome prediction in test matches played by Indian cricket team. Accuracy of 75% was achieved by random forest classifier. Data for the year 2018 was analyzed by Raju et al. [59] to obtain useful statistics like team with maximum one day matches, teams having maximum win or loss and monthly games by a team.

Effect of injuries on team and players performance is studied by Sarlis et al. [60]. They listed number of factors that are correlated with players injuries. The said factors have been used to find the impact of injuries on efficiency of players and teams in National Basketball Association games. The data used by [60] have been scraped from online sources using Python. KNIME and Excel tools were used for data mining and machine learning tasks. For time period of 2010 to March 2020, injury data of 1298 players which contain 11225 records were

collected. The data contained information about position, weight, height and age etc. Four research questions were addressed by Sarlis et al. [60] that are related to health and injuries, teams and players with mot of injures, effect of injuries on team and player performance and data science techniques utilized for analysis of plays injuries. The results of the study [60] revealed that in NBA most common type of injuries are musculoskeletal injuries. The results also showed that performance of teams and players might be degraded dute to injureis. Considering basketball analytics, the study [60] stated that teams with balanced rest and load have better performacne. Comparison of machine learning and data mining mehtods showed that to get better insights these models should be combined.

The study in Sarlis et al. [61] seeks to collect all of the necessary analytics utilized in sports as cutting-edge performance indicators using sports statistics in decision making for basketball games, teams, and players. The research work [61] investigated these four questions in their study: performance evaluation of players and teams, optimization of techniques and ratings, performance analytics and correlation between them, important factors that are helpful in predicting defender and most valuable palyer of the year. Forecasting for three basketball seasons from 2017 to 2020 have been investigated by [61]. The data was collected from various online sources and then was aggregated. After preprocessing the data was divided into four groups. The first group cotains first quarter 20 games, the second group contains second quarter 40 games, third group conatins third quarter 60 games. NBA 20 players were selected with criteria of 15 minutes of minimum playing time and minimum of 40 games played in a season. Most Valuable Players (MVP) were correctly predicted by [61].

Advanced statistics were used by Pantzalis et al. [62] for team and players performance prediction in game of football. Defenders, even though they are not necessarily more significant in team strategy. So, in terms of player evaluation, this study [62] seeks to find adequate talents and characteristics that create successful defenders. For team performance prediction [62] used data from four european national footbal leagues. For every team using more than 40 features, they [62] predicted whether a team will have better or worse season as compared to last year. Previous season was used for training whereas for testing final season was used. Results showed that random forests achieved 70% of accuracy and standard deviation was less than 10%. They [62] also investigated whihch factors are affecting ratings of defender. For the said purpose they selected 59 central defenders who palyed minimum of 10 games in English Premier League in 2016-2017 season. The resluts showed that classif defensive actions and few attacking skills have impact on rating of central defenders.

2.3 Match Outcome Prediction in Sports

The fuzzy rule-based system (FRBS) is proposed by Trawinski [63] for the prediction of the basketball match outcome. Feature selection was applied for the selection of best features and various fuzzy models were used for the prediction of match outcome. A model for college basketball was proposed by Ruiz et al. [64] that combined a simple soccer model and poisson factorization. The simple soccer model identifies each team by its attack and defence coefficients whereas the poisson factorization considers the elements of the matrix that are independent of the poisson random variables. For match outcome prediction in basketball an integrated model called Hybrid Support Vector Machine and Decision Tree (HSVMDT) is proposed by Pai et al. [65]. Feature selection was used to select the best features (7 features were selected out of 17). HSVMDT was tested on both selected features and on all 17 features. HSVMDT achieved 82.25% with feature selection and without feature selection, the accuracy was 67%. The decision tree generates many rules that can cause confusion for decision makers. Rules pruning was used to limit the number of rules. For measuring the quality of decision rules, the sum of testing accuracy and coverage index was used. The results showed that decision rules have better quality after pruning. The rules generated by said model aim to help coaches to identify which factors are affecting match outcome. Analysis based on classification and regression tree was performed by Miguel et al. [66] to find best predictor in order to classify teams as winning or losing teams. Their analysis showed that in fast paced games the importance of defensive rebounds is 100%, importance of free throws is 94.7%, assists 86.1% and importance of fouls is 55.9%. On the other hand the importance of variables in slow paced games are: free throws is 100%, defensive rebounds 82.3%, fouls 68.4%, assists 66.9%, 2-points 62.2% and importance of 3-point field goals is 62.1%. Data driven and data envelopment analysis based techniques were used by Li et al. [67] for predicting the performance of sports team. They used multivariate logistic regression to find relationship between winning probability and match outcome. Their study suggests that team coaches and managers should focus on communication and cooperation of team. Various machine learning models are used by Thabtah et al. [68] for the prediction of match outcome in basketball. They examined the strength of various features for match outcome prediction. The defensive rebound was observed to be the most suitable feature for match outcome prediction. Discrete-time and finite-state Markov chain has been used by Shi and Song [69] to predict the outcome of the match when the game is in progress. The aim of the said model is to find the difference between the home team and the visiting team score at

some time point. The predictions for the ongoing match can be made on the current score of the team instead of past data.

Classification and regression models were used by Valero [45] for prediction of match outcome in baseball. They used past data in their study. Data Mining techniques were used by Tolbert et al. [46] for championship winner. They used various game statistics as attributes for winner prediction. The features they used are: runs scored, stolen bases, batting average, on base percentage, slugging percentage, team wins, team losses, earned runs average, save percentage, strikeouts per nine innings, opponent batting average, walk plus hits per inning pitched, fielding independent pitching, double plays turned, fielding percentage and win-above replacement. Machine learning methods were used by Hamilton et al. [47] to predict pitch type. They improved performance by applying various classification models and feature selection approach. The features they used for prediction of pitch type are: percentage of fast balls thrown in previous inning, previous pitch velocity, strike result percentage. A single measurement criteria used by Yang et al. [48] is based on team strength, batting ability of team and starting pitchers. Home field advantage variable and relative strength variable were used to propose a two stage bayesian model for major league baseball match outcome prediction. Machine Learning techniques were used by Donaker [49] for prediction of individual games outcome and identifying important elements of teams. Total of 25 features were used for prediction. Logistic regression, Naive Bayes, support vector machines and ensemble classifiers were used as classification models. Random forests and XGBoost classifiers were used along with historical baseball data by Elfrink [50]. XGBoost classifier achieved highest accuracy. Experiments carried out by Jia et al. [51] concludes that baseball data is noisy and using baseball statistics to predict match outcome are not enough. Their accuracy of predicting match outcome is 60%. The accuracy was increase upto 65% when later portion of season data was used. Gated Bayesian Networks were used by Bendtsen [52] to model careers of baseball players with focus on identification of regimes in data.

Various machine learning and deep learning methods have been used and compared for prediction of match outcome in Major League Baseball Huang et al. [70]. Data of 30 teams from 2019 season was acquired for match outcome prediction. Feature selection and normalization of data was done prior to applying the prediction models. Results of match outcome prediction were compared with and without feature selection [70]. Results showed that artificial neural network achieved highest accuracy of 93.91% as compared to one-dimensional convolutional neural network and support vector machine models. The Markov process ap-

proach along with runner advancement model were used to estimate the anticipated runs in an MLB game for teams depending on their batting order and pitcher [71]. Data used by Chang [71] consists of 70 matches of Major League Baseball for time period of 15 september 2018 to 30 september 2018. Machine learning techniques were used by Harikrishnan et al. [72] for match result prediction in games of Korean Baseball Organization. The dataset used by [72] consists of various statistics of players that are related to performance of players. Neural networks, support vector machine and decision tree models were used for prediction of results [72]. Yaseen et al. [73] used Support vector classifier and logistic regression models for prediction of playoff for year 2019. Major League Baseball data for time period of 2015 to 2019 was used for the prediction of next match outcome by Li et al. [74]. The results showed that support vector machine achieved highest accuracy of 65.75% as compared to one-dimensional convolutional neural network and artificial neural network.

Impact of passing network on match outcome prediction in football is examined by Ievoli et al. [75]. Indicators of passing networks have been used for estimating the probability of winning in the game of football [75]. Dataset used by [75] consists of 96 games for 2016-2017 Group of European Football Association. Based on performance factors binomial logistic regression achieved better results [75]. Comparison of statistical and machine learning methods have been made for match outcome prediction in football by Beal et al. [76]. Data used by [76] consists of six season data from English Premier League and match previews from Guardian newspaper. The experimental result showed that models presented in [76] achieved accuracy of 63.18% which was 6.9% more than the accuracy of statistical methods. Comparison of performance between team and players rating is made by Arntzen et al. [77] while predicting match outcome in football. Players were rated on the basis of plus-minus rating whereas rating of teams were calculated by using Elo rating system. Experimental results showed that there is no such difference in prediction results while using competing risk models and ordered logit regression model. Traditional match outcome predictions methods use match statistics like red card, yellow cards, shot on targets etc whereas Pipatchatwala et al. [78] presented fusion based method which does not use any game statistic for match outcome prediction in football. The fusion based model is based on players ratings from video games. Two fusion models hierarchical and ensemble models were proposed by Pipatchatwala et al [78]. Data of season 2010-2011 to 2014-2015 from English Premier League have been used for experiment purpose. The proposed model achieved accuracy of 56%. Various machine learning classifiers and features have been investigated by Haruna et al. [79] to find a better machine learning classifier and better features for the prediction

of match outcome in English Premier League. Experimental results showed that goal difference, first twenty two players from both teams, away team and home team and K-NN classification model achieved highest accuracy of 83.95% [79]. Score based match outcome methods in soccer have been investigated by Hubavcek et al. [80]. Statistical methods based on Weibull and Poisson distribution and various ranking methods are closely discussed by [80]. Experimental results showed that overall performances and individual prediction are almost same on the tested methods. Neural network models have been used by Guan et al. [81] for prediction of football match outcome prediction. Comparative analysis based on Poisson model is carried out by Maozad et al. [82] for match outcome prediction in football. Season 2015-2016 to 2020-2021 of English Premier League have been used as training data whereas season 2021-2022 is used as test data. The results showed that Dixon-Colse model performed better as compare to other methods. A deep learning LSTM model is used by Nivetha et al. [83] for win, loss and tie in games of English Premier League. Various machine learning algorithms were used for prediction of football match results by Rodrigues et al. [84] while using different statistics of players as attributes for prediction. Data of four seasons from 2013-2014 to 2016-2017 have been used for training purpose whereas the data of season 2018-2019 is used as test data [84]. SVM achieved highest accuracy of 61.32% while using total of 18 features [84].

Match outcome prediction in cricket while the game is in progress is presented by Goel et al. [85]. Approach adopted by [85] is based on relative strength of both teams and dynamically changing context of the match. To train machine learning models data of 2008 to 2018 matches is used whereas 59 matches from 2019 season are used as test data. Experimental results showed that machine learning models achieved accuracy of 76%. Awan et al. used [86] Linear regression for finding match outcome by using both scikit learn and big data SparkML framework. Linear regression scored 96% of accuracy by using Spark machine learning framework [86] Prediction of score after 50 overs based on current situation is presented by Kamble et al. [87]. Winning percentage of both teams before the start of match is also discussed by Kamble et al. [87]. Neural network and CART classifiers have been used by Kumar [88] for match outcome prediction in Indian Premier League. The data used by Kumar [88] consists of Indian Premier League 334 matches of year 2016 to 2018. Variety of supervised learning models have been implemented and compared by Priya et al. [89] for predicting match winner of IPL. Dataset was obtained from popular data science competition platform Kaggle. Results showed that Random Forest models achieved better accuracy of 74%. Relation between independent factors is investigated by using logistic

regression by Sarangi et al. [90]. Analysis of the independent factors revealed that bowler economy and fielding dismissals strongly affect the outcome of the match [90]. Comparison of various ensemble and non-ensemble classification methods is presented by Pramanik et al. [91] for match outcome prediction in T20 cricket matches of Bangladesh Premier League. Experimental results showed that KNN and Gradient Boosting methods presented better performance for match outcome prediction.

2.4 Rising Star Prediction in Academic Network

ScholarRank method proposed by Zhang et al. [92] combine statistical factors and influence calculation methods in heterogenous academics networks. ScholarRank method is based on citations count, mutual influence and mutual reinforce among various entities of the heterogeneous network. ScholarRanks method overcome limitations of previous studies where reinforce factor is ignored when evaluating rising stars and used only mutual influence among the authors. To assess the rising stars impact, ScholarRank is applied on data of American Physical Society by considering the authors who started their career in 1993. Effectiveness of ScholarRanks was compared with StarRank [2] and CocaRank [93] methods. Comparison of average count of citations of top 10 rising stars for the year 2013 showed that ScholarRank had better citation count mean as compared to CocaRank and StarRank methods.

The concept of rising star has been used by Zhu et al. [94] for finding rising stars in the field of technology. They identified rising stars and their attributes and then used those attributes to build the profiles of inventors. Framework for searching rising technology stars have been proposed by Zhu et al. [94], which is based on profiles of the inventors. The profiles on the inventors are constructed on the basis of technological performance, sociability and innovation caliber. They [94] also introduced four types of rising technology stars. To identify which group had most of the rising technology stars a method called nth percentile is also presented. Empirical study of inventors in the field of 3D printing was conducted by [94] for the validation of their proposed methodology. Inventors in the field of 3D printing from the period 2007-2016 were selected and co-inventors with single patent were ignored which resulted in total of 8821 inventors. Entropy weight method is used by [94] to identify potential features in inventors profiles. K-means clustering was used to divide inventor profiles into various clusters.

In geo-social networks the concept of rising stars is addressed by Ma et al. [95]. A novel method called FS-ELM which is based on extreme learning machine was proposed by [95]

for the evaluation of stars of the future. The said method comprises of three parts. The first part builds features by using behavior of users and social topology. Supervised information is extracted in second part by searching topic experts, topic experts are considered as rising stars. The third part finds whether a user is rising star or not at a specific time by using classification. The data used by [95] was obtained by using Foursquare API. The data collected contain 76503 users and 153137 connections. Checkin for each user is also recorded. The results showed that the proposed method FS-ELM achieved better results as compared to existing models.

Classification of authors based on transfer learning is presented by Abbasi et al. [96]. The goal of [96] is to adapt new tasks in the target area in order to extract outer information from the source domain. Author classification based on transfer learning consists of three steps. In first step the structure of features is constructed from two networks. In second step features of target network are reconstructed by utilizing similar signature subgraph using both networks. In third step for classifier learning in target network to categorize and rebuild features of the authors. The results [96] showed that proposed methods have many advantages over similarity based method.

Finding rising stars is a challenging issue if they are working in multiple domains. A method called Hot Topics Rising Star Rank (HTRS-Rank) is proposed by Daud et al. [97]. HTRS-Rank method overcomes the limitation of previous rising star finding methods where they only focused on co-author networks and ignored the textual content. HTRS-Rank method identifies junior authors who participated in hot topics at the start of their career and based on their engagement in hot topic publications, these authors are ranked. HTRS-Rank method extracts titles of research articles to get hot topics. Latent Dirichlet Allocation [98] is used for the construction of topics by finding probabilities of words for each cluster. To validate the proposed method data from Aminer was obtained for the time period of 2005-2009. Results showed that the proposed method achieved better results as compared to the baseline methods. Results showed that authors discovered through HTRS-Rank method were participants of many active research areas.

Rising star approach has been used by Daud et al. [99] for finding rising business managers in the field of telecommunications. Instead of using attributes of managers, the study of Daud et al. [99] considered attributes of the co-business managers for prediction of rising business managers. Attributes of co-business managers have been designed by [99] which are fed to machine learning classifiers for rising business manager prediction. Various types of at-

tributes designed by [99] are key performance indicators of co-business managers and senior business managers. The data was acquired from one of the top telecom company of Pakistan for the time period of 2014 to 2015. For classificatin purpose NB,BN, NN, and SVM have been used by [99]. The results showed that SVM and NN achieved better results.

Prediction of rising stars in yelp review network has been introduced by Nawaz and Malik [100]. Variety of features are proposed by [100], these features belongs to these types of categories: meta data features, RFA (Recency-Frequency-Activity) features and temporal features. Data used for rising star prediction in reviewer nework is obtained from yelp reveiw platform. First already available dataset is obtained that has iformation about reviews and reviewers for duration of 2004 to 2017. After necessary preprocessing 800 top reviewers are acquired and 400 top of them are labelled as rising stars and remaining 400 are considered as not rising stars [100]. The experimental results showed that gradient boosted and decision tree models performed well. F-measure of 84% is achieved by meta data attributes. Ranking of top 10 reviewers (rising stars) is also presented by [100].

A novel method called RiseNet is proposed by Yang et al. [101] for rising star prediction. RiseNet is based on items dynamic features and initial dissemination of user interest. RiseNet is validated on chinese online shopping platform Taobao. They [101] defined rising star as an item that is not rated in top 3% in initial four weeks but afterwards rises to be ranked in the top 1% in the next two weeks. Three datasets used by Yang et al. [101] have been provided by Taobao platform. The three datasets are: Taocode diffusion data, Taobao purchase data and basic information of taobao items. Taocode diffusion data consists of 68,888,906 records of taocode diffusion. Taobao purchase data consists of 204,398,128 purchase records. Basic information of Taobao items consists of 31,093,128 items category, name and price. Exploratory analysis revealed that there is close correlation between Taocode diffusion graph and identification of rising star from node and graph level. Experimental results showed that the proposed RiseNet model achived better performance.

2.5 Rising Star Prediction in Sports

The concept of rising star prediction using machine learning has been applied by Ahmad et al. [14]. They used the concept of co-player of rising star players. They considered those players as co-players of rising star that have appeared with rising stars in some common time span. They used CART (Classification and Regression Treee) MEMM (Maximum Entropy Markov Model), SVM (Support Vector Machines), BN (Bayesian Networks) and NB (Naive

Bayes). They used co-players, team and opponent team features for prediction of rising stars. They used data from 2006 to 2013 for rising star prediction. For labeling rising stars they used weighted average which is based on number of runs, average and strike rate. Weight threshold of 33.33 was used as weight while calculating weighted average.

Said research [14] has certain limitations. Co-player selection on basis of time span may select false co-players. The concept of co-players they used may consider a player as co-player who appeared in some common time span but might possible that he never appeared in a game with rising star players.

In sports ranking of players depends on their individual performance whereas rising stars are predicted by using their co-player features.

Ranking players does not tell about future ranking of players whereas rising stars are the players that have the tendency to make their place in high ranked players in near future.

Since ranking of players are based on player's individual performance so this ranking can not be used to select best combination of players. Co-players of rising stars can be used to select players that perform well while they play together.

Table 2.1: Literature Summary

S.No	Research Objective	Ref
1	Assessing the abilities of NBA Players.	[39]
2	Finding the impact player on chance of team winning.	[40]
3	Ranking Players of NBA.	[41]
4	The impact of individual status on network linkages among NBA teammates	[42]
5	Elite basketball performance profiles and opponent involvement during game play.	[43]
6	Scorekeeper bias adjustment in NBA box scores.	[44]
7	Comparison of data mining techniques for match result prediction in MLB.	[45]
8	Using data mining for championship winner prediction in MLB.	[46]
9	Predicting Baseball pitch through machine learning methods.	[47]
10	MLB winner prediction using two stage bayesian model.	[48]
11	MLB analysis and prediction using machine learning.	[49]

12	MLB outcome prediction using machine learning.	[50]
13	Forecasting MLB season.	[51]
14	Representation of baseball players regimes by utilizing career data.	[52]
15	Investigate the impact of time spent in college and outcome in NBA	[53]
16	Finding the impact of peer presence on session rating.	[54]
17	Finding how performance of individual is affected by its peers.	[55]
18	Clustering of ball possession time based on role of player.	[57]
19	Analysis of Indian cricket team performance.	[58]
20	Using data mining techniques to analyse cricket team performance.	[59]
21	To find impact of injuries on team and player performance in basketball.	[60]
22	Evaluating performance of team and players in basketball.	[61]
23	Predicting player performance in football.	[62]
24	Predicting result of basketball games using fuzzy classification system.	[63]
25	Prediction of game outcome in basketball using generative model.	[64]
26	Using SVM and decision tree models to analyze basketball games.	[65]
27	Classifying winning and losing in basketball by using regression tree.	[66]
28	A data-driven prediction technique for sports team performance in NBA	[67]
29	Predicting NBA game results using feature analysis and machine learning.	[68]
30	Prediction of NBA games using finite state Markov chain.	[69]
31	MLB games outcome prediction using machine learning and deep learning.	[70]
32	Proposed predictive model for match outcome prediction in MLB.	[71]
33	Predictive analysis of fantasy games using machine learning.	[72]
34	MLB playoff prediction using multimodal machine learning.	[73]
35	Identification of features for prediction of MLB games outcome.	[74]
36	Prediction of football match outcome using passing network indicators.	[75]
37	Football match outcome prediction through human experts and machine learning.	[76]
38	Using team and player ratings for prediction of football match outcome.	[77]
39	Proposed fusion based supervised model for football match outcome prediction.	[78]

40	Using machine learning for football match outcome prediction.	[79]
41	Experimental review of forty years score based match outcome prediction in soccer.	[80]
42	Using neural network to optimize prediction of match outcome in football.	[81]
43	Comparison of poisson models for football match outcome prediction.	[82]
44	Prediction of match outcome in football using deep learning.	[83]
45	Applying machine learning techniques to predict football match outcome.	[84]
46	Predicting match outcome on ball by ball in second innings in cricket.	[85]
47	Predicting team score using regression and big data.	[86]
48	Predicting team score before and during cricket match by using machine learning.	[87]
49	Using neural network and CART for match prediction in IPL cricket match.	[88]
50	Comparison of machine learning models for match outcome prediction in cricket.	[89]
51	Examine attributes of winning in one day cricket match.	[90]
52	Comparison of classification models for match prediction in T20 cricket.	[91]
53	Proposed ScholarRank method for rising star in heterogenous network.	[92]
54	Improved PubRank method is presented for rising stars.	[2]
55	Proposed Cocarank method for rising stars in academic networks.	[93]
56	Finding rising stars in field of technology using unsupervised method.	[94]
57	Find rising stars in geo social network by using extreme learning approach.	[95]
58	Classification of authors and rising star prediction by using transfer learning	[96]
59	Proposed HTRS-Rank method to find rising stars by hot topic detection.	[97]
60	Prediction of rising business managers using supervised classification.	[99]
61	Predicting rising stars in yelp reviewer platform using binary classification.	[100]
62	RiseNet method that is based on diffusion process of user interest.	[101]
63	Presented co-player features for rising star prediction in cricket.	[14]

2.6 Chapter Summary

This Chapter covered the related work in sports. The main research discussed in literature is related to ranking of players in sports, performance analysis of players, match outcome prediction in sports, rising star prediction in academic networks and rising star prediction in sports. Ranking of players used different metrics to rank players in basketball and baseball. Performance analysis of player consider various factors that affect the performance of players during the game. Match outcome prediction in team sports widely used statistical and machine learning methods to predict outcome of the match. Various machine learning classifiers have been used for match outcome prediction and variety of features have been identified in the literature for prediction of match outcome in team sports. Rising star prediction in academic network identify potential researchers by identifying various factors and various machine learning models are used for rising star prediction in academic networks. Rising star prediction in sports used supervised machine learning models and features of co-players to predict risig stars in game of cricket.

