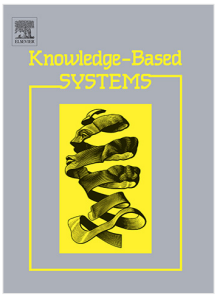
**Journal Pre-proof**




Using machine learning techniques for rising star prediction in basketball

Zafar Mahmood, Ali Daud, Rabeeh Ayaz Abbasi

PII: S0950-7051(20)30635-3
DOI: https://doi.org/10.1016/j.knosys.2020.106506
Reference: KNOSYS 106506

To appear in: *Knowledge-Based Systems*

Received date : 17 June 2020
Revised date : 30 September 2020
Accepted date : 6 October 2020

Please cite this article as: Z. Mahmood, A. Daud and R.A. Abbasi, Using machine learning techniques for rising star prediction in basketball, *Knowledge-Based Systems* (2020), doi: https://doi.org/10.1016/j.knosys.2020.106506.

# Using Machine Learning Techniques for Rising Star Prediction in Basketball

Zafar Mahmood[a,1,*], Ali Daud[b], Rabeeh Ayaz Abbasi[c]

[a]*Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan.*
[b]*Department of Information Systems and Technology, College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia.*
[c]*Department of Computer Science, Quaid-i-Azam University, Islamabad, Pakistan*

**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 their decisions making. Concept of rising star has been applied for finding rising authors in research community, rising business managers in telecommunication industry and rising players in the game of cricket. In this paper 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. The 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 shows 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.

*Keywords:* Rising Star,Machine Learning,Classification,Prediction,Ranking

*Corresponding Author:
*Email address:* alimsdb@gmail.com (Ali Daud)

## 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.

There has been some work done in finding the experts in author networks. PubRank algorithm [1] explored rising stars in author's network. StarRank algorithm is presented by [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 [3], they define 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 [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 [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 [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 [7] for extracting and analyzing profiles of scholars. Inner factors were used by [8] to evaluate rising stars in a heterogeneous social network. The influence of co-authors and the quality cited papers have been used by [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 [11]. Ranking of cricket teams based on ranking algorithms is presented by [12]. Reference [13] discuss about falsely predicted rising stars. In sports, the concept of the rising star has only been applied to the game of cricket [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.

Predicting rising stars in the game of basketball is a machine learning problem. For a given player and set of features the objective is to predict whether a

player is a rising star or not. Prior to predict rising stars we labeled players as rising stars and not rising stars on the basis of their efficiency. After labeling the players different basketball statistics of the co-players are used as attributes to machine learning models.

We introduced three types of co-players, players of same team in same game, player of opponent team in same game and combination of players of same and opponent team appeared in same game. The objective of proposing these three types of co-players is to explore which type of co-players features achieve better accuracy. We used existing machine learning models and improved the accuracy of these models through feature engineering. We classified features by type and by size and also further categories of these features were created. To conduct experiment we created three datasets from the scratch. Extensive experiments are done with three dataset in order to examine the significance of different categories of features and machine learning models. Rising stars are also compared with NBA Most Improved Players.

To the best of our knowledge this is the first attempt of predicting rising stars in game of basketball. In other sports the rising star prediction is applied in the game of cricket [14], they consider those players as co-players who play in some common time span, the limitation of their concept of co-player is that the co-players who played in some common time span might be possible not appeared in same games as well. We consider only those players as co-players who appeared in the actual games played by rising star players.

Main contributions of this paper are:

- 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.

- 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.

- The three datasets used in this study are constructed by us and are not available anywhere. Each dataset consists of player name, different features of co-players and class as rising star or not rising star. The first dataset that we call Dataset A consist features of those co-players that

belong to the player team. The second dataset that we call Dataset B consists of features of those co-players that belong to opponent team. The third type of dataset that we call Dataset C consists of features of co-players that belong to both same and opponent team. In future we will provide these datasets on GitHub or Kaggle forum for experiment and research purposes.