Chapter 3

Rising Star Prediction in Basketball

D
TH-25938

3 Rising Star Prediction in Basketball

Detailed discussion of datasets, features and analysis of experimental results presented in this chapter.

3.1 Basic Concepts of Basketball

In this section, we presented different terminologies that are related to the game of basketball.

3.1.1 Court

The basketball game is played on a rectangular floor and there is a hoop at each end. A mid line on court divides it into two sections. The basketball court has a center circle and a three-point line. In the center circle, only two players are allowed to enter prior to tipoff. Two point and three-point areas are separated by the three-point line.

3.1.2 Team Structure

Each team consists of five players. Five players of each team are assigned different positions on the court. The position assigned to the players are point guard, shooting guard, small forward, power forward and center.

3.1.3 Basic Rules

The following are the basic rules for the basketball game.

1. Objective of each team is to shot the ball in the basket of the opposing team.
2. Each team with a maximum of five players on the court.
3. The game consists of four periods whereas each period is of 12 minutes, so the overall game is of 48 minutes. 5 minutes overtime is played in case of tie until the game end without a tie.
4. Two points are scored when the ball is put in the basket from the inside three-point arc.
5. Three points are scored when the ball is put in the basket from the beyond three-point arc.

6. One point (also called free throw) is scored when the ball is put in the basket from the free-throw line.
7. The ball may be passed to another player or may be dribbled from one point to another while running. A player cannot dribble again if once he stopped dribbling. When the team is in possession of the ball and has crossed the middle line on the court, then the team cannot cross back the mid line on the court.
8. A team having possession of the ball have a maximum of 24 seconds to make a shot.
9. Illegal contact with opponent player results in a personal foul.

3.2 Player Statistics

Basic game statistics of a basketball player are given in table 3.1.

Table 3.1: Details of Basic Basketball Statistics

Stat	Description
Field Goals(FG)	These are the shots by a player that goes through the basket from above.
Average Field Goals(avg_FG)	Dividing Field Goals by total number of games.
Field Goal Attempts(FGA)	Number of attempts by a player to score Field Goal.
Average Field Goal Attempts(avg_FGA)	Dividing Field Goal Attempts on total number of games.
Field Goals Percent(FGper)	FGper can be obtained through dividing Field Goals by Field Goal Attempts.
Three Points(3PT)	Three points are awarded to a payer when he shots the ball from long distance and ball goes in the basket.
Average Three Points(avg_3PT)	Average of three points can be obtained by dividing player's total three points on number of games played.
Three Point Attempts(3PTA)	This is the number of attempts to score Three Points.

Average Three Point Attempts(avg_3PTA)	Average 3PTA can be calculated by dividing total number of Three Point Attempts on total number of games played by a player.
Three Point Percent(3PTper)	It is the measurement of long distance shooting ability of a player. 3PTper can be calculated by dividing Three points on Three Point Attempts.
Free Throws(FT)	These are shots made from free throw line. One point is awarded for a Free Throw.
Average Free Throws(avg_FT)	Average Free throw is obtained by dividing player's total number of Free Throws on number of games played by a player.
Free Throw Attempts(FTA)	Free Throw Attempts are the number of time a player attempted to make a free throw.
Average Free Throw Attempts(avg_FTA)	It is simply the division of total number of Free Throw Attempts by total number of games played by a player.
Free Throw Percent(FTper)	FTper measure the Free Throw ability of a player. FTper can be calculated by dividing Free Throws on Free Throw Attempts.
Offensive Rebounds(OREB)	An offensive rebound occur when the player that recovers the missed shot is on similar team of the player who shot the ball.
Average Offensive Rebounds(avg_OREB)	Average Offensive Dividing total number of Offensive Rebound on total number of games played by a player.
Defensive Rebounds(DREB)	A defensive rebound happens with the player that recovers the missed shot is on the opposing team as the player that shot the ball.

Average Defensive Rebounds(avg_DREB)	Average Defensive Rebounds are obtained by dividing total number of Defensive Rebounds of a player on total number of games played by a player.
Rebounds(REB)	Player Rebounds are the sum of player's Offensive and Defensive Rebounds.
Average Rebounds(avg_REB)	Average Rebounds of a player are obtained by dividing player's total Rebound on player's total number of games.
Assists(AST)	Assist is awarded to a player who pass the ball to a player of same team who shot the ball in basket.
Average Assists(avg_AST)	Average Assists of a player is obtained by dividing total number of Assists on total number of games played by a player.
Turnover(TOV)	A player is charged with a turnover if he lose possession of the ball to the opposing team before a shot is attempted.
Average Turnover(avg_TOV)	Average Turover is obtained by dividing total number of Turnovers on total number of games played by a player.
Blocks(BLK)	A block occurs when a defensive player diverts a shot attempt of an offensive player.
Average Blocks(avg_BLK)	Average Blocks is obtained by dividing total number of Block on total number of games played by a player.
Personal Fouls(PF)	A personal foul is any violation of the rules of the game that involves personal contact with an opposing player.
Average Personal Fouls(avg_PF)	Average Personal Fouls are obtained by dividing player's total number of personal fouls on player's total number of games.

Points(PTS)	Points are scored when a player puts the ball through the basket.
Average Points(avg_PTS)	Average points are obtained by dividing player's total number of Points on total number of games played by a player.

3.3 Basketball Datasets

Flowchart in Fig.3.1 show the process of formation of datasets. To find rising stars in basketball we first collected five seasons (2004-2005 to 2008-2009) basketball game data from a sports website¹. Necessary data preprocessing was done using MySQL queries and Python Pandas library. The initial data statistics are:

Seasons: 2004-2005, 2005-2006, 2006-2007, 2007-2008, 2008-2009.

Teams: 30

Games: 6150

Players: 727

We selected the players who played at least 300 games in 5 seasons career. We get 100 players out of 727 who played more than 300 games in their 5 seasons career. The top 50 players with the highest average efficiency score (avg_EFF) are labeled as Rising Stars and the remaining 50 with lowest avg_EFF are labeled as Not Rising Stars. The data was shuffled to generate randomness in the data. In basketball, a team wins having maximum points (PTS) at the end of the game. So, a player who scores more points will contribute the team more to the win, but there are other factors too, like fouls, blocks and rebounds etc. which affect team performance. The NBA efficiency formula captures these factors. The formula is

$$EFF = (PTS + REB + AST + STL + BLK - FGA - FTA - TOV). \quad (3.1)$$

The average efficiency of a player is calculated by dividing player efficiency score by total games the player played.

$$avg_EFF = \frac{EFF}{Total\ Games\ Played} \quad (3.2)$$

¹www.espn.com/nba/

We used average efficiency score (avg_EFF) for labeling because it captures all factors related to the performance of a player. For the 100 players, we constructed three types of datasets. The purpose of building three different datasets is to investigate how the players of the same team, the opponent team and both the same and opponent teams are effective in the prediction of rising stars.

3.3.1 Dataset A

This dataset contains features of co-players of labeled players who played in the same team and the same game.

3.3.2 Dataset B

This dataset contains features of co-players of labeled players who played in the opponent team but in the same game.

3.3.3 Dataset C

This dataset contains features of both same and opponent team co-players of labeled players who played in the same game.

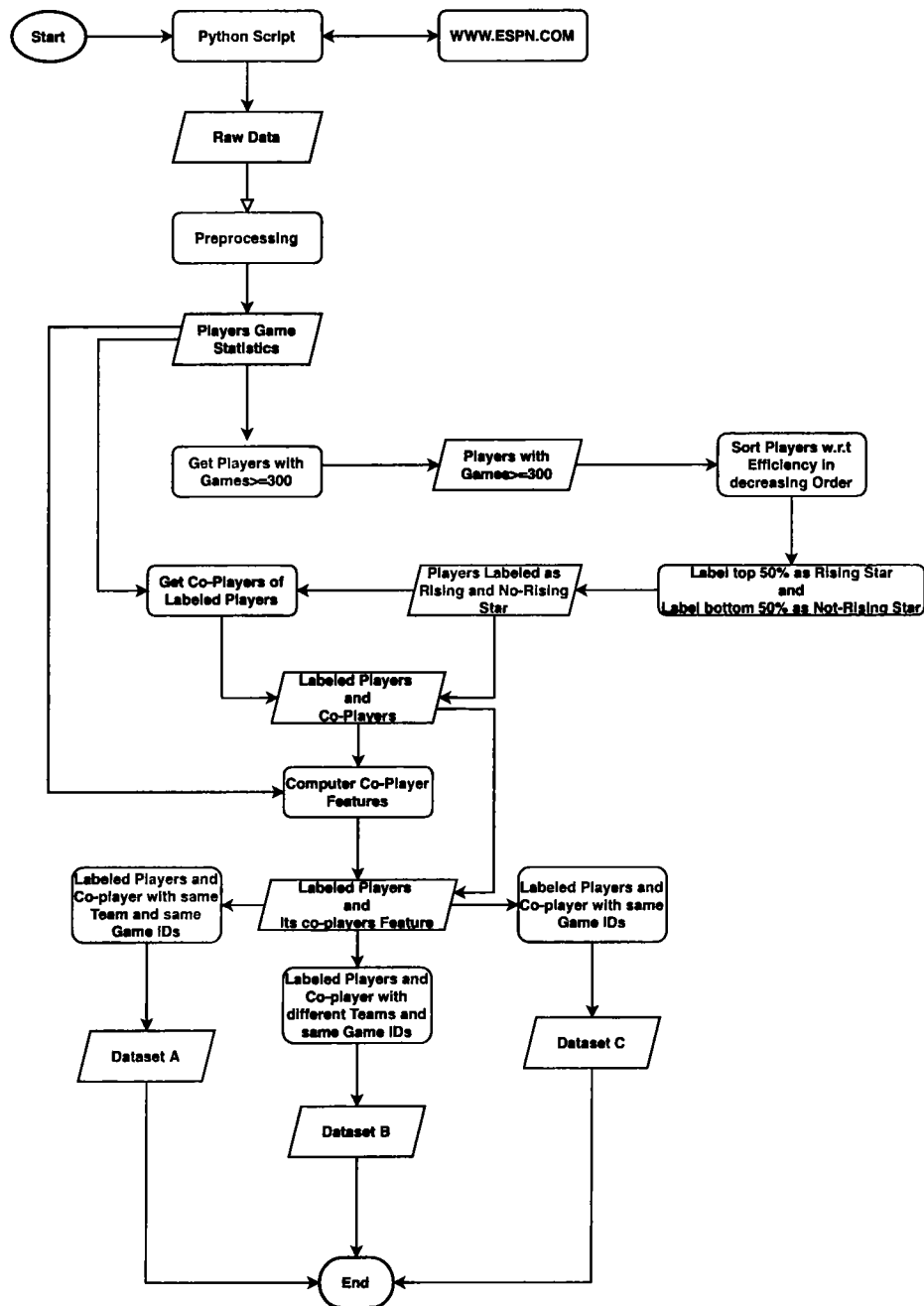


Figure 3.1: Flow Chart showing dataset acquisition process for basketball data.

3.4 Proposed Steps For Rising Star Prediction

Fig. 3.2 shows steps that are followed for prediction of rising star. These steps include co-player selection, feature engineering and implementation of machine learning methods for rising star prediction.

Algorithm 1 Co-Player Selection

```

1: Input: Players Game Statistics  $S = \{pname, gameid, s_1, s_2 \dots s_n\}$ ;
2: Output: Player and its Co_Players  $P = \{pname, co_1, co_2 \dots co_n\}$ ;
3:  $P = \{\}$ ;
4: for pname in S do
5:   if  $sum(gameid \geq 300)$  then
6:      $P \leftarrow pname$ 
7:   end if
8: end for
9: Reorder pname in P by Efficiency
10: Label top 50% as RS and remaining as NRS in P
11: for pname in P do
12:   for i,pname,gameid in S do
13:     if  $(gameid \in P == gameid \in S)$  then
14:        $Co_i \leftarrow pname \in S.$ 
15:        $P \leftarrow Co_i$ 
16:     end if
17:   end for

```

Algorithm 2 Computing Features

```

1: Input: Co-PlayerStatistics  $S = \{pname, gameid, s_1, s_2 \dots s_n\}$ ;
2: Output: Feature Matrix  $F = \{f_1, f_2 \dots f_n\}$ ;
3:  $Co\_F = \{\}$ ;
4:  $F = \{\}$ ;
5: for i,pname in S do
6:    $f\_Co_i \leftarrow average(s) \in S$ 
7:    $Co\_F \leftarrow f\_Co_i$ 
8: end for
9: for i,pname in P do
10:   $F \leftarrow F \cup average(f\_Co_i) \in Co\_F$ 
11: end for

```

Algorithm 1 shows steps for co-player selection and steps to compute features are shown by Algorithm 2. In worst case running time of Algorithm 1 is $O(n^2)$. For Algorithm 2 running time in worst case is $O(n)$.

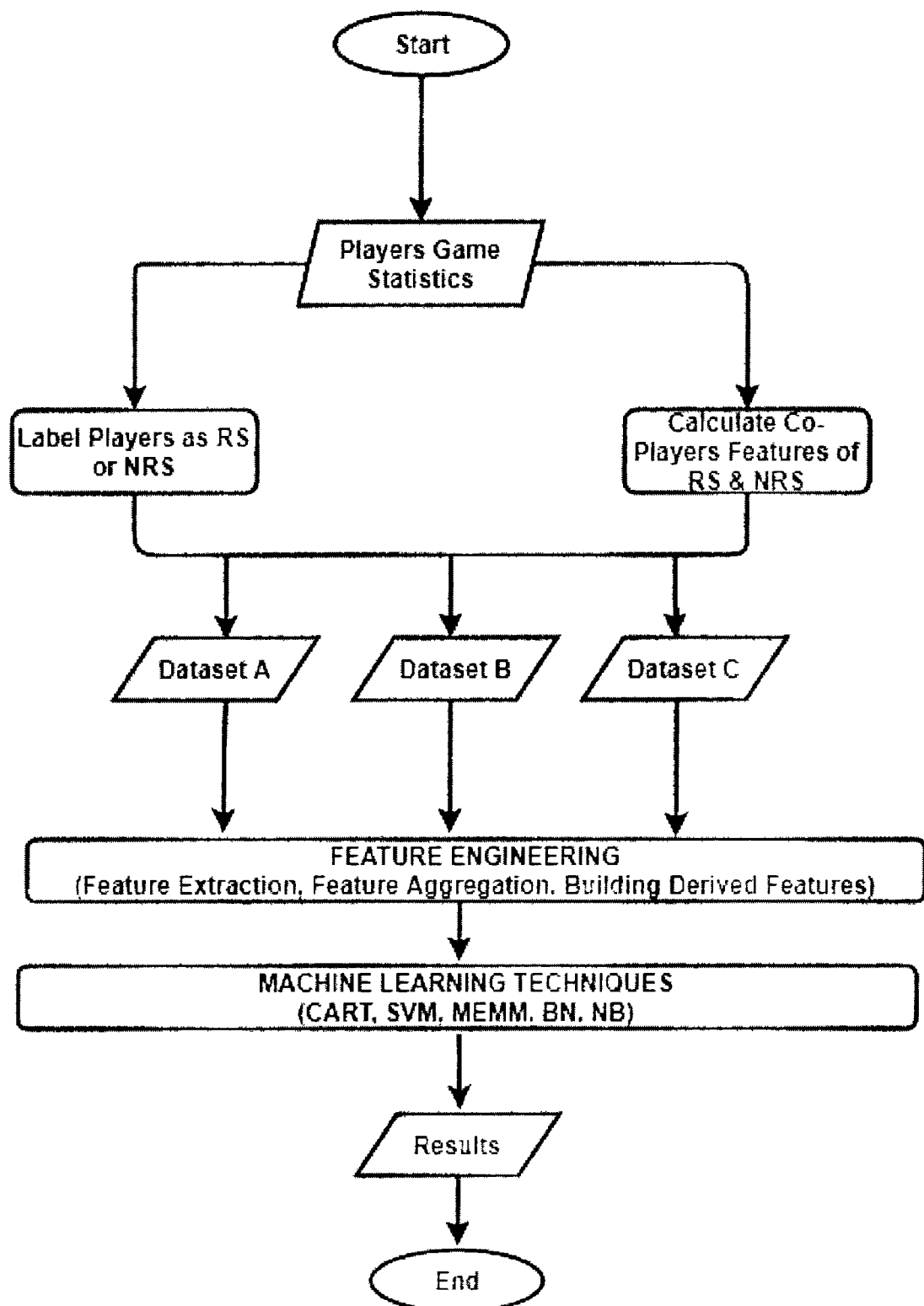


Figure 3.2: Rising Star Prediction Model

3.5 Machine Learning Techniques

CART, SVM, MEMM, BN and NB have been used for prediction of rising star. These techniques have been used for rising star prediction in cricket [14] and co-author network [4] and achieved satisfactory results in relevant domains.

3.5.1 Classification and Regression Trees (CART)

Classification and Regression Trees [102] is a decision tree based model. The working of CART model is based on three steps.

1. Constructing the maximum tree.
2. Choosing the right tree size.
3. Classifying the test data using the constructed tree.

CART splits the dataset based on similarity. Let us suppose two variables, marks and number of study hours. If there are 85% of students with maximum study hours in the first semester were successfully graduated then the tree will be split at the number of hours studied and it will become the top node in the tree. In the said example the 85% of data is pure. CART uses the Gini index to find to measure the impurity in the data.

3.5.2 Support Vector Machines (SVM)

Support Vector Machine [19] is used for both classification and regression tasks. Using SVM, data items are represented on the n-dimensional space. Once the data items are represented on the n-dimensional space, SVM then finds a hyperplane that can efficiently separate the two classes. For linear separable data, the key steps that SVM perform are

1. Plot the data items.
2. Find the margin and support vectors.
3. Find hyperplane with maximum margins.
4. Use the computed margin value to classify the new test data items.

3.5.3 Maximum Entropy Markov Model (MEMM)

Maximum Entropy Markov Model [103] is a sequence modeling algorithm. It extends the famous MEC (maximum entropy classifier)[104] with the feature that is; unknown parameters are assumed to be connected in a markov chain rather than independent to each other.

3.5.4 Bayesian Network (BN) Classifier

Bayesian network [105] uses a directed acyclic graph to represent variables and their conditional dependencies. Bayesian networks assume that features are codependent on each other. The bayesian network can capture codependency and influence of the features.

3.5.5 Naive Bayes (NB) Classifier

Working of naïve bayes classifier[15] is based on Bayesian theorem. Naïve bayes consider that there is no dependency between the features.

3.6 Co-player Selection Criteria

The concept of co-players is used by [14] for the prediction of rising stars in the game of cricket. They define the co-player as "Co-player is a comrade who belongs to the same or opponent team and has played matches during some common period". The limitation of their concept about co-player is that the players may play in some common period but they may not appear in the same games as well. For a player. Unlike [14] who used co-players, team and opposite team features, in this study we only used co-player features for rising star prediction. The reason for not using team and opposing team features is that team features are sum of player features. For example, Team Points are the sum of all player points and we are already considering co-player features. The other reason for not using team features is that total number of games played by team and total number of games played by player may be different. Suppose a player has played 30 games in a team and the total games played by the team are 70, now team feature will contain weights of 70 games. The limitation is that the players are weighted even for those games in which they never played. we consider co-players as the players who played with him in the same game. We further classify co-players into three types

1. Players of the same team and the same game.
2. Players of the opposite team and the same game.

3. Both same and opposite team players in the same game.

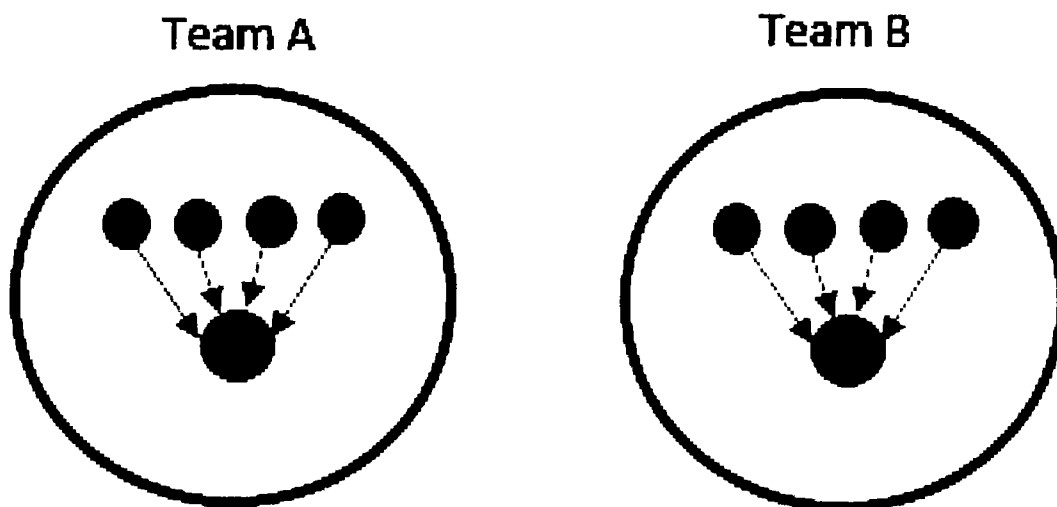


Figure 3.3: Players of same team and same game.

Red circle indicate the rising star player and blue circles represents the co-players. Arrows indicate the influence of co-players on rising star player.

Fig.3.3 represent players of two teams who played a game. In each team co-players of rising stars are those players that are playing in same game and belong to same team. The co-players are connected to rising star players of their own team only. Fig.3.4 represent players of two teams who played a game. In each team co-players of rising stars are those players that are playing in same game but belong to opposite team. The co-players are connected to rising star players of opposite team only. Fig.3.5 represent players of two teams who played a game. In each team co-players of rising stars are those players that are playing in same game and belong to both same and opposite team. The co-players are connected to rising star players of both same and opposite team. The purpose of considering these three types of co-players is to find which type of co-players are more useful in the prediction of risings stars in the game of basketball.