The rest of the paper is organized as, Sec 2 discuss basic concepts of basketball and Sec 3 discuss the related work. 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 4. Details of experiments is discussed in Sec 5. Sec 6 discuss conclusion and future work.

## 2. Basic Concepts of Basketball

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

### *2.1. 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.

### *2.2. 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.

### *2.3. 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.

### *2.4. Player Statistics*

Basic game statistics[1] of a basketball player are given in Table.1

[1] https://www.breakthroughbasketball.com/stats/definitions.html

Table 1: Basic Game Statistics of Baseketball Player

| S.No | Attribute | Definition |
|---|---|---|
| 1 | Field Goals (FG) | These are the shots by a player that goes through the basket from above. |
| 2 | Average Field Goals(avg_FG) | This is simply the division of Field Goals by total number of games played by a player. |
| 3 | Field Goal Attempts(FGA) | Number of attempts by a player to score Field Goal. |
| 4 | Average Field Goal Attempts(avg_FGA) | This can be calculated dividing Field Goal Attempts on total number of games played by a player. |
| 5 | Field Goals Percent(FGper) | It shows the shooting ability of a player. FGper can be obtained through dividing Field Goals by Field Goal Attempts. |
| 6 | Three Points(3PT) | Three points are awarded to a payer when he shots the ball from long distance and ball goes in the basket. |
| 7 | Average Three Points(avg_3PT) | Average of three points can be obtained by dividing player's total three points on number of games played by player. |
| 8 | Three Point Attempts(3PTA) | This is the number of attempts to score Three Points. |
| 9 | 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. |
| 10 | Three Point Percent(3PTper) | It is the measurement of long distance shooting ability of a player. 3PTper can calculated by dividing Three points on Three Point Attempts. |
| 11 | Free Throws(FT) | These are shots made from free throw line. One point is awarded for a Free Throw. |
| 12 | 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. |
| 13 | Free Throw Attempts(FTA) | Free Throw Attempts are the number of time a player attempted to make a free throw. |
| 14 | 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. |
| 15 | 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. |
| 16 | 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. |
| 17 | Average Offensive Rebounds(avg_OREB) | Average Offensive Rebounds are obtained by dividing total number of Offensive Rebound of player on total number of games played by a player. |
| 18 | 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. |
| 19 | Average Defensive Rebounds(avg_DREB) | It is obtained by dividing total number of Defensive Rebounds of a player on total number of games played by a player. |
| 20 | Rebounds(REB) | Player Rebounds are the sum of player's Offensive and Defensive Rebounds. |
| 21 | Average Rebounds(avg_REB) | Average Rebounds of a player are obtained by dividing player's total Rebound on player's total number of games. |
| 22 | Assists(AST) | Assist is awarded to a player who pass the ball to a player of same team who shot the ball in basket. |
| 23 | 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. |
| 24 | 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. |
| 25 | Average Turnover(avg_TOV) | Average Turover is obtained by dividing total number of Turnovers on total number of games played by a player. |
| 26 | Blocks(BLK) | A block occurs when a defensive player diverts a shot attempt of an offensive player. |
| 27 | Average Blocks(avg_BLK) | Average Blocks is obtained by dividing total number of Block on total number of games played by a player. |
| 28 | Personal Fouls(PF) | A personal foul is any violation of the rules of the game that involves personal contact with an opposing player. |
| 29 | 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. |
| 30 | Points(PTS) | Points are scored when a player puts the ball through the basket. The number of points awarded is determined by the type of shot and the distance. |
| 31 | 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. Related Work

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.

### *3.1. Ranking Basketball Players*

NBA game statistics like points, blocks, rebounds, field goals etc [2] are widely used for rating the basketball players. John Hollinger (per) 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 [15]. 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 [16], they used a bayesian linear regression model for finding an individual player impact on the team winning. The said research ranks the players with respect to their team and across the leagues.

slack based measure method is used by [17] to rank players in NBA. They 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. Authors in [18] shows how network ties among players are affected through individual status and group performance. Relationship between game statistics and match outcome is explored by [19]. They also consider how player's technical and physical performance is affected by interaction of opposition.

### *3.2. Match Outcome Prediction*

The fuzzy rule-based system (FRBS) is proposed by [20] 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 [21] that combined a simple soccer model and poisson factorization. The simple soccer model identifies each team by its attack and defense 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 HSVMDT(Hybrid Support Vector Machine and Decision Tree) is proposed by [22]. 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

[2] http://www.espn.com/editors/nba/glossary.html

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 [23] to find best predictor in order to classify teams as winning or loosing 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 [24] 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 [25] 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 [26] to predict the outcome of the match when the game is in progress. The aim of the said model is to model 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.

### *3.3. Applications of Machine Learning Techniques*

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

SVM [27] and Naive Bayes [28] techniques have been used by [29] for classification of movie reviews. Text classification based on document embedding is used by [30]. One of the application of machine learning in the domain of legal documents is presented by [31], where the authors applied various models for multi-label text classification on legislation documents. Words in pair neural networks is presented by [38] for text classification that overcome the limitation of text classification based on single word with multiple meanings. Novel machine learning model SS3 proposed by [32] 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 [40]

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 [33]. Discrete Hidden Markov Model is used by [34] 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 [35].

classical machine learning technique are not much efficient in situations where the decisions are time-dependent, for such situations, [36] 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 [37]. For prediction of breast cancer, Support Vector Machines and Artificial Neural Networks has been applied by [39] on Wisconsin Breast Cancer dataset.

## 4. Problem and Proposed Method

### *4.1. Problem Definition*

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

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

The set of Players is defined by vector P.
$P = \{Player_1, Player_2, Player_3, ..., Player_m\}$

The corresponding set of attributes or features is defined by vector F.
$F = \{co\_F_1, co\_F_2, co\_F_3, ..., co\_F_n\}$

Set of labels are define by vector Y.
$Y = \{y_1, y_2, y_3, ..., y_m\}$
where
$y_1, y_2, y_3, ..., y_m \in \{RisingStar, Not\text{-}RisingStar\}$

For a give feature vector x:
$x = \{x_1, x_2, x_3, ..., x_n\}$
where
$x_1 \in co\_F_1, x_2 \in co\_F_2, x_3 \in co\_F_3, ..., x_n \in co\_F_m$

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:

$$f: F \rightarrow Y$$

### *4.2. Proposed Method (Machine Learning Techniques)*

As we see in previous section, that our objective is to find a function that can predict class label as rising star or not rising star for a give feature set. We used machine learning techniques that are capable of predicting such tasks.

#### *4.2.1. CLASSIFICATION AND REGRESSION TREES (CART)*

CART [41] 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.

#### *4.2.2. SUPPORT VECTOR MACHINES (SVM)*

SVM [27] 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.

#### *4.2.3. MAXIMUM ENTROPY MARKOV MODEL (MEMM)*

MEMM [42] is a sequence modeling algorithm. It extends the famous MEC (maximum entropy classifier)[43] with the feature that is; unknown parameters are assumed to be connected in a markov chain rather than independent to each other.

#### *4.2.4. BAYESIAN NETWORK (BN)*

Bayesian network [44] 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.

#### *4.2.5. NAÏVE BAYES (NB)*

Working of naïve bayes classifier[28] is based on Bayesian theorem. Naïve bayes consider that there is no dependency between the features.