Fig.3.6, Fig.3.7 and Fig.3.8 show the relationship between rising stars and their co-players by using pearson correlation method. In these figures we can see how efficiency of rising star players is correlated to their co-players.

In Fig.3.6 we can see that there is a negative correlation between efficiency of rising star players and their co-players. We can clearly see that for the co-players who belong to same

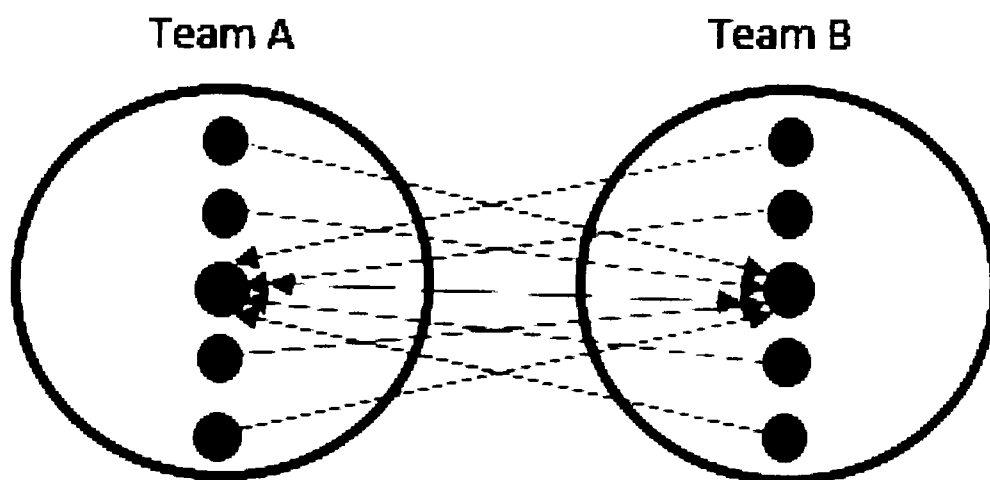


Figure 3.4: Players of opposite team and same game.

Red circle indicate the rising star player and blue circles represents the co-players. Arrows indicate the influence of co-players on rising star player.

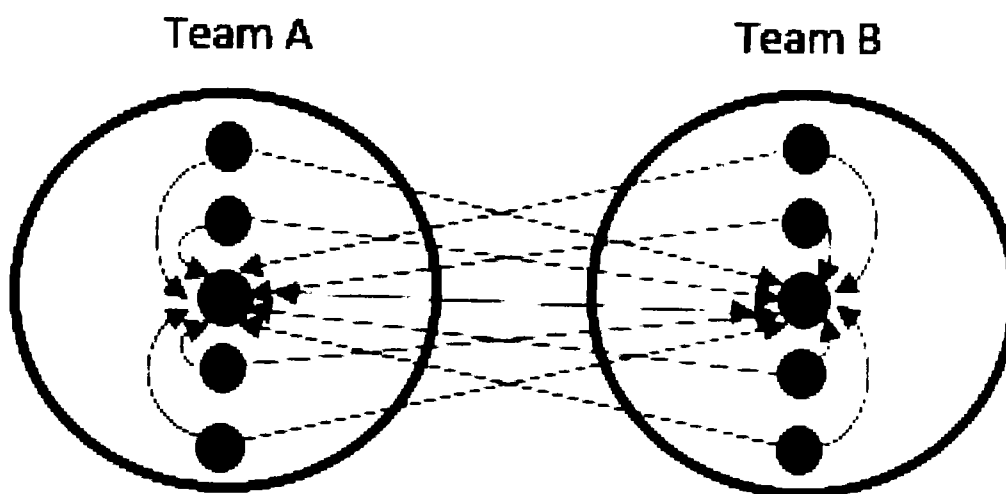


Figure 3.5: Players of same and opposite team in same game.

Red circle indicate the rising star player and blue circles represents the co-players. Arrows indicate the influence of co-players on rising star player.

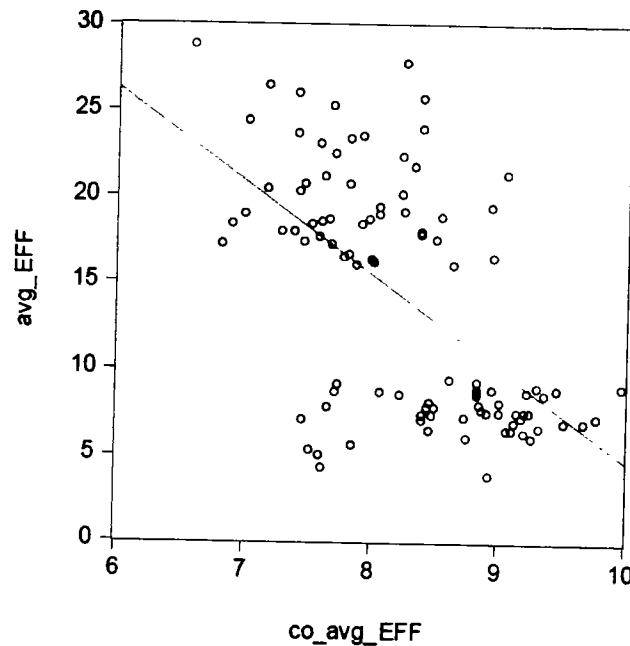


Figure 3.6: Relationship between co-players(same team) and rising star players. (x-axis represents average efficiency of co-players. y-axis represents average-efficiency of players labeled as rising and not-rising stars)

team of rising star players, the efficiency of rising star player tends to decrease when the efficiency of co-players increases. This means that with weak co-players rising star players have more chance to show his strength. On the other hand if co-players are performing well then rising star players have low chance to show his strength.

Fig.3.7 shows that there is positive correlation between efficiency of rising star players and their co-players. We can observe that for co-players who belong to opponent team of rising star players, the efficiency of rising star players tends to increase with the increase in the efficiency of co-players. This means that the more stronger the opponent team players then there is more chance for rising star players to show their strength.

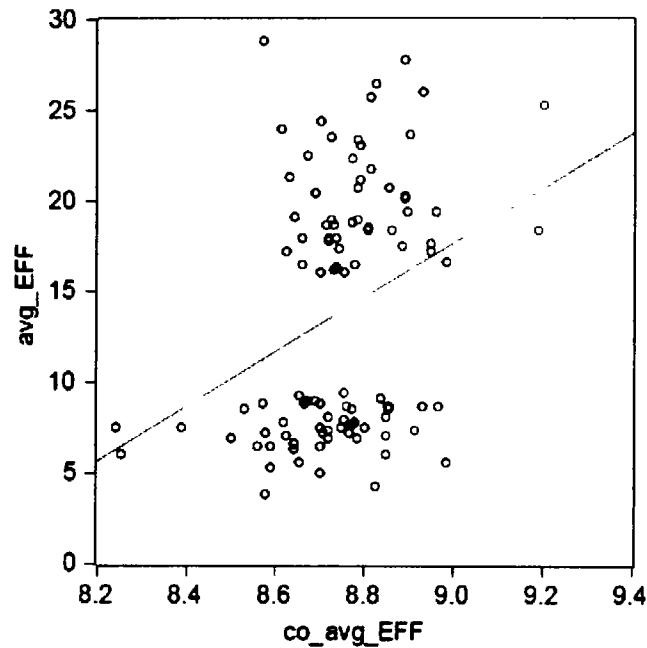


Figure 3.7: Relationship between co-players(opponent team) and rising star players. (x-axis represents average efficiency of co-players. y-axis represents average-efficiency of players labeled as rising and not-rising stars)

Fig.3.8 shows that there is a weak positive correlation between of rising star players and their co-players. It can be observed that for co-players who belong to both same and opponent teams of the rising star players, the efficiency of rising stars slightly tends to increase with the increase in efficiency of co-players.

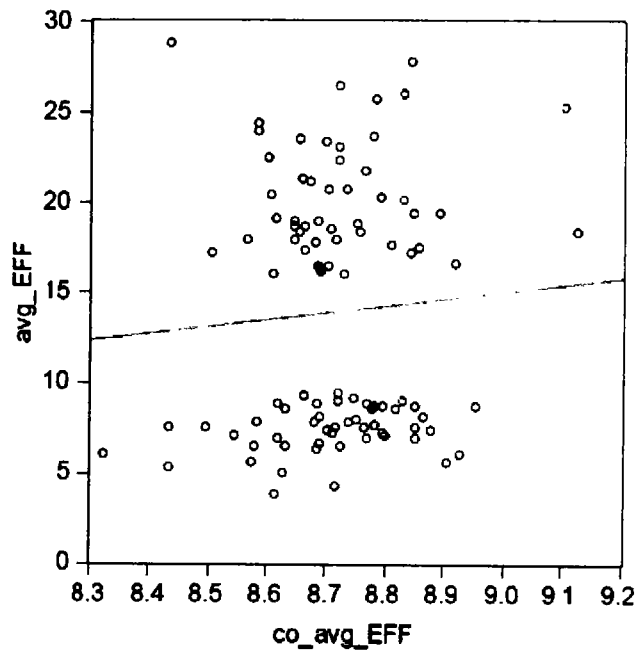


Figure 3.8: Relationship between co-players(both same and opponent team) and rising star players.

(x-axis represents average efficiency of co-players. y-axis represents average-efficiency of players labeled as rising and not-rising stars)

3.7 Features for Rising Star Prediction

Performance of machine learning models greatly depends upon the features supplied to them. To know the effectiveness of different features we classify features by type and size Fig.3.9 shows the pictorial representation of features categorization.

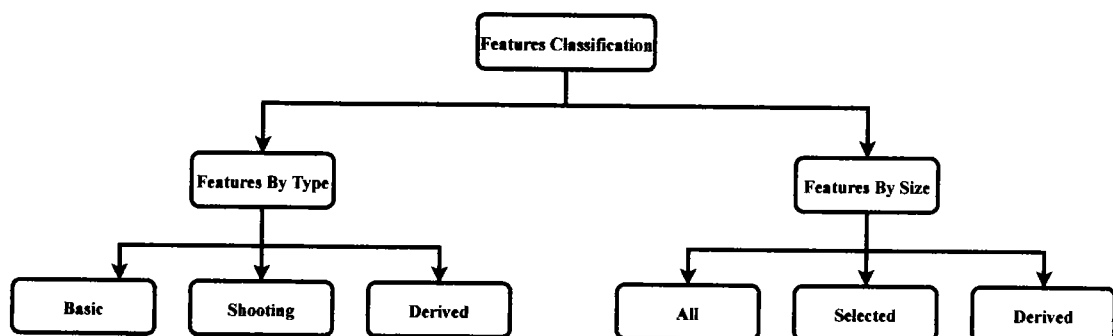


Figure 3.9: Features Categorization.

3.7.1 Features by Type

Based on type², we have classified features into basic, shooting and derived features.

3.7.1.1 Basic Features The basic features contain rebounds, assists, turnovers, blocks, fouls and points. Table.3.2 shows different number of basic features.

3.7.1.2 Shooting Features Fields goals, average field goals, field goal attempts, average field goal attempts, field goal percent, three points, average three points, three points attempt, average three point attempts, three points percent, free throw, average free throw, free throw attempts, average free throw attempts, and free throw percent. Table.3.2 shows different number of shooting features.

3.7.1.3 Derived Features These features are constructed from basic and shooting features. All of the derived features are related to player efficiency except the influence and average influence of a co-player. Derived features are listed in Table.3.2. Sec 3.8 give a detailed explanation and formulation of derived features.

3.7.2 Features By Size

The purpose of classifying features based on size is to know the impact of feature size on prediction results. Features classified by size are all features, selected features and derived features.

3.7.2.1 All Features All features are the combination of basic, shooting and derived features. All feature set consists of 47 features. Table.3.3 shows a list of all features

3.7.2.2 Selected Features Since the full feature set consists of 47 features. Correlation-based Feature Subset Selection technique [106] was used to acquire a subset of features that are highly correlated with the class while having low intercorrelation. The number of selected features for dataset A, dataset B and dataset C are 7, 9 and 3 respectively. Table.3.4 shows a list of selected features for each dataset.

²<https://www.basketball-reference.com/about/glossary.html>

3.7.2.2.1 Correlation Based Feature Subset Selection One of objective of machine learning is to identify the features that are more releavent to the class label. CFS(Correlation Based Feature Selection) method was proposed by [106] which utilizes correlation and heuristic search approach to identify important features. CFS assumes that the features have no dependency on each other. If the there is dependency among feautres, then CFS might igore important features when select subset of features. CFS is filter based method which ranks feautre subsets according to correlation based heuristic evaluation function. The func-ton find features that have high correlation with class label but have very low correlation or no correlation with each other. Features that are not releavent are ignored because that have low or no correlation with class label.

3.7.2.3 Derived Features The derived feature set consists of 16 different features. The size of the derived feature set is greater than the selected feature set and is smaller in size as compared to the size of all feature set. Table.3.5 shows a list of derived features.

Table 3.2: Features Classification By Type

S.No	Basic		Shooting		Derived	
	Feature	Description	Feature	Description	Feature	Description
1	OREB	Offensive Rebound	FG	Field Goal	co_inf	Co-player influence
2	avg_OREB	Average Offensive Rebound	avg_FG	Average Field Goal	avg_co_inf	Average Co-player influence
3	DREB	Defensive Rebound	FGA	Field Goal Attempt	EFF	Efficiency
4	avg_DREB	Average Defensive Rebound	avg_FGA	Average Field Goal Attempt	avg_EFF	Average Efficiency
5	REB	Rebounds	FGper	Field Goal Percentage	EFF_begg	Efficiency at beginning of Season
6	avg_REB	Average Rebound	3PT	Three Points	avg_EFF_begg	Average Efficiency at Beginning of Season
7	AST	Assists	avg_3PT	Average Three points	EFF_mid	Efficiency at Mid of Season
8	avg_AST	Average Assists	3PTA	Three Points Attempt	avg_EFF_mid	Average Efficiency at Mid of Season
9	BLK	Blocks	avg_3PTA	Average Three Points Attempt	EFF_end	Efficiency at End of Season
10	avg_BLK	Average Blocks	3PTper	Three Points Percentage	avg_EFF_end	Average Efficiency at End of Season
11	TOV	Turn Over	FT	Free Throws	Co_H-index	sum of H-indices to co-players
12	avg_TOV	Average Turn Overs	avg_FT	Average Free Throws	avg_Co_H-index	Average H-index of co-players
13	avg_PF	Personal Fouls	FTA	Free Throw Attempts	HGS	Hollinger Score
14	PF	Personal Fouls	FTper	Free Throw Percentage	avg_HGS	Average Hollinger Score
15	PTS	Points			Points_share	co-player points divided by team points
16	avg_PTS	Average Points			avg_Co_Point_share	Average point share of co-players

Table 3.3: All Features

S.No	Feature	Description	S.No	Feature	Description
1	co_inf	Co-player influence	33	avg_PTS	Average Points
2	avg_co_inf	Average Co-player influence	34	EFF	Efficiency
3	FG	Field Goals	35	avg_EFF	Average Efficiency
4	avg_FG	Average Field Goals	36	EFF_begg	Efficiency at Beginning of Season
5	FGA	Field Goal Attempts	37	avg_EFF_begg	Average Efficiency at Beginning of Season
6	avg_FGA	Average Field Goal Attempts	38	EFF_mid	Efficiency at Mid of Season
7	FGper	Field Goals Percentage	39	avg_EFF_mid	Average Efficiency at Mid of Season
8	3PT	Three Points	40	EFF_end	Efficiency at End of Season
9	avg_3PT	Average Three Points	41	avg_EFF_end	Average Efficiency at End of Season
10	3PTA	Three Point Attempts	42	Co_H-index	sum of H-indices of co-players
11	avg_3PTA	Average Three Point Attempts	43	avg_Co_H-index	Average H-index of co-players
12	3PTper	Three Points Percentage	44	HGS	Hollinger Score
13	FT	Free Throws	45	avg_HGS	Average Hollinger Score
14	avg_FT	Average Free Throws	46	Points_share	Co-player Points divided by Team Points
15	FTA	Free Throw Attempts	47	avg_Points_share	Average Points Share of Co-players
16	avg_FTA	Average Free Throw Attempts			

Table 3.4: Selected Features

S.No	Dataset A	Description	S.No	Dataset B	Description	S.No	Dataset C	Description
1	FTA	Free Throw Attempts	1	FG	Field Goals	1	FT	Free Throws
2	FTper	Free Throw Percent	2	3PTA	3-Points Attempt	2	EFF_begg	Efficiency at Beginning of Season
3	BLK	Blocks	3	3PTper	3-Points Percent	3	HGS	Hollinger Score
4	avg_EFF_begg	Average Efficiency at Beginning of Season	4	FTA	Free Throw Attempts	-	-	-
5	avg_EFF_mid	Average Efficiency at Mid of Season	5	FTper	Free Throw Percent	-	-	-
6	avg_EFF_end	Average Efficiency	6	DREB	Defensive Rebounds	-	-	-
7	avg_HGS	Average Hollinger Score	7	BLK	Blocks	-	-	-
-	-	-	8	avg_TOV	Average Turnover	-	-	-
-	-	-	9	avg_EFF_mid	Average Efficiency at Mid of Season	-	-	-

Table 3.5: Derived Features

S.No	Feature	Description	S.No	Feature	Description
1	co_inf	Co-Player Influence	9	EFF_end	Efficiency at End of Season
2	avg_co_inf	Average Efficiency	10	avg_EFF_end	Average Efficiency at End of Season
3	EFF	Efficiency	11	Co_H-index	Co-Player's H-index
4	avg_EFF	Average Efficiency	12	avg_Co_H-index	Average H-index of Co-Players
5	EFF_begg	Efficiency at Beginning of Season	13	HGS	Hollinger Score
6	avg_EFF_begg	Average Efficiency at Beginning of Season	14	avg_HGS	Average Hollinger Score
7	EFF_mid	Efficiency at Mid of Season	15	points_share	Co-Players points divided by Team Points
8	avg_EFF_mid	Average Efficiency at Mid of Season	16	avg_point_share	Average Point Share

3.8 Mathematical Formulation of Derived Features

This section shows the mathematical formulae for each of the derived feature.

3.8.1 Co-Player Influence

Co-player influence on a player is calculated by using the formula

$$co_inf(Player, co) = \frac{G_P}{Co_G}. \quad (3.3)$$

G_P: number of games a co-player played with a player P.

Co_G: Total number of games played by a co-player.

The above formula is used to calculate a single co-player influence on a player. Since a player has many co-players, so we need to find all co-player's influence on a player. The following formulae show all co-player influence on a player

$$co_inf = \sum_{i=1}^n co_inf_P(i). \quad (3.4)$$

co_inf_P(i): influence of i^{th} co-player on player P.

co_inf: sum of the influences of all co-players that played with a player P.

3.8.1.1 Example of Co-Player Influence Example given in Table.3.6 shows the calculation of co-player's influence

Table 3.6: Example showing the calculation of co-player's influence
(G_P are Number of Games in which Co-Players appeared with player John.
Co_G are total number of games played by a co-player).

Player	Co-Player	G_P	Co_G	G_P/Co_G	Influence
John	Fred	20	70	20/70	0.286
John	James	15	40	15/40	0.375

Above example shows that James has more influence (0.375) on John as compared to Fred influence (0.286) on John.

3.8.2 Average Influence of Co-Players

Since each co-player of a player has a different influence score. The following formulae show the average influence of co-players of a player P.

$$avg_co_inf = \sum_{i=1}^n \frac{co_inf_P(i)}{Co_N}. \quad (3.5)$$

Co_N: total no of co-players of a player P.

3.8.3 Co-Players Efficiency

Efficiency of basketball players depends on various factor. We use NBA efficiency formulae³ to find the efficiency of co-players of a player. This formula is given by the following equation.

$$Co_EFF = (PTS + REB + AST + STL + BLK - FGA - FTA - TOV). \quad (3.6)$$

The above formulae find the efficiency of a single co-player of a player P. To find the efficiency of all co-players of a player P, we sum up all co-player's efficiencies. The following equation shows the efficiency of co-players of a player P.

$$EFF = \sum_{i=1}^n Co_EFF(i). \quad (3.7)$$

Co_EFF(i): Efficiency of i^{th} Co-player of player P.

EFF: Efficiency of Co-players of Player P.

3.8.4 Average Efficiency of Co-Players

To find the average efficiency of co-players of player P, we just divide the co-players efficiency score (EFF) by total number of co-players of a player P. The following equation show the formulae for average efficiency score of co-players of a player P.

$$avg_EFF = \sum_{i=1}^n \frac{Co_EFF(i)}{Co_N}. \quad (3.8)$$

³<https://www.nbastuffer.com/analytics101/nba-efficiency/>

Co_EFF(i): Efficiency of i^{th} Co-player of player P.

Co_N: total no of co-players of a player P.

3.8.5 Co-Players Efficiency at Beginning of Season

Performance of basketball players changes at different intervals during a season. We can divide a season into beginning, mid and end intervals. The season intervals can be represented as.

$BD_1, BD_2 \dots BD_n, MD_1, MD_2 \dots MD_n, ED_1 \dots ED_n$.

BD: Beginning date of season.

MD :Mid date of season.

ED :End date of season.

To find efficiency of a co-player at the beginning of season, same efficiency formula is used with the condition that the games are played at beginning of the season.

$$EFF_Begg = (PTS + REB + AST + STL + BLK - FGA - FTA - TOV). \quad (3.9)$$

Where $Game_{Date < MD_1}$

The following formulae show the efficiency of all co-players of a player P at the beginning of the season

$$EFF_Begg = \sum_{i=1}^n Co_EFF_Begg(i). \quad (3.10)$$

Co_EFF_Begg(i): Efficiency of i^{th} co-player at begining of season.

3.8.6 Average Efficiency of Co-Players at Beginning of Season

To find average efficiency of co-players of a player P at beginning of a season efficiency the following formula is used

$$avg_EFF_Begg = \sum_{i=1}^n \frac{Co_EFF_Begg(i)}{Co_N}. \quad (3.11)$$

Co_EFF_Begg(i): Efficiency of i^{th} co-player at beginning of season.

3.8.7 Co-Players Efficiency at Mid of Season

To find efficiency of a co-player at the mid of season, same efficiency formulae is used with the condition that the games are played at the mid of season. The following formula shows the efficiency of a co-player of a player P at mid of season.

$$EFF_Mid = (PTS + REB + AST + STL + BLK - FGA - FTA - TOV). \quad (3.12)$$

Where $Game_{Date} > BD_n$ and $Game_{Date} < ED_1$

To find efficiency of all co-players of a player P at mid of season, the following formula is used

$$EFF_Mid = \sum_{i=1}^n Co_EFF_Mid(i). \quad (3.13)$$

Co.EFF_Mid(i): Efficiency of i^{th} co-player at Mid of season.