*4.3. 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 1: 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.1 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.2 represent players of two teams who played a game. In

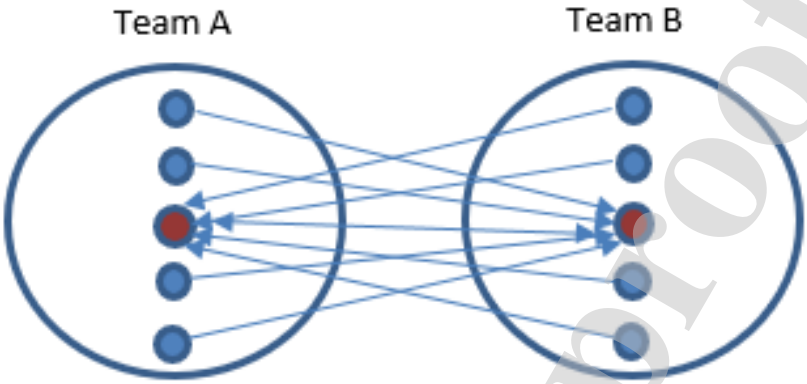



Figure 2: Players of opposite team and same game.

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.

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 represent players of two teams who played

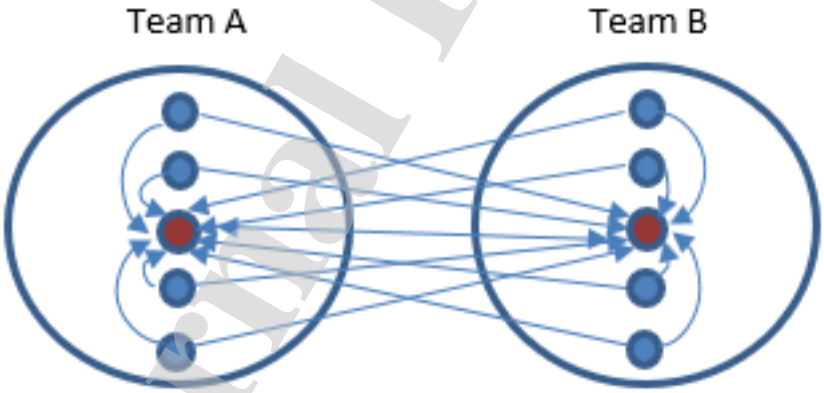



Figure 3: Players of same and opposite team in same game.

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.

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.4,Fig.5 and Fig.6 show the relationship between rising stars and their co-players. In these figures we can see how efficiency of rising star players is correlated to their co-players. In Fig.4 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 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.

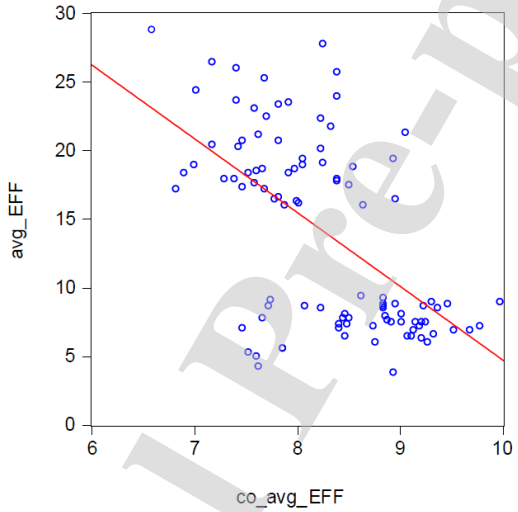



Figure 4: Relationship between co-players(same team) and rising star players.
(x-axis represents average efficiency of co-players. y-axis represents average-efficiency of players labeled as rising and not-rising stars)

Fig.5 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.

Fig.6 shows that there is a week 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.

### *4.4. 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

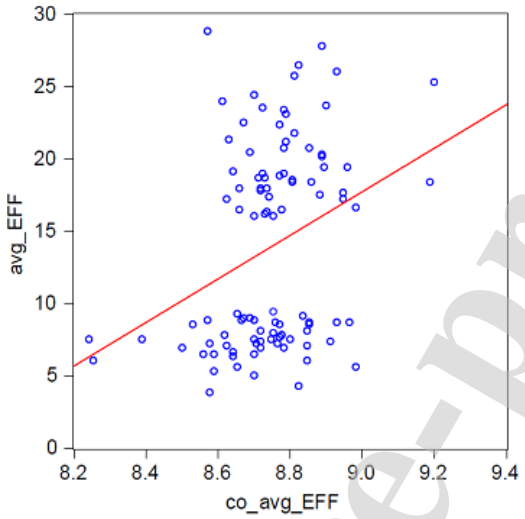



Figure 5: Relationship between co-players(opponent team) and rising star players.
(x-axis represents average efficiency of co-players. y-axis represents average-efficiency of players labeled as rising and not-rising stars)