3.8.8 Average Efficiency of Co-Players at Mid of Season

To find average efficiency of co-players at mid of season we just divide the co-players efficiency at mid of season by total number of co-players of player P. The following equation show average efficiency of co-players of a player P at mid of season

$$avg_EFF_Mid = \sum_{i=1}^n \frac{Co_EFF_Mid(i)}{Co_N}. \quad (3.14)$$

3.8.9 Co-Players Efficiency at End of Season

To find efficiency of a co-player at the end of season, same efficiency formulae is used with the condition that the games are played during end of season

$$EFF_Beg = (PTS + REB + AST + STL + BLK - FGA - FTA - TOV). \quad (3.15)$$

Where $Game_{Date} > ED_n$

To find efficiency at end of season of all co-players of player P, the following formulae is used

$$EFF_End = \sum_{i=1}^n Co_EFF_End(i). \quad (3.16)$$

3.8.10 Average Efficiency of Co-Players at End of Season

To find average efficiency of co-players at end of season, we just divide the co-players efficiency at end of season by total no of co-players of player P. The following equation shows the average efficiency of co-players of a player P at end of season.

$$avg_EFF_End = \sum_{i=1}^n \frac{Co_EFF_End(i)}{Co_N}. \quad (3.17)$$

3.8.11 Co-Players H-index

H-index [107] is usually used to measures researchers' productivity but it also has been applied to different areas where the ranking of an individual is required. We calculate h-index for each co-player in order to overcome the lack of Average Efficiency Score of a co-player. Suppose if a co-player has average efficiency score 12, then it is assumed that a co-player efficiency score is 12 for every game he played but in reality, his efficiency score is not necessary to be 12 in every game, so the use of average does not tell about performance consistency of a player. H-index is used to measure how consistent a co-player is in his 6-season career. To find H-index of all co-players of player P, the following formulae is used

$$Co_H-index = \sum_{i=1}^n H-index(i). \quad (3.18)$$

3.8.11.1 Example of Co-Players H-index Lets understand the H-index of Co-Players through an example. Suppose John has played total 8 games. His Average Efficiency can be calculated as show in the Table.3.7

Table 3.7: Example showing the calculation of Average Efficiency.

Games	EFF (Efficiency)	Avg_EFF (Average Efficiency)
1	5	$(5+8+2+7+23+11+4+10)/8=8.75$
2	8	
3	2	
4	7	
5	23	
6	11	
7	4	
8	10	

Now in order to calculate the H-index of John, we simply write the efficiency in decreasing order as show in Table.3.8.

Table 3.8: Example showing the calculation of H-index Efficiency.

Games	EFF (Efficiency)
1	23
2	11
3	10
4	8
5	7
6	5
7	4
8	2

In given example the H-index of John is 5, because John have 5 such games in which his efficiency is greater than 5.

H-index(i) is the h-index of i^{th} co-player of a player. Lets understand through example as show in Table.3.9

Table 3.9: Example showing H-index of Co-Players of John.

Player	i	Co-Player	H-index
John	1	Mark	5
John	2	Brian	3
John	2	James	8
John	2	Jacob	6

Now H-index(i) for i=1,2,3,4:

H-index(1)=5

H-index(2)=3

H-index(3)=8

H-index(4)= 6

Now substituting values in equation.3.18

Co_H-index = 5+3+8+6=22

3.8.12 Average H-Index of Co-Players

To find average h-index of co-players of a player P, h-index of all co-players of a player P is divided by total number of co-players of a player P. The following formula is used to calculate average h-index.

$$avg_Co_H-index = \sum_{i=1}^n \frac{H-index(i)}{Co_N}. \quad (3.19)$$

3.8.13 Hollinger Score (HGS)

We use Hollinger linear formula⁴ to calculate co-players efficiency, the formula is given below.

$$\begin{aligned} CO_HGS = & (PTS) + 0.4 * (FG) + 0.7 * (OREB) + 0.3 * \\ & (DREB) + (STL) + 0.7 * (AST) + 0.7 * (BLK) - 0.7 * \\ & (FGA) - 0.4 * (FTA) - 0.4 * (PF) - (TOV). \end{aligned} \quad (3.20)$$

To find Hollinger score of all co-players of player P, the following formula is used

$$HGS = \sum_{i=1}^n Co_HGS(i). \quad (3.21)$$

⁴<https://www.nbastuffer.com/analytics101/game-score/>

3.8.14 Average HGS of Co-Players

To find average of HGS of co-players of a player P, we just divided HGS by total number of co-players of player P.

$$avg_HGS = \sum_{i=1}^n \frac{Co_HGS(i)}{Co_N}. \quad (3.22)$$

3.8.15 Points Share

Points share determine how much a co-player contributed to his team, it is determined by dividing co-player points (PTS) by total points of team.

$$Co_Points_Share = \frac{PTS}{Team_PTS}. \quad (3.23)$$

To find Points Share of all co-players of player P, the following formulae is used.

$$Points_Share = \sum_{i=1}^n Co_Points_Share(i). \quad (3.24)$$

3.8.16 Average Point Share of Co-Players

To find average Point Share of co-players of a player P, we just divide Point Share by total number of co-players of player P.

$$avg_Point_Share = \sum_{i=1}^n \frac{Co_Points_Share(i)}{Co_N}. \quad (3.25)$$

3.9 Experiments

This section covers a detailed discussion about various experiments, such as features distribution, feature analysis and comparison of various machine learning.

3.9.1 Statistical Distribution of Features

Since we have classified features by type and size. Here we only present statistical distribution of selected features for each dataset because they are the best selected features by size and performance on their relevant data set. Fig.3.10 shows the feature distribution of

Dataset A for selected feature set. For dataset A using selected feature set, all features are positively correlated to rising star except free throw attempt (FTA). In Fig.3.10 we can see that for the feature free throw attempt (FTA) values closer to 1 are related to not rising star class. For free throw percent (FTper) and blocks (BLK) we can see that up to the values 0.5 rising star class is dominant whereas after 0.5 not rising stars are getting closer to 1. For the remaining features, we can see that rising star class is dominant but as the feature values are getting higher than 0.7 then the not rising star class gets dominant. Since we observed that the values for rising stars dominant up to some extent but after that, we observe that the higher values belong to not rising stars, the reason for this abrupt change is that since rising stars have played more than 300 games, so there is a chance that they may also be played as co-players with not-rising star players, so their presence in the games with not-rising stars is the cause of the rise of feature values for not rising star players.

For selected features of Dataset B, all features are positively correlated with rising star class except for three point attempt (3PTA), free throw attempt (FTA) and average turnover (avg_TOV). In Fig.3.11 we can see that most of the feature values closer to 1 belong to rising star class and we can observe that feature values are dominant for rising star class. Since three point attempt (3PTA) is negatively correlated with rising star class but in Fig.3.11 we can see that for three point attempt (3PTA) most of the rising stars are closer to 1 as compared to not rising stars. Three point attempt (3PTA) is not the only feature to measure player's three point scoring ability. Three point percent (3PTper) better depicts a player's three point scoring ability.

In the Fig.3.10 we can see that for most of rising stars three point percent (3PTper) value is closer to 1. For free throw attempt (FTA) we know that it is negatively correlated to rising star class but we can see in the Fig.3.10 that for rising star class these values are closer to 1. The free throw is not a sufficient feature to represent the free throw ability of a player. Free throw percent (FTper) feature better depicts a player free throw ability. In the Fig.3.10 we can see that rising stars are closer to 1 as compared to not rising stars.

If we look at features distribution in Fig.3.10 and Fig.3.11, we can see that for most of the features in Fig.3.10 the feature values are dominant for rising stars up to some extent and after that not rising stars are dominant, whereas in Fig.3.11 we can see that rising stars are dominant till the end. The reason for the sudden change in the bar graph of Fig.3.10 is because the dataset A contains of the co-players from the same team only, so some of rising stars may also be co-players of not rising stars so this increase the feature values for not

rising stars. Since Fig.3.11 only contains co-players of the opposite team, so the chance of appearing as co-player with opponent team player is very rare therefore we can see in Fig.3.11 that there is no sudden change in the bar graph and rising stars are dominant for all features. Fig.3.12 shows the statistical distribution for selected features of Dataset C. All of the features of Dataset C using selected features are positively correlated to rising star class. In Fig.3.12 we can observe that more feature values belong to rising stars. Just like in Fig.3.10, we can see in Fig.3.12 that feature values very close to 1 belong to not rising stars, this is because the dataset C contain co-players from both same and opposite team and rising star players may have appeared as co-players with not rising star players. To avoid bias in

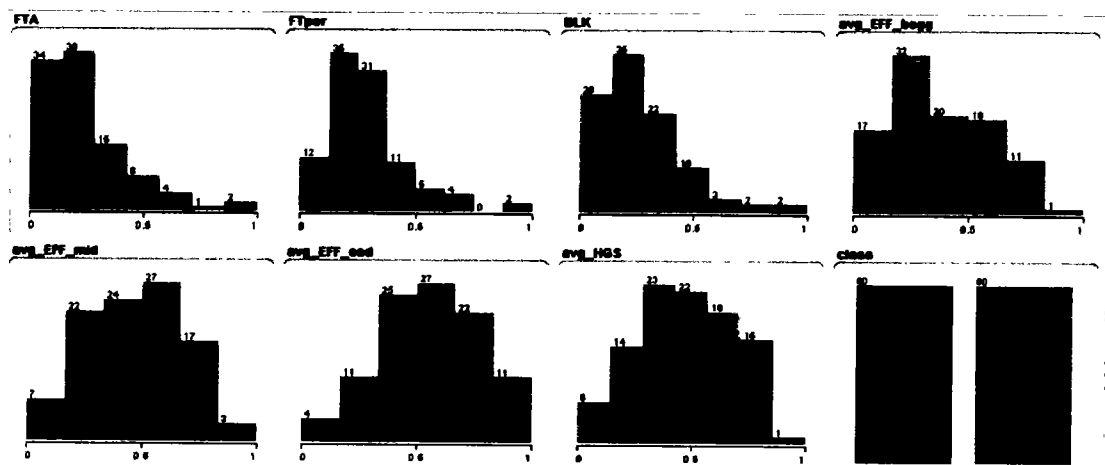


Figure 3.10: Features Distribution of Selected Features for Dataset A. (Red color shows rising star and blue color shows not-rising star. X-axis shows distribution of values and Y-axis shows frequency of values for an attribute.)

the results, all datasets are kept balanced by keeping the number of rising stars and not rising stars equal.

3.9.2 Features Evaluation

In this section, we used different features evaluation metrics to measure the relevance and importance of features for rising star prediction. Information gain, gain ratio and chi-squared statistics are used to measure the importance of different features. The said ranking metrics are only applied to the selected feature set of all three datasets.

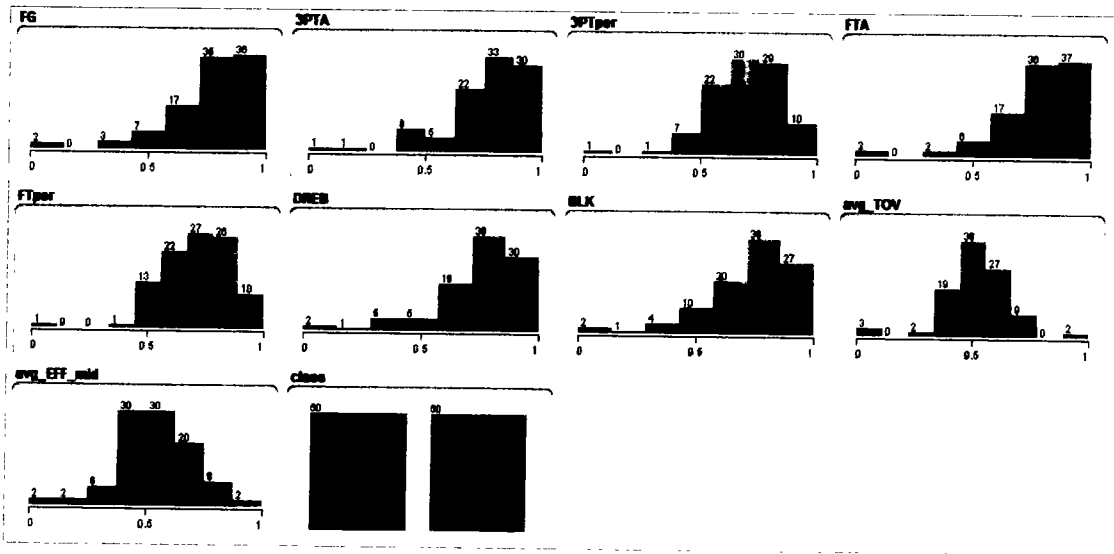


Figure 3.11: Features Distribution of Selected Features for Dataset B.

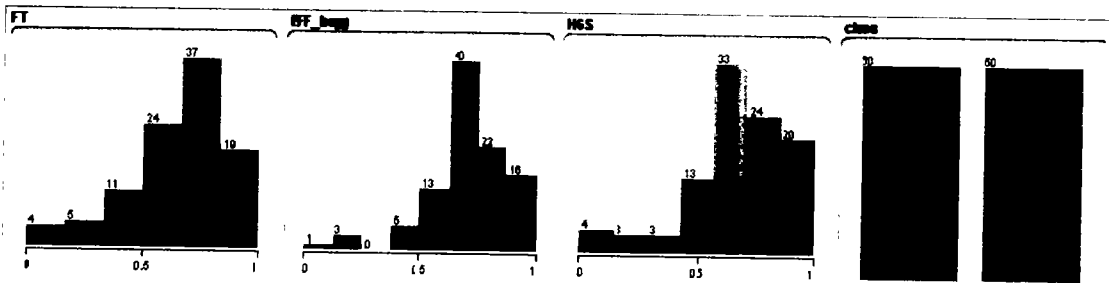


Figure 3.12: Features Distribution of Selected Features for Dataset C.

Table 3.10: Ranking of Features of Dataset A

Rank	Attribute	Info Gain	Attribute	Gain Ratio	Attribute	Chi-Square
1	avg_EFF_mid	0.354	avg_EFF_end	0.372	avg_EFF_mid	43.463
2	BLK	0.331	avg_EFF_mid	0.369	avg_HGS	40.96
3	avg_HGS	0.32	FTA	0.349	avg_EFF_begg	40.96
4	avg_EFF_begg	0.32	avg_HGS	0.32	FTA	35.406
5	FTA	0.303	avg_EFF_begg	0.32	BLK	33.134
6	avg_EFF_end	0.296	BLK	0.266	avg_EFF_end	31.579
7	FTper	0.24	FTper	0.24	FTper	31.36

In Table.3.10 we can see the ranking of selected features for dataset A, average efficiency score at mid of season (avg_EFF_mid) is ranked 1st by info gain and chi-squared statistics

whereas gain ratio ranked it 2nd. Free throw percent (FTper) is ranked 7th by all of the three metrics.

Table 3.11: Ranking of Features of Dataset B

Rank	Attribute	Info Gain	Attribute	Gain Ratio	Attribute	Chi-Square
1	FTA	0.281	FTA	0.315	FTA	34.0813
2	FG	0.256	FG	0.31	FG	29.9376
3	DREB	0.222	BLK	0.307	DREB	29.1717
4	3PTA	0.212	3PTA	0.273	3PTA	24.9012
5	BLK	0.209	DREB	0.222	BLK	21.9512
6	3PTper	0.165	FTper	0.202	3PTper	20.79
7	FTper	0.154	3PTper	0.2	FTper	18.8811
8	avg_EFF_mid	0.111	avg_EFF_mid	0.183	avg_TOV	14.0351
9	avg_TOV	0.108	avg_TOV	0.136	avg_EFF_mid	13.2549

Table.3.11 shows the ranking of selected features for dataset B, we can see that free throw attempt (FTA) is ranked 1st by all of the three metrics whereas average turnover (avg_TOV) is ranked 9th by info gain and gain ratio and is ranked 8th by the chi-squared statistic.

Table 3.12: Ranking of Features of Dataset C

Rank	Attribute	Info Gain	Attribute	Gain Ratio	Attribute	Chi-Square
1	EFF_begg	0.344	EFF_begg	0.409	HGS	39.317
2	HGS	0.337	HGS	0.377	EFF_begg	36.986
3	FT	0.296	FT	0.372	FT	31.579

Table.3.12 represent the ranking of selected features of Dataset C. Efficiency at the beginning of the season (EFF_begg) is ranked 1st by info gain and gain ratio and is ranked 2nd by the chi-squared statistic. Free throw (FT) is ranked 3rd by all of the three metrics.

3.9.3 Experimental Setup

Experimental setup is discussed below.

3.9.3.1 K-Fold Cross-validation . To measure the strength of classifiers and features, k-fold cross validation method [108] is used to train and validate the classifiers. In k-fold validation, the dataset is divided into k equal parts. One part is used for testing and k-1 parts are used training purpose. All experiments conducted in this reserch work used 10-fold validation (k=10), which means that one part is used for testing and 9 parts are used for training. Fig.3.13 shows pictorial representation of 10-fold cross validation(image is taken from [109]).

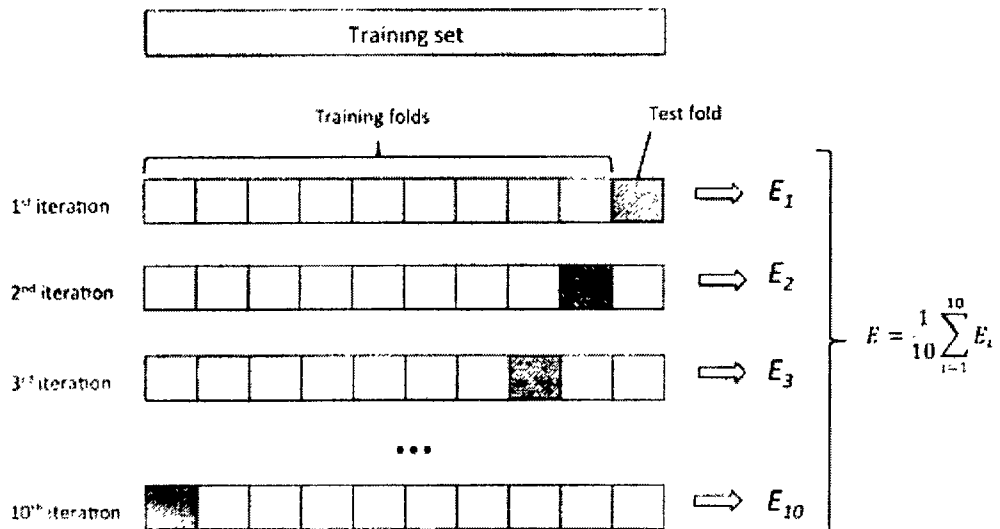


Figure 3.13: K-fold cross validation with k=10, E is Evaluation metric.

3.9.3.2 Dataset Size. Each dataset consist of 100 labeled samples is divided into 10 equal parts, such as 10,20, 30... 100. All classifiers are trained for each partition.

3.9.3.3 Parameters of Machine Learning Classifiers. Weka dafault parameter settings are used for each classifier. Parameters for each of classifier are mentioned below:

1. CART: confidence factor=0.25, minium number of instances per leaf=2.

2. SVM: kernel function= PolyKernel, complexity parameter C=1.0, epsilon=1.0E-12.
3. BN: estimator=simple estimator, search algorithm=hill climbing.
4. NB: kernel estimator=false, supervised discretization=false.
5. MEMM: multinomial logistic regression.

3.9.3.4 Performance Metrics. Precision, F-measure and Recall are computed for each dataset. We only present average F-measure for analysis of the results because F-measure is the combination of both Precision and Recall. The evaluations presented in the experiments are the average of 10 observations of F-measure.

3.9.3.5 Tools and Languages. Python, MySQL Workbench, Weka and MS-Excel are used from data scrapping to results presentation. Python is used for data scraping and pre-processing. MySQL Workbench is used for filtering data and feature engineering. All experiments are performed using open source software WEKA. MS-Excel is used for to present results through various types of charts.

3.10 Results and Discussion

This section discuss individual feature analysis and comparison of various results.

3.10.1 Individual Feature Analysis

To find the impact of each feature on rising star prediction, each classifier is trained while exploiting each feature of the selected feature set of each dataset. This process is done in 7,9 and 3 cycles for Dataset A, Dataset B and Dataset C respectively

3.10.1.1 Individual Feature Analysis of Dataset A Fig.3.14 shows the analysis of individual features from the selected feature set for dataset A. Using CART classifier with avg_HGS gives the highest average F-measure of 82.3% while the BLK feature gives the lowest score of 71%. SVM gives the highest average F-measure score 84% using avg_EFF_begg feature and lowest score 70% using FTper feature. MEMM gives the highest average F-measure score of 83% using avg_EFF_begg and the lowest score of 77% using FTper feature. Bayesian Network classifier gives the highest average F-measure score of 83% using avg_EFF_mid feature and gives the lowest score of 66% using FTper feature. Naïve Bayes

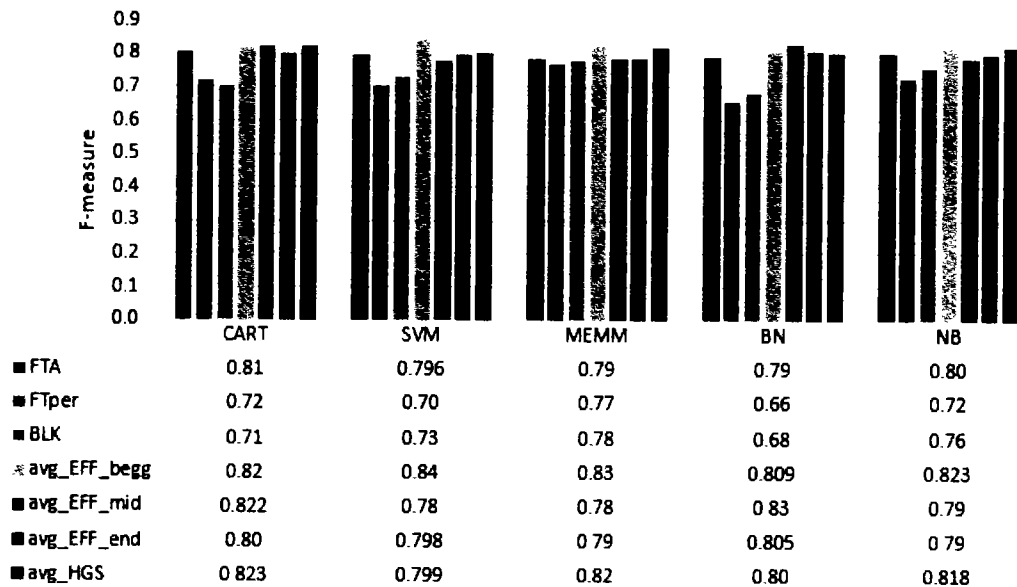


Figure 3.14: Individual Feature Analysis of Dataset A.

classifier gives the highest average F-measure score of 82.3% on avg_EFF_begg and whereas the FTper feature achieved the lowest average F-measure score of 72% on Naïve Bayes Classifier. Overall avg_EFF_begg feature gives the best average F-measure score whereas the FTper perform worst.