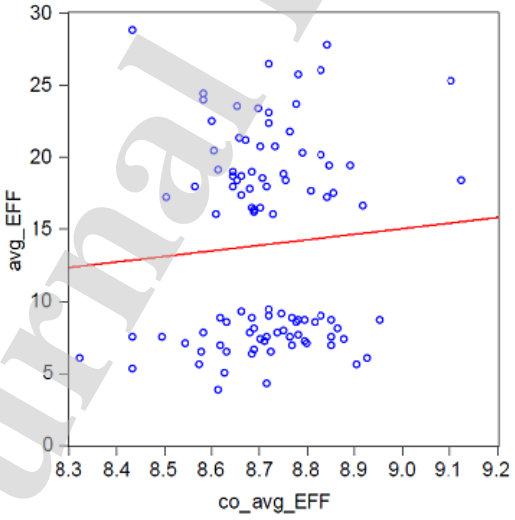



Figure 6: Relationship between co-players(both same and opponent team) and rising star players.
(x-axis represents average efficiency of co-players. y-axis represents average-efficiency of players labeled as rising and not-rising stars)

features by type and size Fig.7 shows the pictorial representation of features classification.

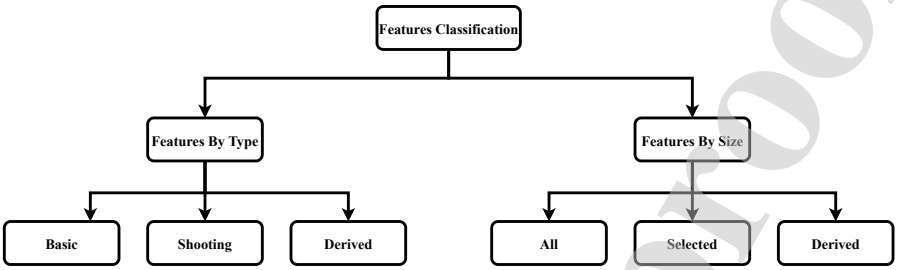



Figure 7: Features Classification.

*4.4.1. Features by Type*

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

*Basic Features.* The basic features contain rebounds, assists, turnovers, blocks, fouls and points. Table.2 shows different number of basic features.

*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.2 shows different number of shooting features.

*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.2. Sec 4.5 give a detailed explanation and formulation of derived features.

*4.4.2. 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.

*All Features.* All features are the combination of basic, shooting and derived features. All feature set consists of 47 features. Table.3 shows a list of all features

*Selected Features.* Since the full feature set consists of 47 features. Correlation-based Feature Subset Selection technique [45] 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.4 shows a list of selected features for each dataset.

Table 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 begining of Season |
| 6 | avg_REB | Average Rebound | 3PT | Three Points | avg_EFF_begg | Average Efficiency at Begining 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-indexs fo co-players |
| 12 | avg_TOV | Average Turn Overs | avg_FT | Average Free Throws | avg_Co_H-index | Average H-index of co-players |
| 13 | PF | Personal Fouls | FTA | Free Thorw Attempts | HGS | Hollinger Score |
| 14 | avg_PF | Average Personal Fouls | avg_FTA | Average Free Throw Attempts | avg_HGS | Average Hollinger Score |
| 15 | PTS | Points | FTper | Free Throw Percentage | 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: All Features