3.10.1.2 Individual Feature Analysis of Dataset B Fig.3.15 shows individual feature analysis of selected feature set of Dataset B. FTA feature achieve highest average F-measure score of 79% and avg_TOV feature has the lowest score of 52% using CART classifier. SVM classifier achieves the highest average F-measure score of 77% using FTA and achieves the lowest score of 52% using avg_TOV feature. MEMM model achieves the highest average F-measure of 80% using FG feature and the lowest score of 57% using avg_TOV feature. Bayesian Network achieves the highest average F-measure score of 77% using FTA feature and the lowest score of 56% using avg_EFF_mid feature. Using FTA feature Naïve Bayes classifier achieves the highest average F-measure score of 80% and has the lowest average F-measure score of 53% using avg_TOV feature. Among nine features FTA feature performs well on all classifiers while avg_TOV performs worst on all classifiers.

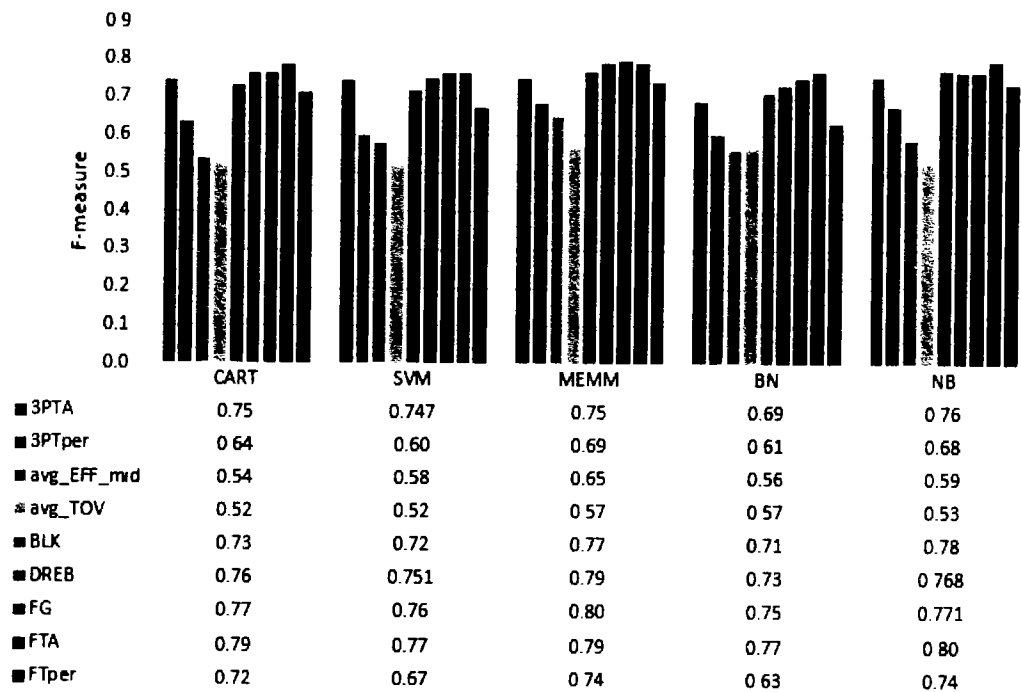


Figure 3.15: Individual Feature Analysis of Dataset B.

3.10.1.3 Individual Feature Analysis of Dataset C Fig.3.16 shows individual feature analysis of the selected feature set of Dataset C. CART classifier achieves the highest average F-measure score of 76.1% using EFF_begg and achieves the lowest average F-measure score of 75% using FT feature. Using the FT feature, SVM model achieves the highest average F-measure of 76% while using EFF_begg feature SVM achieves the lowest score of 73%. Using the MEMM model with FT feature achieves the highest average F-measure score of 78% while EFF_begg feature achieves the lowest score of 74%. Using the Bayesian Network classifier with FT feature gives the highest average F-measure of 73% whereas HGS gives the lowest score of 69.8%. Among the three features, FT achieves the best average F-measure score on all classifiers whereas EFF_begg achieves a low score on SVM, MEMM and Naïve Bayes classifiers as compare to FT (Free Throw) and HGS (Hollinger Score) features.

3.10.2 Category Wise Analysis

This section presents category wise analysis of various features. Combination of features into different categories is important to know the strenght of each category on classification

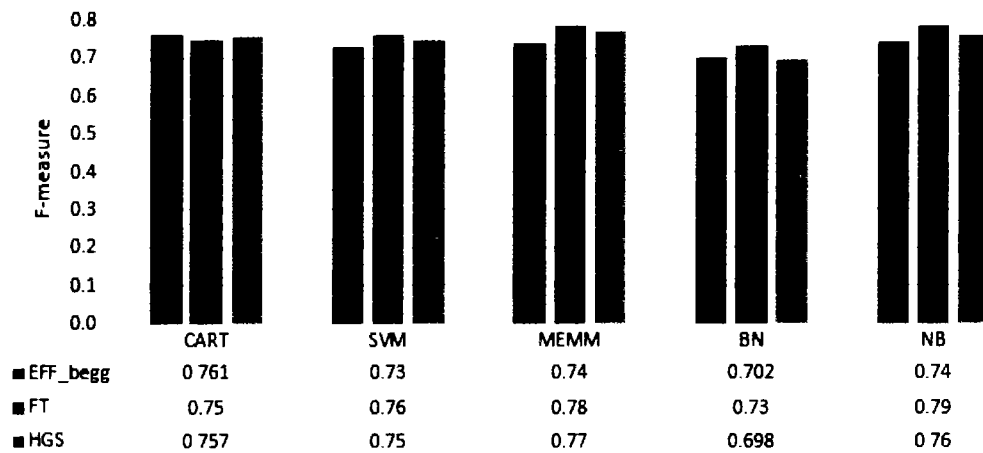


Figure 3.16: Individual Feature Analysis of Dataset C.

results. Feature types w.r.t to type and size are analyzed in this section for all of the three datasets. Fig.3.17 shows visual comparison of different categories of features.

3.10.2.1 Analysis of Features Classified By Type Table.3.13 shows analysis of three types of features for dataset A. Derived features achieve highest average F-measure score on all classifiers.

Table 3.13: F-Measure Analysis of Features Classified By Type

Classifier	DatasetA			DatasetB			DatasetC		
	Basic	Shooting	Derived	Basic	Shooting	Derived	Basic	Shooting	Derived
CART	0.85	0.83	0.89⁺	0.82	0.83	0.86	0.80	0.811	0.809
SVM	0.88⁺	0.83	0.89⁺	0.80	0.85⁺	0.84	0.74	0.78	0.79
MEMM	0.84	0.86 ⁺	0.87	0.81	0.84	0.87⁺	0.96⁺	0.91 ⁺	0.95 ⁺
BN	0.85	0.83	0.88	0.826 ⁺	0.833	0.834	0.64	0.63	0.68
NB	0.84	0.81	0.87	0.77	0.81	0.80	0.65	0.72	0.78

Table 3.14: F-Measure Analysis of Features Classified By Size

Classifier	DatasetA			DatasetB			DatasetC		
	All	Derived	Selected	All	Derived	Selected	All	Derived	Selected
CART	0.88⁺	0.892⁺	0.889	0.83	0.86	0.81	0.83	0.81	0.71
SVM	0.88⁺	0.89	0.90⁺	0.86⁺	0.839	0.844 ⁺	0.91	0.79	0.72
MEMM	0.85	0.87	0.90⁺	0.86 ⁺	0.87⁺	0.81	0.94 ⁺	0.95⁺	0.75 ⁺
BN	0.85	0.88	0.89	0.835	0.83	0.844⁺	0.66	0.68	0.69
NB	0.85	0.87	0.88	0.805	0.80	0.813	0.77	0.78	0.72

Bold values represent feature type with highest F-measure on a specific dataset.
+: Classifier with highest F-measure on a specific feature Type.

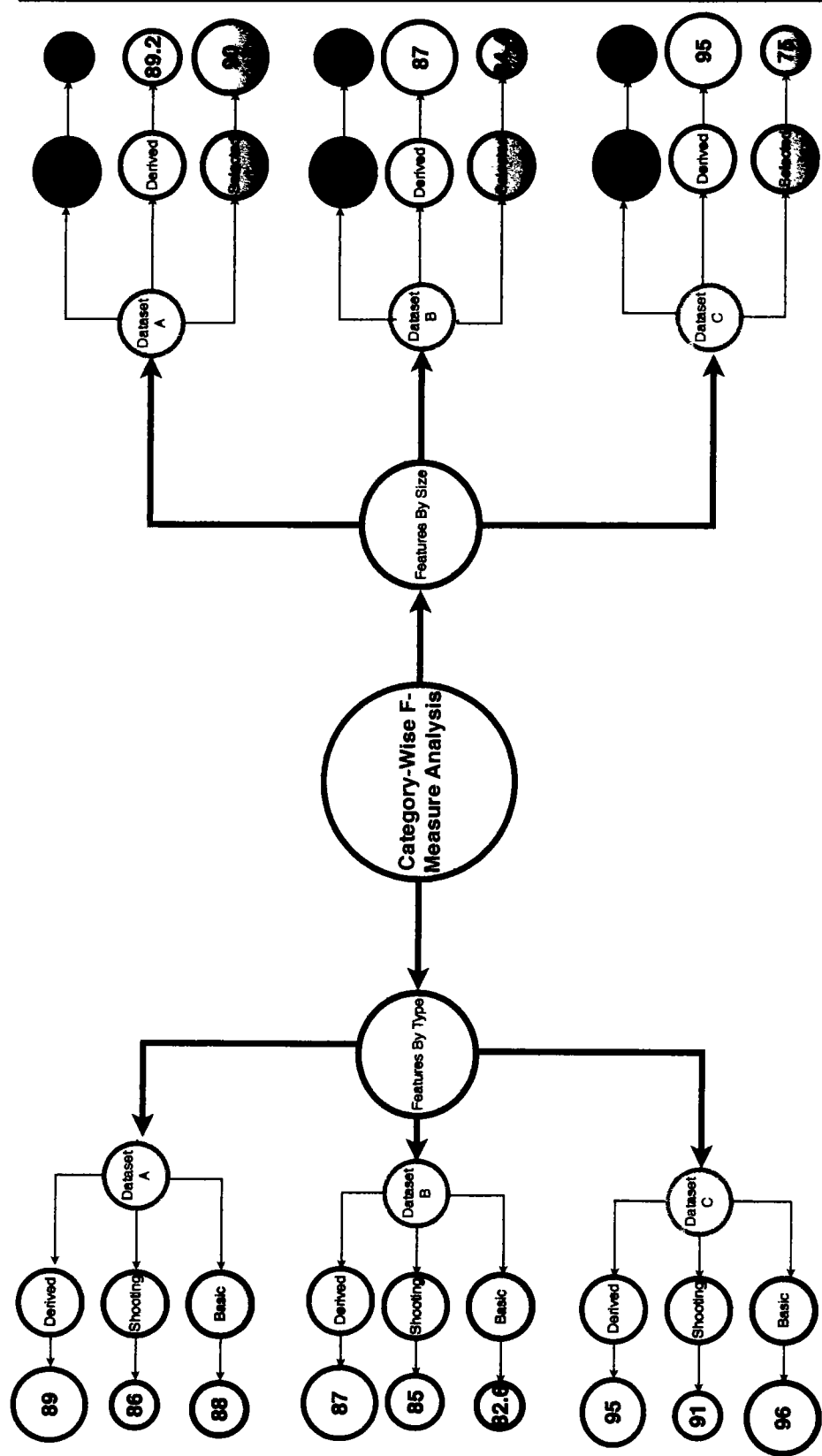


Figure 3.17: Comparison of F-measure score of different feature categories.

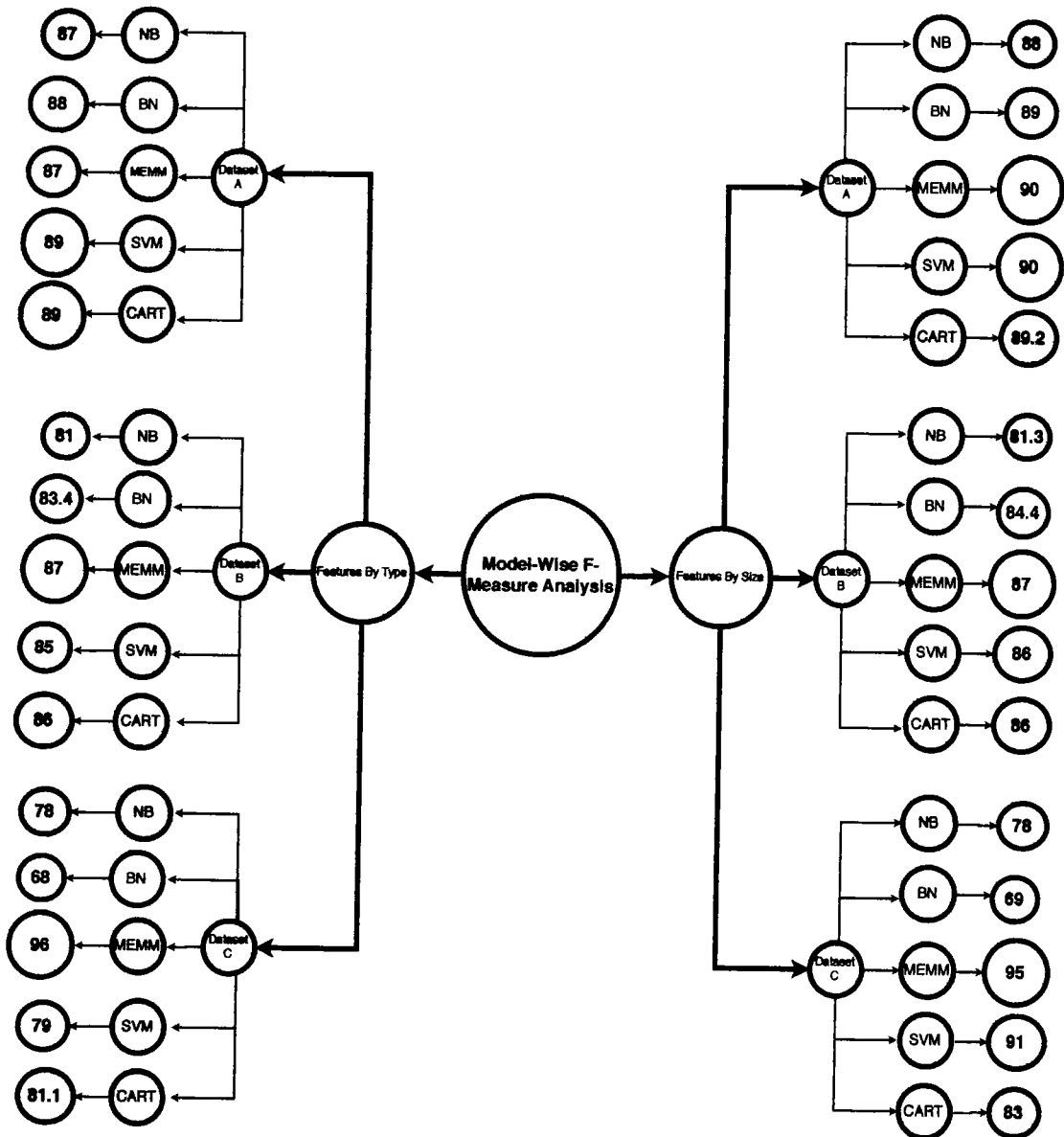


Figure 3.18: Comparison of F-measure score of different classifiers on three datasets.

Using Derived Features, CART and SVM produce highest average F-measure score of 89% whereas MEMM and Naïve Bayes classifiers produce low average F-measure score of 87% using Derived Features. Second highest average F-measure score is produced by using Basic Features. By using Basic Features SVM achieve highest average F-measure score of 88% whereas MEMM and Naïve Bayes produce a low score of 84%. Shooting Features produce low average F-measure score as compare to Basic and Derived Features. Shooting Features

achieve highest average F-measure score of 86% on MEMM model whereas it produces low average F-measure score of 81% using Naïve Bayes classifier. Table.3.13 shows the analysis of features classified by type for dataset B. Derived Features give highest average F-measure score of 86%, 87% and 83.4% for CART, MEMM and Bayesian Network classifiers respectively. On SVM classifier Derived Features produce average F-measure score of 84% which is greater than the average F-measure scores of Basic features and is less than average F-measure score of Shooting Features. On Naïve Bayes classifier Derived Features achieve average F-measure score of 80% which is greater than average F-measure score of Basic Features but less than the average F-measure score of Shooting Features on Naïve Bayes classifier. Shooting Features on SVM and Naïve Bayes Classifier achieve highest average F-measure score of 85% and 81% respectively. Using CART and MEMM classifiers produce average F-measure score of 83% and 84% respectively which is greater than the average F-measure scores on CART and MEMM for Basic Features and less than the average F-measure scores of the Derived Features on said classifiers. Using Basic features on CART, SVM, MEMM, Bayesian Network and Naïve Bayes achieve average F-measure scores of 82%, 80%, 81%, 82.6% and 77% respectively which are less than the average F-measure scores of shooting and derived features for the said classifiers. Table.3.13 show the analysis of features classified by type for dataset C. Derived Features achieve average F-measure scores of 80.9%, 79%, 95%, 68% and 78% on CART, SVM, MEMM, Bayesian Network and Naïve Bayes classifiers respectively. Shooting Features achieve average F-measure score of 81.1% which is greater than both basic and derived Features used on CART classifier. Using shooting Features on SVM and Naïve Bayes classifier achieve average F-measure score of 78% and 72% which is greater than average F-measure score of 74% and 65% for basic features used on said classifiers. MEMM and Bayesian Network classifier achieve average F-measure score of 91% and 63% using shooting features which is less than from both basic and derived features used on the said classifiers. Overall Derived Features achieve better results on all classifiers.

3.10.2.2 Analysis of Features Classified By Size In the previous section we presented detailed analysis of features classified by type for all of the three datasets. In this section we are going to see how the size of feature set affects the accuracy of rising star prediction. Table.3.14 shows average F-measure scores of different size of feature sets on different classifiers for dataset A. Selected Features achieve highest average F-measure scores on all classifiers except CART where average F-measure score of selected features is greater than

all feature set but less than derived feature set. Using selected feature set on CART, SVM, MEMM, Bayesian Network and Naïve Bayes classifiers achieve average F-measure scores of 88.9%, 90%, 90%, 89% and 88% respectively. Using derived features with CART, SVM, MEMM, Bayesian Network and Naïve Bayes classifiers achieve average F-measure scores of 89.2%, 89%, 87%, 88% and 87% respectively. Using all feature set on CART, SVM, MEMM, Bayesian Network and Naïve Bayes classifier achieve average F-measure scores of 88%, 88%, 85%, 85% and 85% respectively. We can clearly observe that selected features set with only 7 features achieve better average F-measure scores as compare to derived and all feature sets. The derived feature set only with 16 features achieve better average F-measure scores as compare to all feature set that have total 47 features. This concludes that for dataset A, the average F-measure score is improved by reducing the number of features. Table.3.14 shows different average F-measure scores for different classifiers with different size of feature sets using dataset B. selected feature set achieve highest average F-measures scores of 84.4% and 81.3% on Bayesian Network and Naïve Bayes classifiers whereas derived features achieve lowest average F-measure scores of 83% and 80% for Bayesian Network and Naïve Bayes respectively. Derived Feature set achieve highest average F-measure scores of 86% and 87% on CART and MEMM classifiers respectively whereas both classifiers achieve low average F-measure score of 81% using selected features. All feature set achieve highest average F-measure score of 86% using SVM classifier. Average F-measure scores of CART and MEMM using all feature set is greater than average F-measure scores of selected features on said classifiers whereas the same scores are less than the average F-measure scores of derived features on CART and MEMM classifiers. The analysis concludes that selected feature set with only 9 features achieve highest average F-measure scores on Bayesian Network and Naïve Bayes Classifiers whereas Derived Feature set with 16 features achieve highest average F-measure scores on CART and MEMM models. All Feature set with 47 features achieve highest average F-measure score on SVM classifier. We observed that reduced number of features (selected feature set and derived feature set) achieved highest average F-measure score on CART, MEMM, Bayesian Network and Naïve Bayes classifiers. Table.3.14 shows different average F-measure scores for different classifiers with different size of feature sets using dataset C. Using CART and SVM on all feature set achieve highest average F-measure scores of 83% and 91% respectively whereas all feature set achieve low average F-measure score of 66% on Bayesian Network classifier. Derived Features achieve highest average F-measure score of 95% and 78% on MEMM and Naïve Bayes classifiers. Selected features achieve highest average F-measure score of 69% on Bayesian network classifier whereas

same features achieve low average F-measure scores of 71%, 72%, 75% and 72% on CART, SVM, MEMM and Naïve Bayes classifiers respectively. Derived Feature set achieve highest average F-measure of 95%, second and third highest average F-measure score of 94% and 91% respectively is achieved by all feature set. MEMM classifier outperforms than other classifiers on each of the feature set.

3.10.3 Classifier Wise Analysis

In this section we present comparison of average F-measure scores for various classification models for the three datasets. Each dataset is divided into 10 to 100 instances. Fig.3.18 shows visual comparison of different machine learning models.

3.10.3.1 Analysis of Features Classified By Type In this section we will see how well various classification model performs on three datasets using the features classified by type.

3.10.3.1.1 Dataset A Table.3.13 show F-measure score analysis of different classifiers while using Basic Feature set. Every classifier achieved a maximum of 100% F-measure but SVM dominates all other classifiers by achieving 88% of average F-measure score for 10-100 instances. CART and BN stands second by achieving 85% of average F-measure score. NB and MEMM achieved lowest average F-measure of 84%. Table.3.13 shows F-measure score analysis for different classifiers while using Shooting Feature set. MEMM classifier dominates all other by achieving average F-measure score of 86% for 10-100 instances. The second-best average F-measure score of 83% is achieved by CART and SVM and BN. NB achieved the lowest average F-measure score of 81%. Same experiment was performed for dataset A using derived feature set. Table.3.13 shows detailed model analysis for dataset A by using derived feature set. CART and SVM dominates all others with average F-measure score of 89% for 10-100 instances. BN stands second with average F-measure score of 88% whereas MEMM and NB are ranked third 87% of average F-measure score.