| S.No | Feature | Decription | S.No | Feature | Description | S.No | Feature | Description |
|---|---|---|---|---|---|---|---|---|
| 1 | co_inf | Co-player influence | 17 | FTper | Free Throw Percentage | 33 | avg_PTS | Average Points |
| 2 | avg_co_inf | Average Co-player influence | 18 | OREB | Offensive Rebounds | 34 | EFF | Efficiency |
| 3 | FG | Field Goals | 19 | avg_OREB | Average Offensive Rebounds | 35 | avg_EFF | Average Efficiency |
| 4 | avg_FG | Average Field Goals | 20 | DREB | Defensive Rebounds | 36 | EFF_begg | Efficiency at Beginning of Season |
| 5 | FGA | Field Goal Attempts | 21 | avg_DREB | Average Rebounds | 37 | avg_EFF_begg | Average Efficiency at Beginning of Season |
| 6 | avg_FGA | Average Field Goal Attempts | 22 | REB | Rebounds | 38 | EFF_mid | Efficiency at Mid of Season |
| 7 | FGper | Field Goals Percentage | 23 | avg_REB | Average Rebounds | 39 | avg_EFF_mid | Average Efficiency at Mid of Season |
| 8 | 3PT | Three Points | 24 | AST | Assists | 40 | EFF_end | Efficiency at End of Season |
| 9 | avg_3PT | Average Three Points | 25 | avg_AST | Average Assists | 41 | avg_EFF_end | Average Efficiency at End of Season |
| 10 | 3PTA | Three Point Attempts | 26 | BLK | Blocks | 42 | Co_H-index | sum of H-indexs of co-players |
| 11 | avg_3PTA | Average Three Point Attempts | 27 | avg_BLK | Average Blocks | 43 | avg_Co_H-index | Average H-index of co-players |
| 12 | 3PTper | Three Points Percentage | 28 | TOV | Turn Overs | 44 | HGS | Hollinger Score |
| 13 | FT | Free Throws | 29 | avg_TOV | Average Turn Overs | 45 | avg_HGS | Average Hollinger Score |
| 14 | avg_FT | Average Free Throws | 30 | PF | Personal Fouls | 46 | Points_share | Co-player Points divided by Team Points |
| 15 | FTA | Free Throw Attempts | 31 | avg_PF | Average Personal Fouls | 47 | avg_Points_share | Average Points Share of Co-players |
| 16 | avg_FTA | Average Free Throw Attempts | 32 | PTS | Points | - | - | - |

Table 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 5: Derived Features

| S.No | Feature | Description | S.No | Feature | Description |
|---|---|---|---|---|---|
| 1 | co_inf | Co-Player Influece | 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 | Average 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 |

*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.5 shows a list of derived features.

### *4.5. Mathematical Formulation of Derived Features*

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

#### *4.5.1. Co-Player Influence*

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

$$co\_inf(Player, co) = \frac{G\_P}{Co\_G}. \tag{1}$$

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

$$co\_inf = \sum_{i=1}^{n} co\_inf\_P(i). \tag{2}$$

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. Fig.A.18 shows example of Co-Player Influence.

#### *4.5.2. 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.

$$avg\_co\_inf = \sum_{i=1}^{n} \frac{co\_inf\_P(i)}{Co\_N}. \tag{3}$$

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

#### *4.5.3. Co-Players Efficiency*

Efficiency of basketball players depends on various factor. We use NBA efficiency formulae[3] to find the efficiency of co-players of a player. This formula is given by the following equation.

$$CO\_EFF = (PTS + REB + AST + STL + BLK - FGA - FTA - TOV). \tag{4}$$

[3]https://www.nbastuffer.com/analytics101/nba-efficiency/

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.

$$EFF = \sum_{i=1}^{n} Co\_EFF(i). \tag{5}$$

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

*4.5.4. 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.

$$avg\_EFF = \sum_{i=1}^{n} \frac{Co\_EFF(i)}{Co\_N}. \tag{6}$$

*4.5.5. 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.
$BD_1, BD_2 \ldots BDn, MD_1, MD_2 \ldots MD_n, ED_1 \ldots ED_n$.
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.

$$\begin{aligned} EFF\_Begg = (&PTS + REB + AST + STL + BLK - \\ &FGA - FTA - TOV). \\ &Where \;\; Game_{Date<MD_1} \end{aligned} \tag{7}$$

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

$$EFF\_Begg = \sum_{i=1}^{n} Co\_EFF\_Begg(i). \tag{8}$$

*4.5.6. 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

$$avg\_EFF\_Begg = \sum_{i=1}^{n} \frac{Co\_EFF\_Begg(i)}{Co\_N}. \tag{9}$$

*4.5.7. 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.

$$\begin{aligned} EFF\_Mid = (PTS + REB + AST + STL + BLK - \\ FGA - FTA - TOV). \\ Where \;\; Game_{Date > BD_n} and \;\; Game_{Date < ED_1} \end{aligned} \tag{10}$$

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

$$EFF\_Mid = \sum_{i=1}^{n} Co\_EFF\_Mid(i). \tag{11}$$

*4.5.8. 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

$$avg\_EFF\_Mid = \sum_{i=1}^{n} \frac{Co\_EFF\_Mid(i)}{Co\_N}. \tag{12}$$

*4.5.9. 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

$$\begin{aligned} EFF\_Beg = (PTS + REB + AST + STL + BLK - \\ FGA - FTA - TOV). \\ Where \;\; Game_{Date > ED_n} \end{aligned} \tag{13}$$

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

$$EFF\_End = \sum_{i=1}^{n} Co\_EFF\_End(i). \tag{14}$$

*4.5.10. 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.

$$avg\_EFF\_End = \sum_{i=1}^{n} \frac{Co\_EFF\_End(i)}{Co\_N}. \tag{15}$$

*4.5.11. Co-Players H-index*

H-index [46] 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

$$Co\_H\text{-}index = \sum_{i=1}^{n} H\text{-}index(i). \tag{16}$$

Fig.A.20 shows example to calculate H-index.

*4.5.12. Average H-Index of Co-Players*

To find average h-index of co-players of a player P, h-index 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.

$$avg\_Co\_H\text{-}index = \sum_{i=1}^{n} \frac{H\text{-}index(i)}{Co\_N}. \tag{17}$$

*4.5.13. Hollinger Score (HGS)*

We use Hollinger linear formula[4] to calculate co-players efficiency, the formula is given by

$$\begin{aligned} CO\_HGS = (PTS) + 0.4 * (FG) + 0.7 * (OREB) + 0.3* \\ (DREB) + (STL) + 0.7 * (AST) + 0.7 * (BLK) - 0.7* \\ (FGA) - 0.4 * (FTA) - 0.4 * (PF) - (TOV). \end{aligned} \tag{18}$$

[4]https://www.nbastuffer.com/analytics101/game-score/

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

$$HGS = \sum_{i=1}^{n} Co\_HGS(i). \tag{19}$$

*4.5.14. 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.

$$avg\_HGS = \sum_{i=1}^{n} \frac{Co\_HGS(i)}{Co\_N}. \tag{20}$$

*4.5.15. 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.

$$Co\_Points\_Share = \frac{PTS}{Team\_PTS}. \tag{21}$$

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

$$Points\_Share = \sum_{i=1}^{n} Co\_Points\_Share(i). \tag{22}$$

400

*4.5.16. 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.

$$avg\_Point\_Share = \sum_{i=1}^{n} \frac{Co\_Points\_Share(i)}{Co\_N}. \tag{23}$$

## 5. Experiments

### *5.1. Datasets*

Flowchart in Fig.A.22 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 website[5]. Necessary data preprocessing was done using MySQL queries and Python Pandas library. The initial data statistics are:

[5] www.espn.com/nba/

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

$$EFF = (PTS + REB + AST + STL + BLK - FGA - FTA - TOV). \tag{24}$$

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

$$avg\_EFF = \frac{EFF}{Total\ Games\ Played}. \tag{25}$$

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.