3.10.3.1.2 Dataset B This section present analysis for different model by using dataset B with different type of feature sets. Table.3.13 shows that BN dominates all other classifiers by achieving average F-measure score of 82.6% for 10-100 instances. The second-best average F-measure score of 82% is achieved by CART classifier. MEMM is ranked third with average F-measure score of 81% whereas SVM is ranked fourth with average F-measure score of 80%. NB achieved lowest average F-measure score of 77%. The same experiment was

conducted by for dataset B using shooting feature set. Table.3.13 shows that SVM achieve highest average F-measure score of 85% for 10-100 instances. The second highest average F-measure score of 84% is achieved by MEMM model. BN is ranked third by achieving 83.3% of average F-measure score. CART is ranked fourth with average F-measure score of 82.5%. NB achieved lowest average F-measure score of 81%. Table.3.13 shows average F-measure score analysis for different classifiers while using Derived Feature set. MEMM dominates all other classifiers by achieving average F-measure score of 87% for 10-100 instances. Second best average F-measure score of 86% is achieved by CART model. SVM is ranked third with average F-measure score of 84%. BN is ranked fourth with average F-measure score of 83.4%. NB achieved lowest average F-measure score of 80%.

3.10.3.1.3 Dataset C This section present analysis for different model by using dataset C with different type of feature sets. Table.3.13 shows that by using Basic feature, the highest average F-measure score of 96% is achieved by MEMM model for 10-100 instances. CART is ranked second with average F-measure score of 80%. SVM with average F-measure score of 74% is ranked third. NB is ranked fourth with average F-measure score of 65%. BN is ranked last with an average F-measure score of 64%. Table.3.13 show that by using shooting feature set, highest average F-measure score of 91% is achieved by MEMM model. The second highest average F-measure score of 81.1% is achieved by CART model. SVM with average F-measure score of 78% is ranked third whereas NB with 72% and BN with 63% of average F-measure score are ranked fourth and fifth respectively. Table.3.13 shows that by using derived feature set, highest average F-measure score of 95% is achieved by MEMM model. CART is ranked second with average F-measure score of 81%. The third highest average F-measure score of 79% is achieved by SVM model. NB is ranked fourth with average F-measure score of 78% whereas BN is ranked fifth with average F-measure score of 68%.

3.10.3.2 Analysis of Features Classified By Size In previous section we discussed the effectiveness of various classification models with respect to features classified by type. In this section we will see the effectiveness of different classification models with respect to features classified by size for each dataset.

3.10.3.2.1 Dataset A Table.3.14 shows that by using all feature set, CART and SVM models achieved highest average F-measure score of 88% whereas MEMM, BN and NB

achieved average F-measure score of 85%. Table.3.14 show effectiveness of different models by using derived feature set. CART model achieved highest average F-measure score of 89.2%. SVM with average F-measure score of 89% is ranked second. BN with average F-measure score of 88% is ranked third. MEMM and NB are ranked fourth with average F-measure score of 87%. Table.3.14 shows effectiveness of different models by using selected feature set. SVM and MEMM achieved highest average F-measure score of 90%. Second highest average F-measure score of 89% is achieved by BN. CART and NB are ranked third and fourth with average F-measure scores of 88.9% and 88% respectively.

3.10.3.2.2 Dataset B Table.3.14 shows that by using all feature set SVM and MEMM models achieved highest average F-measure score of 86%. BN is ranked second with average F-measure score of 83.5%. Third highest average F-measure score of 83% is achieved by CART model. NB achieved lowest average F-measure score of 80.5%. Table.3.14 shows that by using derived feature set MEMM achieved highest average F-measure score of 87%. Second highest average F-measure score of 86% is achieved by CART model. SVM is ranked third with average F-measure score of 83.9%. BN with average F-measure score of 83% is ranked third. The lowest average F-measure score of 80% is achieved by NB model. Table.3.14 shows that by using selected feature set SVM and BN achieved highest average F-measure score of 84.4%. Second highest average F-measure score of 81.3% is achieved by NB classifier. MEMM and CART achieved lowest average F-measure score of 81%.

3.10.3.2.3 Dataset C Table.3.14 shhows that by using all Feature set MEMM is ranked first by achieving highest average F-measure score of 94%. SVM with average F-measure score of 91% is ranked second. The third highest average F-measure score of 83% is achieved by CART model. NB is ranked third with average F-measure score of 77%. BN achieved lowest average F-measure score of 66%. Table.3.14 shows effectiveness of different models by using derived feature set. MEMM achieved highest average F-measure score of 95%. Second highest average F-measure score of 81% is achieved by CART model. SVM with average F-measure score of 79% is ranked third. Fourth highest average F-measure score of 78% is achieved by NB model. BN achieved lowest average F-measure score of 68%. Table.3.14 shows that by using selected feature set. Highest average F-measure score of 75% is achieved by MEMM model. Second highest average F-measure score of 72% is achieved by SVM and NB models. CART achieved third highest average F-measure score of 71%. BN achieved lowest average F-measure score of 69%.

3.11 Season-wise Rankig Comparison Of Top 20 Labeled Rising Stars

From the labeled rising stars, Table.3.15 shows ranking of top-20 labeled rising stars. JamesL who is ranked at top in the labeled rising stars remain in top in four seasons (2009-2010 To 2012-2013), while the same player is still ranked in top 5 in 2013-2014 and 2014-2015 season. GarnetK has better ranking (appeared in top-100) in seasons 2009-2010 to 2012-2013, whereas the ranking of same player is reduced in subsequent seasons. Other players like NowitzkiD, BryantK, WadeD, PaulC, DuncanT, BoshC, GasolP, HowardD and AnthonyC are ranked in top-100 in all seasons given in the Table.3.15. Season-wise ranking presented in Table.3.15 are taken from a sports website⁵. Fig.3.19 shows season wise change in the ranking of top-20 labeled players. Fig.3.19 clearly indicates that out of 20 top-labeled rising stars most of the players are ranked in top-100. Only 3 to 5 players are ranked above 100.

⁵<http://www.espn.com/nba/seasonleaders>

Table 3.15: SEASON-WISE RANKING OF TOP 20 LABELED RISING STARS

Name	Rank	2009-2010	2010-2011	2011-2012	2012-2013	2013-2014	2014-2015
JamesL	1	1	1	1	1	2	5
GarnettK	2	56	40	29	50	185	180
NowitzkiD	3	9	14	19	37	15	53
BryantK	4	6	12	5	3	80	18
WadeD	5	3	4	8	10	30	24
PaulC	6	5	18	6	8	7	8
DuncanT	7	20	47	37	12	32	32
MarionS	8	105	90	99	66	127	302
MingY	9	135	Not Played	Not Played	Not Played	Not Played	Not Played
BoshC	10	21	4	30	33	42	58
GasolP	11	11	10	13	16	44	22
HowardD	12	40	14	2	4	18	17
NashS	13	Not Played	17	20	32	69	194
IversonA	14	120	Not Played	Not Played	Not Played	Not Played	Not Played
CambyM	15	Not Played	65	131	148	383	Not Played
PierceP	16	30	24	26	101	133	50
CarterV	17	111	164	102	139	335	74
KiddJ	18	82	140	173	Not Played	Not Played	43
AnthonyC	19	5	5	13	6	10	18
JamisonA	20	179	376	Not Played	44	58	58

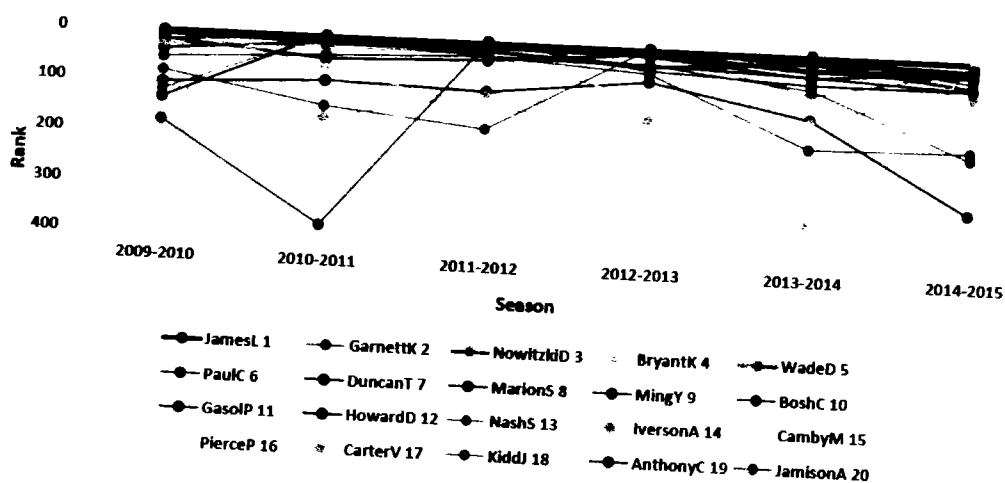


Figure 3.19: Season Wise Ranking of Top 20 Labeled Rising Stars.

3.12 Rising Stars VS NBA Most Improved Player

NBA Most Improved Players (MIP) are selected every year. The selection of MIP is based on the voting. The sportswriters cast their vote and the player with highest number of votes is awarded as Most Improved Player. The vote based selection of MIP is based on personal judgment. On the other hand Rising Stars are selected on the basis of their performance. In order to compare the rising stars with MIP⁶, we selected Most Improved Players for season 2004-2005 to 2008-2009. In total we have five Most Improved Players, one MIP for each season. For each season Average Efficiency of top five rising stars is compared with Most Improved Player of that season. Fig.3.20 shows the rising stars have better average efficiency in each season as compared to Most Improved Players of that season.

⁶<https://www.nba.com/history/awards/most-improved>

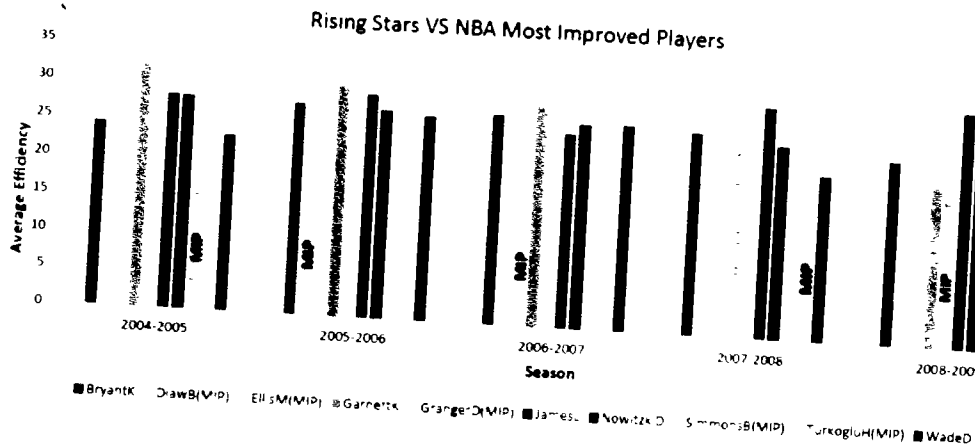


Figure 3.20: Comparison of Rising Stars Vs NBA Most Improved Players (Bar labeled with text "MIP" shows NBA most improved player of relevant season) .

3.13 Chapter Summary

This chapter covers rising star prediction in game of basketball. The chapter discussed basic concepts of basketball such as court, team structure and basic rules of basketball. Player game statistics such as field goal, points and free throws etc are presented. Process of dataset acquisition, dataset statistics and dataset labeling is discussed in detail. Model for rising star prediction in basketball is presented. Various machine learning models that are used for rising star prediction in basketball are presented in the chapter. Co-player selection and three types of co-players are introduced. Features are categorized by size and type which are further divided into various categories. Mathematical formulation of derived features is also presented. Feature evaluation shows importance of features by using info gain, gain ratio and chi-square. Experimental results include individual feature analysis, model wise analysis and category wise analysis. Comparison of rising stars with NBA most improved players is also presented.

Chapter 4

Exploring attributes for Rising Stars in Baseball

4 Exploring Attributes for Rising Stars in Baseball

This chapter discuss basic concepts of baseball, baseball dataset, features for rising star prediction and experimental results.

4.1 Basic Concepts of Baseball

This section discuss baseic terminologies of baseball like,baseball field, team structure and player positions in the field.

4.1.1 Baseball Field

The ground where baseball game is called baseball field. Due to its shape baseball field is also called diamond. The baseball field consist of four bases, pitcher's mound, infield and outfield positions.

4.1.2 Team Structure

Each team in baseball comprises on nine players. Batting team has hitter and 3 Base runners, whereas other players sit outside field. In case of strike the player sitting outside come into play. The fielding team has pitcher and catcher and fielders.

4.1.3 Positions

Positions of players in baseball are as following:

1. **Pitcher:** It is the player who pitches ball to the hitter.
2. **Catcher:** Player of opponent team standing behind the hitter is called pitcher.
3. **1st Baseman:** Fielder who positions near 1st Base.
4. **2nd Baseman:** Fielder who positions on 2nd Base.
5. **3rd Baseman:** Fielder who positions on 3rd Base.
6. **Shortstop:** Fielder positions between 2nd and 3rd Base.
7. **Leftfielder:** Outfield behind 2nd and 3rd base.

8. **Centerfielder:** Outfield behind 2nd base.
9. **Rightfielder:** Outfield behind 1st and 2nd base.

4.1.4 Basic Rules

Following are the general rules of baseball game.

1. Baseball game comprise of two teams with nine players each.
2. Fielding team comprises of pitcher,catcher,first baseman,second baseman,third baseman,three outfielders at left field,right field and centre field.
3. There are nine innings in baseball game, extra inning is played if game is tie after ninth inning.
4. choosen batting order can not be changed during the game.
5. If the batter hit the ball from the pitcher, at least they must try to get first base. They can then run to as many bases as they wish before being tagged out. Each base must be touched with some part of the batters body when running past.
6. When a batter tries to hit the ball but miss to hit then its is called strike. Three continuous strikes results in out of the batter.
7. Dismissal of a batter can occur through strikeout, force out, fly out and tag outs.

4.2 Baseball Dataset

The dataset was constructed by extracting players statistics from baseball retrosheet log files. Only those players are included in dataset who played at least 100 games. For labeling of Rising Stars only those players are selected who debuted in season 2005 and 2006. Total of 140 players were selected who debuted in year 2005 and 2006. Out of 140 players, Top 50 players with highest OPS (On-base plus slugging) are labeled as rising stars and bottom 50 players with lowest OPS are labeled as not-rising stars. Overall Statistics of the dataset are as following:

Years: 2005-2009

Total Teams: 30

Total Games: 12149

Total Players:1809

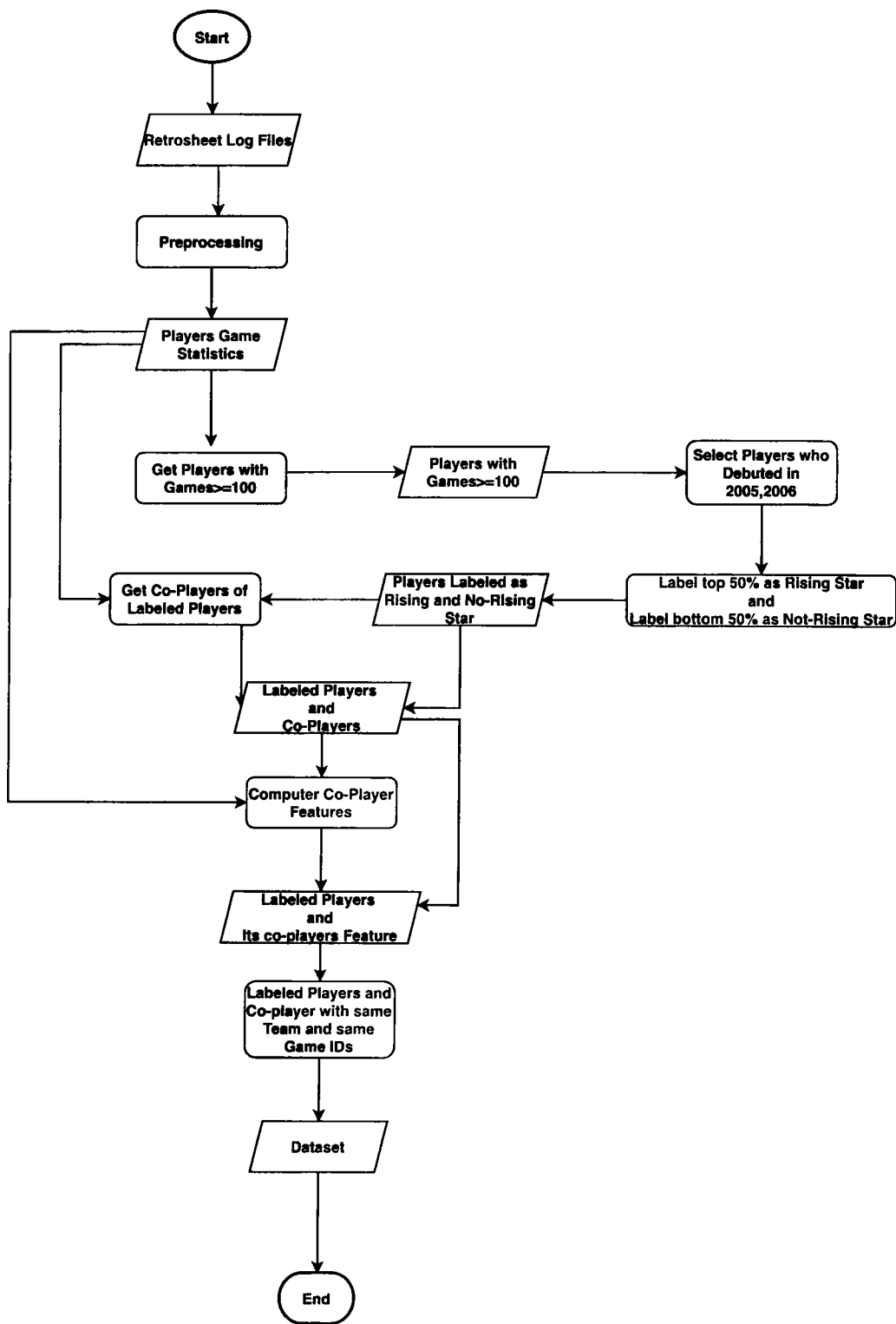


Figure 4.1: Flow Chart showing dataset acquisition process for baseball data.

Debuted in 2005-2006: 140

For each labeled player his co-players are selected who appeared in same games in which the labeled player played. For all of the co-players their features are computed. Flowchart in Fig.4.1 show the processing of building dataset.

4.3 Features for Rising Star Prediction

Features used for rising star prediction in baseball are listed in table 4.1.

Table 4.1: Details of Features Used for Rising Star Prediction

S.No	Feature	Description
1	G	Number of Games
2	PA	Plate Appearance
3	AB	At Bat
4	H	Hits
5	1B	Single Run
6	2B	Double Run
7	3B	Triple Run
8	HR	Home Run
9	R	Runs
10	RBI	Runs Batted In
11	BB	Base on Ball
12	k	Strikes
13	BA	Batting Average
14	OBP	On Base Percentage
15	SLG	Slugging Percentage
16	OPS	On-base plus slugging
17	TB	Total Bases
18	GDP	Grounded into Double Play
19	HBP	Hit By Pitch
20	SH	Sacrifice Hits
21	SF	Sacrifice Fly
22	IBB	International Base on Balls
23	RBOE	Reached Base on Error
24	BABIP	Batting average on balls in play

4.4 Features Categorization

Features are categorized by using feature selection, these categories of features are obtained by using feature selection method.. We used random forest classifier for feature selection.

Table 4.2: Top 5 Selected Features

S.No	Feature	Description
1	G	Number of Games
2	BB	Base on Balls
3	OBP	On Base Percentage
4	SH	Sacrifice Hits
5	BABIP	Batting average on balls in play

Table 4.3: Top 10 Selected Features

S.No	Feature	Description
1	G	Number of Games
2	PA	Plate Appearance
3	3B	Triple Run
4	BB	Base on Balls
5	k	Strikes
6	OBP	On Base Percentage
7	SLG	Slugging Percentage
8	HBP	Hit By Pitch
9	SH	Sacrifice Hits
10	BABIP	Batting average on balls in play

4.5 Experiments

This section discusses about category and classifier wise experiments performed.

4.5.1 Category-Wise Analysis

Table.4.5.1 shows F-measure score achieved by each category of feature by using different classification models. The first category,Top 5 Features achieved best results as compare to other categories. The first category not only have the minimum number of features but also have better accuracy than other categories. The highest F-measure score of 84% is

Table 4.4: Top 15 Selected Features

S.No	Feature	Description
1	G	Number of Games
2	1B	Singles
3	3B	Triple Run
4	R	Runs
5	RBI	Runs Batted In
6	BB	Base on Balls
7	k	Strikes
8	SLG	Slugging Percentage
9	OPS	On-Base Plus slugging
10	TB	Total Bases
11	GDP	Grounded into Double Play
13	HBP	Hit By Pitch
14	SH	Sacrifice Hits
15	SF	Sacrifice Fly

Table 4.5: Top 20 Selected Features

S.No	Feature	Description
1	G	Number of Games
2	AB	At Bat
3	H	Hits
4	1B	Singles
5	2B	Doubles In
6	3B	Triples
7	HR	Home Run
8	R	Runs
9	RBI	Runs Batted In
10	BB	Base on Balls
11	k	strikes
12	OBP	On Base Percentage
13	SLG	Slugging Percentage
14	OPS	On-base plus slugging
15	TB	Total Bases
16	HBP	Hit By Pitch
17	SH	Sacrifice Hits
18	SF	Sacrifice Fly
19	IBB	International Base on Balls
20	BABIP	Batting average on balls in play

achieved by first category of features by using Navive Bayes classifier. Top 10, Top 15 and Top 20 categories achieved same highest F-measure score of 83% by using SVM classifier . Comparison shows that there is no significant improvement in F-measure score when feature size is above 10. Fig.4.2 shows average F-measure score of each category of feature on CART,SVM,MEMM,NB and BN classifiers.

<i>Model</i>	<i>Top5Features</i>	<i>Top10Features</i>	<i>Top15Features</i>	<i>Top20Features</i>	<i>AllFeatures</i>
CART	0.80	0.81	0.81	0.81	0.81
SVM	0.83	0.83	0.83	0.83	0.83
MEMM	0.81	0.79	0.77	0.77	0.80
BN	0.87	0.76	0.74	0.74	0.73
NB	0.84	0.75	0.73	0.73	0.73

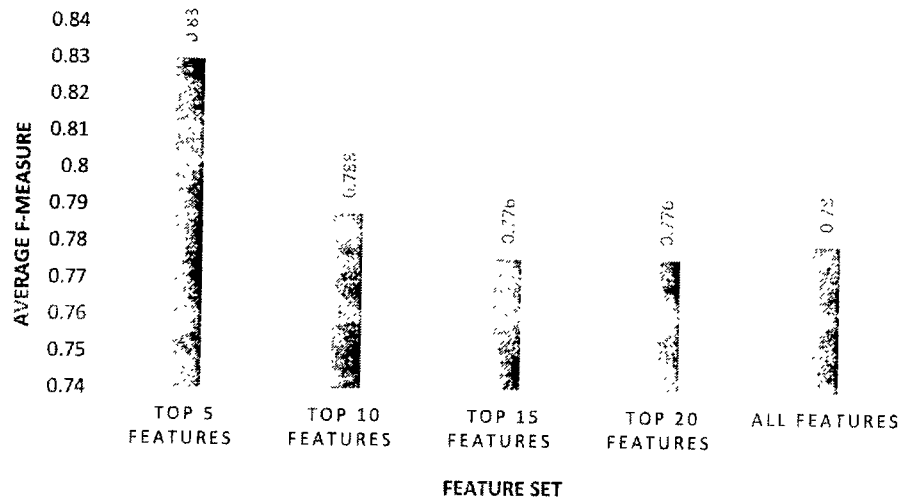


Figure 4.2: Comparison of different Feature Sets.

4.5.2 Classifier-Wise Analysis

Each classification models was applied on Top5, Top 10, Top 15 and Top 20 selected feature categories. In Fig.4.2 for each of classification model their F-measure score is average of all

categories of features. The comparison shows that SVM achieved highest average F-measure score of 83%.

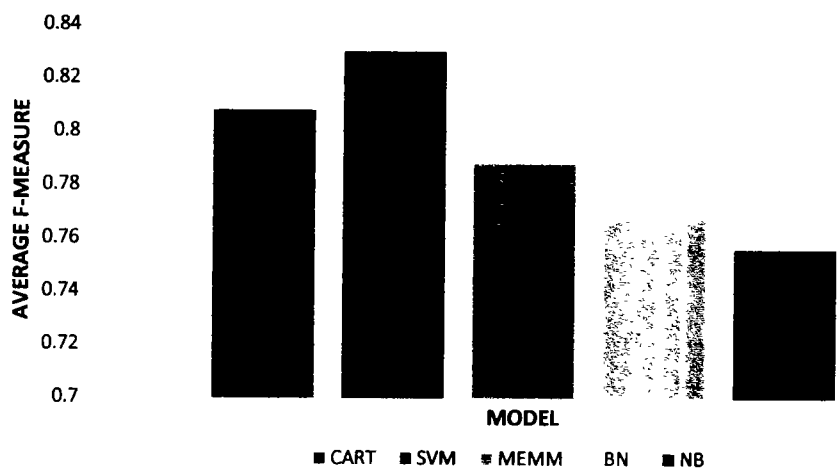


Figure 4.3: Comparison of different Machine Learning Classifiers.

4.6 ML models For Rising Star Prediction in other Domains

As we used machine learning models for rising star prediction in the game of basketball. Similar machine learning models have been used for rising star prediction on other domains as well. Table.4.6 shows comparison of F-measure score of various machine learning models used for rising star prediction in various domains.

Comparison of different ML models in different domains shown in Table.4.6 are best F-measure scores on these domains from different experiments using different features and datasets. It can be observed that MEMM classifier showed better performance in Basketball and co-author network domains whereas in cricket MEMM was not applied. Performacne of CART models was greater than 80% in all domains expt the domain of Pakistani Research Community where its performance is worst. Performance of SVM is not much different in Basketball and Cricket domain and this model is not used in co-author netowrk and Pakistani

Table 4.6: Comparison of F-Measure of ML models in different Domains for Rising Star Prediction

Domain	CART	SVM	MEMM	BN	NB
Baseball	80.0	83.0	81.0	87.0	84.0
Basketball	89.2	91.0	96.0	89.0	88.0
Cricket[14]	90.1	92.6	-	91.1	94.5
Co-Author Netowrk[4]	99.0	-	100.0	93.0	92.0
Pakistani Research Community[110]	50.0	-	98.0	96.0	96.0

research community domains. BN and NB models performed well in Pakistani research community and baseball domains as compared to its performance in other domains.

Bayesian Network performed better in Baseball domain because only releavent features were used and Bayeseian Network handle both dependent and independent relaitonship among features. There may be two reasons of better performance of MEMM in Basketball domain: discriminative nature of MEMM classifier and distance between data points of the two classes. As for as the misclassification of machine learning models is concerned, its is due to position of data points. When certain data points are very close to both classes then it is very hard for the classifier to correctly classify and it may lead to misclassify the instance.

4.7 Chapter Summary

In this chapter we explored attributes that can be useful for rising star prediction in baseball. Basic terminologies related to baseball such as baseball field, team structure and positions and basic rules of baseball are discussed. The process of acquiring baseball dataset and labeling of dataset is discussed in detail and flowchart for data acquisition is also presented. Various types of features and their description is also presented. Random forest classifier have been used to group features into three categories. Experimental results include cateogy wise and classifier wise analysis. At the end comparison of varios machine learning models in basketball and baseball with other related domains is presented.

Chapter 5

Impact of Expert Players on Performance of Junior Players

5 Impact of Expert Players on Performance of Junior Players

This chapter discuss dataset and experimental results that find the impact of expert players on performance of junior players.

5.1 Dataset Description and Experiment Setup

Here in this section we discussed dataset description and experimental settings that we have used for analysis. We have taken data of baseball hitters from year 2005 to 2009 from www.retrosheet.org platform. Important statistics of the data are:

- Total Games (2005-2009): 12149
- Total Teams: 30
- Total Players: 1809
- No of Expert Players: 15
- Number of players in Group A (Players who appeared with Expert Player): 988
- Number of candidates in Group A in IP: 197
- Selected from Group A having maximum games: 158
- Number of players in Group B (Players never played with Expert Players): 806
- Number of players in Group B in IP: 158

Table 5.1: Experiment Setup

S.No	Terminology	Description
1	Identifying Period (IP)	Time Period of 2005,2006
2	Verifying Period (VP)	Time Period of 2007-2009
3	Expert Players	Top 15 selected players
4	Group A	Players that have played with expert players
5	Group B	Players that have never played with expert players

We divided the time span into identifying and verifying periods. Identifying period consist of year 2005 and 2006, whereas verifying period consists of year 2007 to 2009. The purpose of identifying period is to select players who have debuted in year 2005 and year 2006. The aim of dividing time span into identifying and verifying period is to see whether performance of players have been improved or declined in the verifying period. Further, we have selected

selected top 15 expert players. We further divided players into Group A and Group B. Players in Group A are those players who have played with expert players whereas players of Group B are those players who never appeared in any game with expert players.

5.2 Candidates Comparison of Group A and Group B

Candidates comparison of Group A and Group B are shown in Table 5.2. The number of candidates in each group are 158 which make both groups balanced. Group A has more number of players that have played games greater than 100 as compared to group B. Similarly for number of Games greater than 200 Group A is still dominant. If number of Games played in identified period are compared, then we can observe that candidates of Group A has played more games as compared to candidates of Group B. Similar comparison of number of games played in verifying period shows that candidates of Group A have played more number of games as compared to candidates of Group B. If we compare the last game played, it can be seen that players of Group B are quitting their career very early as compared to Group A. The comparison also shows that players of Group A have better OPS as compared to players of Group B. The comparison of both groups conclude that the players of Group A are more active and better in performance as compared to players of Group B.

Table 5.2: Number of Games and OPS Comparison of Group A and Group B

Candidates	Group A	Percentage A	Group B	Percentage B
Number	158		158	
Games \geq 100	79	50.000	41	25.949
Games \geq 200	58	36.709	27	17.089
Games in IP \geq 50	42	26.582	26	16.456
Games in IP \geq 100	23	14.557	14	8.861
Games in VP \geq 50	83	52.532	42	26.582
Games in VP \geq 100	66	41.772	34	21.519
Games in VP \geq 150	55	34.810	27	17.089
Last Game in 2005	5	3.165	24	15.190
Last Game in 2006	14	8.861	20	12.658
Last Game in 2007	7	4.430	26	16.456
Last Game in 2008	23	14.557	22	13.924
Last Game in 2009	109	68.987	66	41.772
Having OPS \geq 0.7 (G \geq 50)	37	23.418	21	13.291
Having OPS \geq 0.5 (G \geq 50)	76	48.101	46	29.114
Having OPS \geq 0.3 (G \geq 50)	83	52.532	49	31.013

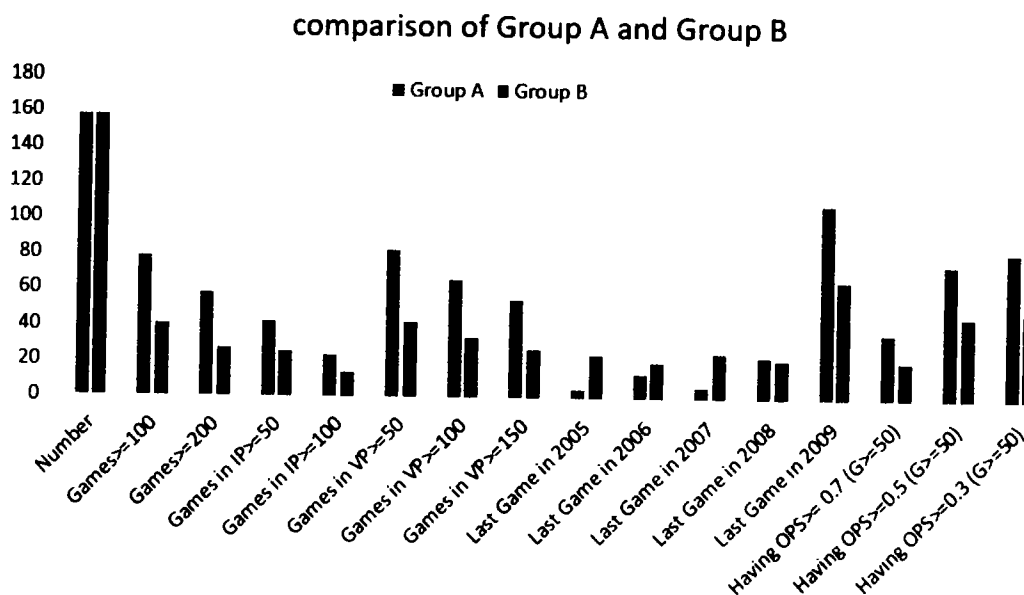


Figure 5.1: Candidates comparison of Group A and Group B.

5.3 Performance Analysis of Group A and Group B

A detailed analysis of Group A and Group B is carried out in Table 5.3. The number of candidates in Group A are more as compared to Group B in both identified and verified periods. If we look at number of games played during identified and verified periods, the results shows that Group A is dominant in term of number of games played in verified period. If we look at OPS comparison, it can be clearly observed that OPS of players of Group A have been much improved in verified period as compared to the improvement in OPS of players of Group B in verified period. Comparison of runs R of Group A and Group B in both identified and verified period shows that Group A have more increase in R value as compared to R value of players of Group B. Home runs (HR) comparison shows that candidates of Group A have better improvement in verified period as compared to improvement in HR of Group B. Players of Group A have better improvement in RBI (runs batted in) in verified period as compared to improvement in RBI of Group B.

Table 5.3: Performance Analysis of Group A and Group B

Candidates	Group A(IP)	Group B(IP)	Group A(VP)	Group B(VP)	Percent A(IP)	Percent B(IP)	Percent A(VP)	Percent B(VP)
Number	153	130	139	114				
Games ≥ 100	23	14	66	34	15.033	10.769	47.482	29.825
Games ≥ 200	4	3	51	25	2.614	2.308	36.691	21.930
Having OPS ≥ 0.7 (G ≥ 50)	25	12	38	21	16.340	9.231	27.338	18.421
Having OPS ≥ 0.5 (G ≥ 50)	38	26	70	35	24.837	20.000	50.360	30.702
Having OPS ≥ 0.3 (G ≥ 50)	42	26	78	40	27.451	20.000	56.115	35.088
Having R ≥ 30	31	18	63	34	20.261	13.846	45.324	29.825
Having R ≥ 50	18	11	54	32	11.765	8.462	38.849	28.070
Having R ≥ 100	6	2	41	22	3.922	1.538	29.496	19.298
Having HR ≥ 10	19	10	51	25	12.418	7.692	36.691	21.930
Having HR ≥ 20	8	4	32	24	5.229	3.077	23.022	21.053
Having HR ≥ 30	3	0	27	19	1.961	0.000	19.424	16.667
Having RBI ≥ 25	33	19	62	34	21.569	14.615	44.604	29.825
Having RBI ≥ 50	15	12	52	28	9.804	9.231	37.410	24.561
Having RBI ≥ 100	5	3	36	22	3.268	2.308	25.899	19.298
Having BA ≥ 0.1 (G ≥ 50)	42	26	80	41	27.451	20.000	57.554	35.965
Having BA ≥ 0.2 (G ≥ 50)	37	23	64	33	24.183	17.692	46.043	28.947
Having BA ≥ 0.3 (G ≥ 50)	3	4	5	1	1.961	3.077	3.597	0.877
Having K ≥ 50	32	16	76	37	20.915	12.308	54.676	32.456
Having K ≥ 100	10	4	52	25	6.536	3.077	37.410	21.930
Having K ≥ 150	3	1	36	20	1.961	0.769	25.899	17.544
Having 1B ≥ 25	39	25	74	36	25.490	19.231	53.237	31.579
Having 1B ≥ 50	25	16	60	34	16.340	12.308	43.165	29.825
Having 1B ≥ 100	9	10	49	25	5.882	7.692	35.252	21.930
Having 2B ≥ 10	38	20	69	35	24.837	15.385	49.640	30.702
Having 2B ≥ 20	22	12	56	28	14.379	9.231	40.288	24.561
Having 2B ≥ 30	7	7	46	23	4.575	5.385	33.094	20.175
Having 3B ≥ 4	16	5	40	20	10.458	3.846	28.777	17.544
Having 3B ≥ 7	7	3	26	12	4.575	2.308	18.705	10.526
Having 3B ≥ 10	2	2	15	6	1.307	1.538	10.791	5.263
Having H ≥ 50	34	19	70	35	22.222	14.615	50.360	30.702
Having H ≥ 100	16	12	54	33	10.458	9.231	38.849	28.947
Having H ≥ 150	10	8	48	24	6.536	6.154	34.532	21.053
Having AB ≥ 100	46	29	84	42	30.065	22.308	60.432	36.842
Having AB ≥ 300	23	14	59	34	15.033	10.769	42.446	29.825
Having AB ≥ 500	11	9	52	26	7.190	6.923	37.410	22.807
Having PA ≥ 100	50	30	86	42	32.680	23.077	61.871	36.842
Having PA ≥ 300	29	16	62	35	18.954	12.308	44.604	30.702
Having PA ≥ 500	12	10	53	27	7.843	7.692	38.129	23.684

Since players of Group A have played more number of Games in both identified and verified periods, they also have high number of strike outs as compared to players of Group B. The comparison of other indicators like 1B, 2B, 3B, H, AB and PA shows that players of Group A have better improvement as compared to players of Group B. The analysis conclude that playing with expert players can have positive impact on performance of junior players.

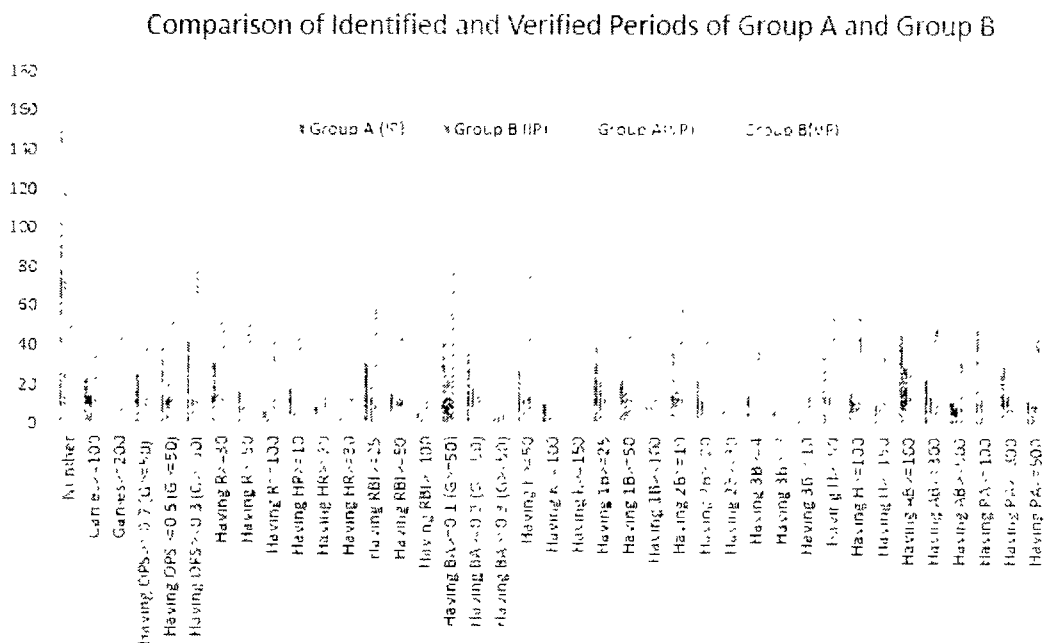


Figure 5.2: Performance analysis of Group A and Group B.

Spearsman correlation is performed for both Group A and Group B for identified and verified periods. Reason for using spearsman correlation test [111] is because both group shows proportion of players and the data is not normalized, so its nature is nonparametric. For both groups the spearsman correlation value is 0.9, which show a strong relationship between identified and verified periods of both groups. To further examine which group has better improvements in verified period, average of both identified and verified periods was calculated. For Group A, average of performance indicators was 24.25 in identified period whereas the average of performance indicators in verified period was improved to 55.79. For Group B average of performance indicators in identified period was 15.66 whereas average of performance indicators was improved to 30.33 in verified period. The said comparison

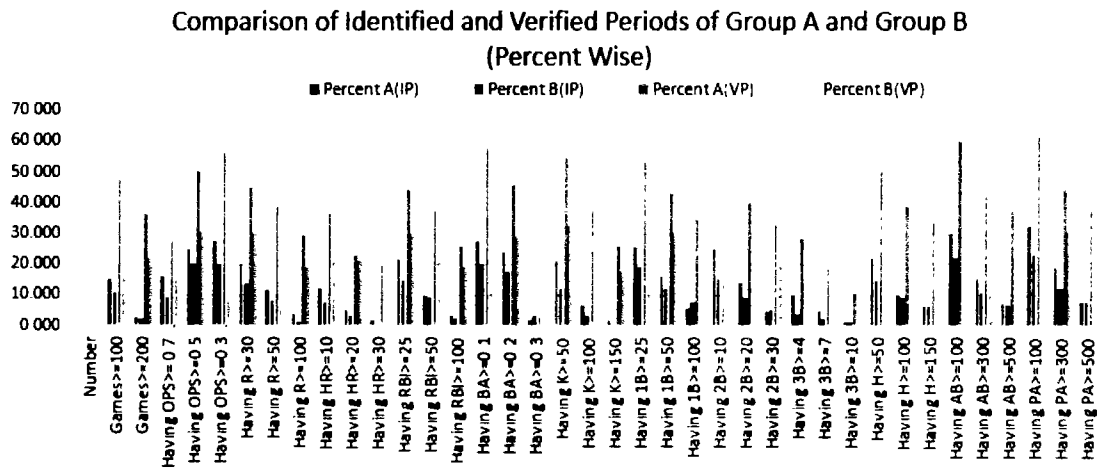


Figure 5.3: Performance analysis of Group A and Group B (Percent Wise).

shows that performance of players in Group A was much improved as compared to performance of players in group B.

5.4 Chapter Summary

In this chapter we performed analysis whether playing with expert and senior player will increase or decrease performance of junior players?. To find the impact of expert players on performance of junior player, first top fifteen expert players were selected. Two groups were formed, the first group contained the players who appeared in same games where expert players were part of the team. The second group contained players who never appeared with expert players in any game. The comparison of two groups showed that players who appeared with expert players have better performance as compared to the players who never appeared with expert players.

Chapter 6

Conclusion and Future Work

6 Conclusion and Future Work

This section discuss limitation coclusion and expected future work of rising star prediction in various other domains.

6.1 Conclusion

Ranking of players on basis of their career statistics is useful when the player has played enough games but it is difficult to measure the strength of the players who just started his career and has played few games. One of the limitations of ranking on past performance is that it does not tell about future performance of the players. Machine learning techniques are widely used nowadays for future prediction. In this thesis we used machine learning techniques to predict whether a player is rising star or not rising star. Unlike to ranking of players where players past performance is used to rank them. Here in this study we used the features of the co-players of players to predict player as rising star or not rising star. We also introduced three types of co-players in this thesis and analysis was performed to measure how each type of co-player's features are effective in predicting the rising star in game of basketball. The co-player features are based on their game statistics, these features were further classified on the basis of features type and features size, which were further divided into various categories.

The individual feature analysis shows that the avg_EFF_begg (Average Efficiency at Beginning of Season) whcih belong to derived feature set, achieved highest F-measure of 84% using SVM classifier on dataset A. If we a look at category wise analysis of feature type, though basic feature type achieved highest average F-measure of 84% but derived feature type is dominant on the three datasets. Category wise analysis base on feature size shows that derived features are dominant on dataset B and dataset C in terms of F-measure whereas selected features set dominant on dataset A. The model wise analysis shows that MEMM classifier is dominant in term of F-measure on both features classified by type and by size. The comparison of ranking of top 20 labeled rising stars with their ranking for the next 6 seasons shows that most of the rising star players have been ranked in the top 100 players which shows the effectiveness of our rising star prediction. Top five rising star were compared with five most improved players (MIP) of NBA, which showed that rising stars are better than those of most improved players in term of efficiency.

In this research we used game statistics of co-players as features for rising star prediction in

basketball. One of the limitation of this study is that it does not consider the physical features of rising stars or co-players. Physical features like age, weight, endurance, running speed and other physical attributes may be useful for rising star prediction. The other limitation of this research work is that it focus on feature engineering and evaluation of various machine learning models for rising star predictions in team sports but it does not show how machine learning classifiers used can be further improved.

6.2 Future Work

In the future, we might extend approach of rising star prediction to other sports like football and hockey. We will work on motion analysis of rising star players and their co-players to know the physical interaction of rising stars and their co-players. The other prominent future work is analysis of player's performance in local level games before debut in NBA. Interesting work is to use social media data for prediction of rising star in sports.

Rising star concept can be used for the prediction of viral videos. Just like rising stars viral videos have some properties and using these properties as features to machine learning classifier a video can be predicted in advance whether it will become viral or not.

Concept of rising star can also be applied to study the increasing demand of energy. Finding factors that are directly related to demand of energy, the demand of energy can be forecasted in advance.

Potential application of rising star can be identify the demanding skills in any industry through which such skills can be recommended to the youth in advance.

With the rise of big data, before any election event, much of data is generated on social media like twitter and facebook. Such data can be used by using rising star concept to predict the winner of election in advance.

Since this thesis used machine learning models with limited datasets, the current work can be extended to utilize big data platforms like hadoop and use deep learning algorithms to further enhance the current work.

Since the current work mostly focus on the applicability of rising star concept in various domains, there is need to develop generalized statistical and mathematical models for finding rising stars.

Bibliography

- [1] Xiao-Li Li, Chuan Sheng Foo, Kar Leong Tew, and See-Kiong Ng. Searching for rising stars in bibliography networks. In *International conference on database systems for advanced applications*, pages 288–292. Springer, 2009.
- [2] Ali Daud, Rashid Abbasi, and Faqir Muhammad. Finding rising stars in social networks. In *International conference on database systems for advanced applications*, pages 13–24. Springer, 2013.
- [3] George Tsatsaronis, Iraklis Varlamis, Sunna Torge, Matthias Reimann, Kjetil Nørvåg, Michael Schroeder, and Matthias Zschunke. How to become a group leader? or modeling author types based on graph mining. In *International Conference on Theory and Practice of Digital Libraries*, pages 15–26. Springer, 2011.
- [4] Ali Daud, Muhammad Ahmad, MSI Malik, and Dunren Che. Using machine learning techniques for rising star prediction in co-author network. *Scientometrics*, 102(2):1687–1711, 2015.
- [5] Ali Daud, Naif Radi Aljohani, Rabeeh Ayaz Abbasi, Zahid Rafique, Tehmina Amjad, Hussain Dawood, and Khaled H Alyoubi. Finding rising stars in co-author networks via weighted mutual influence. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 33–41, 2017.
- [6] Tehmina Amjad, Ying Ding, Jian Xu, Chenwei Zhang, Ali Daud, Jie Tang, and Min Song. Standing on the shoulders of giants. *Journal of Informetrics*, 11(1):307–323, 2017.
- [7] George Panagopoulos, George Tsatsaronis, and Iraklis Varlamis. Detecting rising stars in dynamic collaborative networks. *Journal of Informetrics*, 11(1):198–222, 2017.

-
- [8] Feng Ding, Yuqing Liu, Xin Chen, and Feng Chen. Rising star evaluation in heterogeneous social network. *IEEE Access*, 6:29436–29443, 2018.
 - [9] Yubing Nie, Yifan Zhu, Qika Lin, Sifan Zhang, Pengfei Shi, and Zhendong Niu. Academic rising star prediction via scholar’s evaluation model and machine learning techniques. *Scientometrics*, 120(2):461–476, 2019.
 - [10] Ali Daud, Min Song, Malik Khizar Hayat, Tehmina Amjad, Rabeeh Ayaz Abbasi, Hassan Dawood, Anwar Ghani, et al. Finding rising stars in bibliometric networks. *Scientometrics*, pages 1–29, 2020.
 - [11] Ali Daud, Naveed ul Islam, Malik Khizar Hayat, Rabeeh Ayaz Abbasi, and Hussain Dawood. Prediction of rising business managers in telecommunication networks. 2020.
 - [12] Ali Daud, Faqir Muhammad, Hassan Dawood, and Hussain Dawood. Ranking cricket teams. *Information Processing & Management*, 51(2):62–73, 2015.
 - [13] Ali Daud, Tehmina Amjad, Tahir Khaliq, and Hussain Dawood. All that glitters is not gold: Falsely predicted rising stars. *Researchpedia Journal of Computing*, page Accepted, 2020.
 - [14] Haseeb Ahmad, Ali Daud, Licheng Wang, Haibo Hong, Hussain Dawood, and Yixian Yang. Prediction of rising stars in the game of cricket. *IEEE Access*, 5:4104–4124, 2017.
 - [15] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.
 - [16] Simon Nusinovici, Yih Chung Tham, Marco Yu Chak Yan, Daniel Shu Wei Ting, Jialiang Li, Charumathi Sabanayagam, Tien Yin Wong, and Ching-Yu Cheng. Logistic regression was as good as machine learning for predicting major chronic diseases. *Journal of clinical epidemiology*, 122:56–69, 2020.
 - [17] Bahzad Charbuty and Adnan Abdulazeez. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01):20–28, 2021.
 - [18] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

-
- [19] Corinna Cortes and Vladimir Vapnik. Support-vector networks machine learning vol. 20, 1995.
- [20] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.
- [21] Maurice Roux. A comparative study of divisive and agglomerative hierarchical clustering algorithms. *Journal of Classification*, 35(2):345–366, 2018.
- [22] Jing-Ying Cai, Fu-Ding Xie, and Yong Zhang. Fuzzy c-means algorithm based on adaptive mahalanobis distances. *Jisuanji Gongcheng yu Yingyong(Computer Engineering and Applications)*, 46(34):174–176, 2010.
- [23] Bahare Kiumarsi, Kyriakos G Vamvoudakis, Hamidreza Modares, and Frank L Lewis. Optimal and autonomous control using reinforcement learning: A survey. *IEEE transactions on neural networks and learning systems*, 29(6):2042–2062, 2017.
- [24] Abhijit Gosavi. Reinforcement learning: A tutorial survey and recent advances. *INFORMS Journal on Computing*, 21(2):178–192, 2009.
- [25] Deepanshu Mehta. State-of-the-art reinforcement learning algorithms. *International Journal of Engineering Research and Technology*, 8:717–722, 2020.
- [26] Suchita V Wawre and Sachin N Deshmukh. Sentiment classification using machine learning techniques. *International Journal of Science and Research (IJSR)*, 5(4):819–821, 2016.
- [27] Roberta A Sinoara, Jose Camacho-Collados, Rafael G Rossi, Roberto Navigli, and Solange O Rezende. Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, 163:955–971, 2019.
- [28] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. Large-scale multi-label text classification on eu legislation. *arXiv preprint arXiv:1906.02192*, 2019.
- [29] WU Yujia, LI Jing, SONG Chengfang, Jun CHANG, et al. Words in pairs neural networks for text classification. *Chinese Journal of Electronics*, 29(3):491–500, 2020.

- [30] Sergio G Burdisso, Marcelo Errecalde, and Manuel Montes-y Gómez. A text classification framework for simple and effective early depression detection over social media streams. *Expert Systems with Applications*, 133:182–197, 2019.
- [31] Yujia Wu, Jing Li, Jia Wu, and Jun Chang. Siamese capsule networks with global and local features for text classification. *Neurocomputing*, 390:88–98, 2020.
- [32] Olusola Abayomi-Alli, Sanjay Misra, Adebayo Abayomi-Alli, and Modupe Odusami. A review of soft techniques for sms spam classification: Methods, approaches and applications. *Engineering Applications of Artificial Intelligence*, 86:197–212, 2019.
- [33] Tian Xia and Xuemin Chen. A discrete hidden markov model for sms spam detection. *Applied Sciences*, 10(14):5011, 2020.
- [34] Tian Xia. A constant time complexity spam detection algorithm for boosting throughput on rule-based filtering systems. *IEEE Access*, 8:82653–82661, 2020.
- [35] Yiyang Chen and Yingwei Zhou. Machine learning based decision making for time varying systems: Parameter estimation and performance optimization. *Knowledge-Based Systems*, 190:105479, 2020.
- [36] José A Castellanos-Garzón, Ernesto Costa, Juan M Corchado, et al. An evolutionary framework for machine learning applied to medical data. *Knowledge-Based Systems*, 185:104982, 2019.
- [37] Ebru Aydındag Bayrak, Pinar Kırıcı, and Tolga Ensari. Comparison of machine learning methods for breast cancer diagnosis. In *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, pages 1–3. IEEE, 2019.
- [38] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.
- [39] Paul Fearnhead and Benjamin Matthew Taylor. On estimating the ability of nba players. *Journal of quantitative analysis in sports*, 7(3), 2011.
- [40] Sameer K Deshpande and Shane T Jensen. Estimating an nba player’s impact on his team’s chances of winning. *Journal of Quantitative Analysis in Sports*, 12(2):51–72, 2016.

- [41] FAISAL ASGHAR, MUBEEN ASIF, MUHAMMAD ATHAR NADEEM, MUHAMMAD ASIM NAWAZ, and MUHAMMAD IDREES. A novel approach to ranking national basketball association players. *Journal of Global Economics, Management and Business Research*, pages 176–183, 2018.
- [42] Jeremy Koster and Brandy Aven. The effects of individual status and group performance on network ties among teammates in the national basketball association. *PloS one*, 13(4):e0196013, 2018.
- [43] Shaoliang Zhang, Alberto Lorenzo, Changjing Zhou, Yixiong Cui, Bruno Gonçalves, and Miguel Angel Gómez. Performance profiles and opposition interaction during game-play in elite basketball: evidences from national basketball association. *International Journal of Performance Analysis in Sport*, 19(1):28–48, 2019.
- [44] Matthew van Bommel and Luke Bornn. Adjusting for scorekeeper bias in nba box scores. *Data Mining and Knowledge Discovery*, 31(6):1622–1642, 2017.
- [45] C Soto Valero. Predicting win-loss outcomes in mlb regular season games—a comparative study using data mining methods. *International Journal of Computer Science in Sport*, 15(2):91–112, 2016.
- [46] Brandon Tolbert and Theodore Trafalis. Predicting major league baseball championship winners through data mining. *Athens Journal of Sports*, 3(4):239–252, 2016.
- [47] Michael Hamilton, Phuong Hoang, Lori Layne, Joseph Murray, David Padget, Corey Stafford, and Hien Tran. Applying machine learning techniques to baseball pitch prediction. In *ICPRAM*, pages 520–527, 2014.
- [48] Tae Young Yang and Tim Swartz. A two-stage bayesian model for predicting winners in major league baseball. *Journal of Data Science*, 2(1):61–73, 2004.
- [49] Gregory Donaker. Applying machine learning to mlb prediction & analysis. CS229—*Stanford University*, 2005.
- [50] Tim Elfrink. Predicting the outcomes of mlb games with a machine learning approach. *Business Analytics Research Paper*, 2018.
- [51] Randy Jia, Chris Wong, and David Zeng. Predicting the major league baseball season. *CS229 FINAL PROJECT*, 2013.

-
- [52] Marcus Bendtsen. Regimes in baseball players' career data. *Data mining and knowledge discovery*, 31(6):1580–1621, 2017.
- [53] Richard J Paulsen. Peer effects and human capital accumulation: Time spent in college and productivity in the national basketball association. *Managerial and Decision Economics*, 2022.
- [54] Geoffrey M Minett, Valentin Fels-Camilleri, Joshua J Bon, Franco M Impellizzeri, and David N Borg. Peer presence increases session ratings of perceived exertion. *International Journal of Sports Physiology and Performance*, 17(1):106–110, 2021.
- [55] Mariia Molodchik, Sofiia Paklina, and Petr Parshakov. Peer effects on individual performance in a team sport. *Journal of Sports Economics*, 22(5):571–586, 2021.
- [56] Leander Forcher, Stefan Altmann, Leon Forcher, Darko Jekauc, and Matthias Kempe. The use of player tracking data to analyze defensive play in professional soccer—a scoping review. *International Journal of Sports Science & Coaching*, page 17479541221075734, 2022.
- [57] Diogo Coutinho, Bruno Gonçalves, Timo Laakso, and Bruno Travassos. Clustering ball possession duration according to players' role in football small-sided games. *Plos one*, 17(8):e0273460, 2022.
- [58] Vishnusai Viswajith Tharoor and NM Dhanya. Performance of indian cricket team in test cricket: A comprehensive data science analysis. In *2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC)*, pages 128–133. IEEE, 2022.
- [59] Udayabhanu NPG Raju et al. Getting useful information from cricket data set using data analytics techniques odi cricket team performance analysis using data mining classification techniques. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(9):684–693, 2021.
- [60] Vangelis Sarlis, Vasilis Chatziilias, Christos Tjortjis, and Dimitris Mandalidis. A data science approach analysing the impact of injuries on basketball player and team performance. *Information Systems*, 99:101750, 2021.
- [61] Vangelis Sarlis and Christos Tjortjis. Sports analytics—evaluation of basketball players and team performance. *Information Systems*, 93:101562, 2020.

- [62] Victor Chazan Pantzalis and Christos Tjortjis. Sports analytics for football league table and player performance prediction. In *2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–8. IEEE, 2020.
- [63] Krzysztof Trawinski. A fuzzy classification system for prediction of the results of the basketball games. In *International conference on fuzzy systems*, pages 1–7. IEEE, 2010.
- [64] Francisco JR Ruiz and Fernando Perez-Cruz. A generative model for predicting outcomes in college basketball. *Journal of Quantitative Analysis in Sports*, 11(1):39–52, 2015.
- [65] Ping-Feng Pai, Lan-Hung ChangLiao, and Kuo-Ping Lin. Analyzing basketball games by a support vector machines with decision tree model. *Neural Computing and Applications*, 28(12):4159–4167, 2017.
- [66] Miguel A Gómez, Sergio J Ibáñez, Isabel Parejo, and Philip Furley. The use of classification and regression tree when classifying winning and losing basketball teams. *Kinesiology: International journal of fundamental and applied kinesiology*, 49(1):47–56, 2017.
- [67] Yongjun Li, Lizheng Wang, and Feng Li. A data-driven prediction approach for sports team performance and its application to national basketball association. *Omega*, page 102123, 2019.
- [68] Fadi Thabtah, Li Zhang, and Neda Abdelhamid. Nba game result prediction using feature analysis and machine learning. *Annals of Data Science*, 6(1):103–116, 2019.
- [69] Jian Shi and Kai Song. A discrete-time and finite-state markov chain based in-play prediction model for nba basketball matches. *Communications in Statistics-Simulation and Computation*, pages 1–9, 2019.
- [70] Mei-Ling Huang and Yun-Zhi Li. Use of machine learning and deep learning to predict the outcomes of major league baseball matches. *Applied Sciences*, 11(10):4499, 2021.
- [71] Chia-Hao Chang. Construction of a predictive model for mlb matches. *Forecasting*, 3(1):102–112, 2021.

- [72] VK Harikrishnan, Harshal Deore, Pavan Raju, Akshat Agrawal, and Mayank Mohan Sharma. Predictive analysis using machine learning techniques for fantasy games. In *Advances in Mechanical Engineering*, pages 683–692. Springer, 2021.
- [73] Aliaa Saad Yaseen, Ali Fadhil Marhoon, and Sarmad Asaad Saleem. Multimodal machine learning for major league baseball playoff prediction. *Informatica*, 46(6), 2022.
- [74] Shu-Fen Li, Mei-Ling Huang, and Yun-Zhi Li. Exploring and selecting features to predict the next outcomes of mlb games. *Entropy*, 24(2):288, 2022.
- [75] Riccardo Ievoli, Lucio Palazzo, and Giancarlo Ragozini. On the use of passing network indicators to predict football outcomes. *Knowledge-Based Systems*, 222:106997, 2021.
- [76] Ryan Beal, Stuart E Middleton, Timothy J Norman, and Sarvapali D Ramchurn. Combining machine learning and human experts to predict match outcomes in football: A baseline model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15447–15451, 2021.
- [77] Halvard Arntzen and Lars Magnus Hvattum. Predicting match outcomes in association football using team ratings and player ratings. *Statistical Modelling*, 21(5):449–470, 2021.
- [78] Chananyu Pipatchatchawal and Suphakant Phimoltares. Predicting football match result using fusion-based classification models. In *2021 18th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6. IEEE, 2021.
- [79] Usman Haruna, Jaafar Zubairu Maitama, Murtala Mohammed, and Ram Gopal Raj. Predicting the outcomes of football matches using machine learning approach. In *International Conference on Informatics and Intelligent Applications*, pages 92–104. Springer, 2021.
- [80] Ondřej Hubáček, Gustav Šourek, and Filip železný. Forty years of score-based soccer match outcome prediction: an experimental review. *IMA Journal of Management Mathematics*, 33(1):1–18, 2022.

- [81] Shuo Guan and Xiaochen Wang. Optimization analysis of football match prediction model based on neural network. *Neural Computing and Applications*, 34(4):2525–2541, 2022.
- [82] Syasya Nadhilah Maozad, Siti Noor Asyikin Mohd Razali, Aida Mustapha, Aziz Nanthamornphong, Mohd Helmy Abdul Wahab, and Nazim Razali. Comparative analysis for predicting football match outcomes based on poisson models. In *2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–4. IEEE, 2022.
- [83] SK Nivetha, M Geetha, RC Suganthe, R Manoj Prabakaran, S Madhuvanan, and A Mohamed Sameer. A deep learning framework for football match prediction. In *2022 International Conference on Computer Communication and Informatics (IC-CCI)*, pages 1–7. IEEE, 2022.
- [84] Fátima Rodrigues and Ângelo Pinto. Prediction of football match results with machine learning. *Procedia Computer Science*, 204:463–470, 2022.
- [85] Rajesh Goel, Jerryl Davis, Amit Bhatia, Pulkit Malhotra, Harsh Bhardwaj, Vikas Hooda, and Ankit Goel. Dynamic cricket match outcome prediction. *Journal of Sports Analytics*, (Preprint):1–12, 2021.
- [86] Mazhar Javed Awan, Syed Arbaz Haider Gilani, Hamza Ramzan, Haitham Nobanee, Awais Yasin, Azlan Mohd Zain, and Rabia Javed. Cricket match analytics using the big data approach. *Electronics*, 10(19):2350, 2021.
- [87] RR Kamble et al. Cricket score prediction using machine learning. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(1S):23–28, 2021.
- [88] Saurabh Kumar. Predicting the outcome of ipl cricket matches using machine learning. *The Journal of Prediction Markets*, 16(1), 2022.
- [89] Shristi Priya, Ankit Kumar Gupta, Atman Dwivedi, and Aryan Prabhakar. Analysis and winning prediction in t20 cricket using machine learning. In *2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–4. IEEE, 2022.
- [90] Subrat Sarangi and RK Renin Singh. Winning one-day international cricket matches: a cross-team perspective. *Journal of Business Analytics*, pages 1–20, 2022.

-
- [91] Md Aktaruzzaman Pramanik, Md Mahmudul Hasan Suzan, Al Amin Biswas, Mohammad Zahidur Rahman, and A Kalaiarasi. Performance analysis of classification algorithms for outcome prediction of t20 cricket tournament matches. In *2022 International Conference on Computer Communication and Informatics (ICCCI)*, pages 01–07. IEEE, 2022.
- [92] Jun Zhang, Zhaolong Ning, Xiaomei Bai, Wei Wang, Shuo Yu, and Feng Xia. Who are the rising stars in academia? In *2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL)*, pages 211–212. IEEE, 2016.
- [93] Jun Zhang, Feng Xia, Wei Wang, Xiaomei Bai, Shuo Yu, Teshome Megersa Bekele, and Zhong Peng. Cocarank: A collaboration caliber-based method for finding academic rising stars. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 395–400, 2016.
- [94] Lin Zhu, Donghua Zhu, Xuefeng Wang, Scott W Cunningham, and Zhinan Wang. An integrated solution for detecting rising technology stars in co-inventor networks. *Scientometrics*, 121(1):137–172, 2019.
- [95] Yuliang Ma, Ye Yuan, Guoren Wang, Xin Bi, Zhongqing Wang, and Yishu Wang. Rising star evaluation based on extreme learning machine in geo-social networks. *Cognitive Computation*, 12(1):296–308, 2020.
- [96] Rashid Abbasi, Ali Kashif Bashir, Jianwen Chen, Abdul Mateen, Jalil Piran, Farhan Amin, and Bin Luo. Author classification using transfer learning and predicting stars in co-author networks. *Software: Practice and Experience*, 51(3):645–669, 2021.
- [97] Ali Daud, Faizan Abbas, Tehmina Amjad, Abdulrahman A Alshdadi, and Jalal S Alowibdi. Finding rising stars through hot topics detection. *Future Generation Computer Systems*, 115:798–813, 2021.
- [98] Ali Daud, Juanzi Li, Lizhu Zhou, and Faqir Muhammad. Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of computer science in China*, 4(2):280–301, 2010.
- [99] Ali Daud, Naveed ul Islam, Xin Li, Imran Razzak, and Malik Khizar Hayat. Identifying rising stars via supervised machine learning. *IEEE Transactions on Computational Social Systems*, 2022.

- [100] Aftab Nawaz and MSI Malik. Rising stars prediction in reviewer network. *Electronic Commerce Research*, 22(1):53–75, 2022.
- [101] Xuan Yang, Yang Yang, Jintao Su, Yifei Sun, Zhongyao Wang, Shen Fan, Jun Zhang, and Jingmin Chen. Who's next: Rising star prediction via diffusion of user interest in social networks. *IEEE Transactions on Knowledge and Data Engineering*, 2022.
- [102] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.
- [103] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598, 2000.
- [104] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67. Stockholm, Sweden, 1999.
- [105] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.
- [106] Mark Andrew Hall. Correlation-based feature selection for machine learning. 1999.
- [107] Jorge E Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences*, 102(46):16569–16572, 2005.
- [108] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. *Encyclopedia of database systems*, 5:532–538, 2009.
- [109] Johar M Ashfaq and Amer Iqbal. Introduction to support vector machines and kernel methods. *publication at <https://www.researchgate.net/publication/332370436>*, 2019.
- [110] Tehmina Amjad, Ali Daud, Sadia Khan, Rabeeh Ayaz Abbasi, and Faisal Imran. Prediction of rising stars from pakistani research communities. In *2018 14th International Conference on Emerging Technologies (ICET)*, pages 1–6. IEEE, 2018.
- [111] Prabhaker Mishra, Chandra Mani Pandey, Uttam Singh, Amit Keshri, and Mayilvaganan Sabaretnam. Selection of appropriate statistical methods for data analysis. *Annals of cardiac anaesthesia*, 22(3):297, 2019.