*5.1.1. Dataset A*

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

*5.1.2. Dataset B*

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

*5.1.3. Dataset C*

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

*5.2. 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.8 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.8 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.9 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.9 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.8 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.8 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.8 we can see that rising stars are closer to 1 as compared to not rising stars.

If we look at features distribution in Fig.8 and Fig.9, we can see that for most of the features in Fig.8 the feature values are dominant for rising stars up to some extent and after that not rising stars are dominant, whereas in Fig.9 we can see that rising stars are dominant till the end. The reason for the sudden change in the bar graph of Fig.8 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.9 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.9 that there is no sudden change in the bar graph and rising stars are dominant for all features. Fig.10 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.10 we can observe that more feature values

belong to rising stars. Just like in Fig.8, we can see in Fig.10 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.

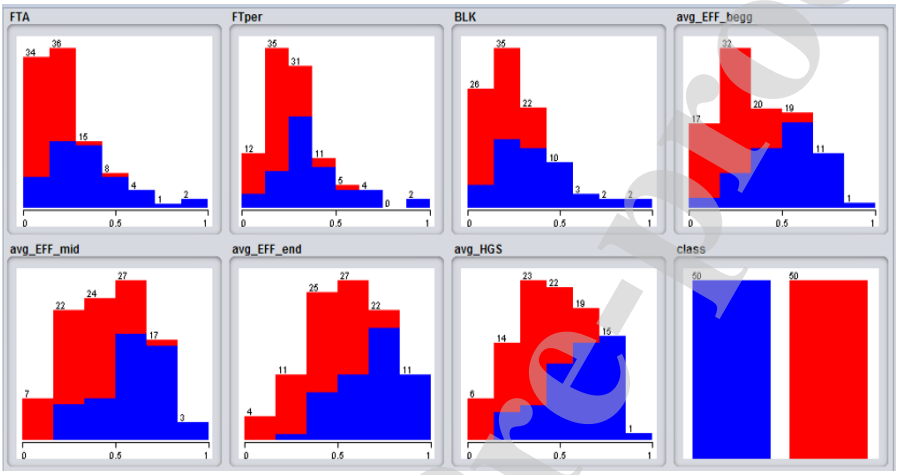



Figure 8: FEATURES DISTRIBUTION OF SELECTED FEATURES FOR DATASET A. (Red color shows rising star and blue color shows not-rising star. X-axis shows distribution of values and Y-axis shows frequency of values for an attribute.)

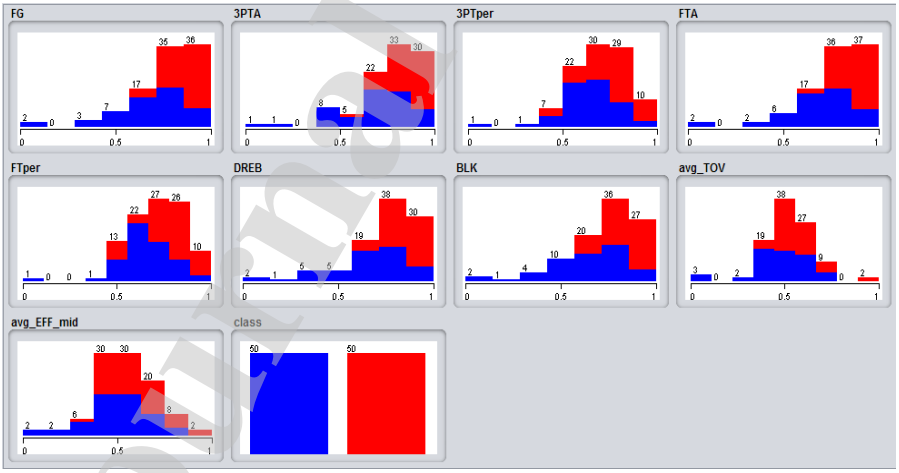



Figure 9: FEATURES DISTRIBUTION OF SELECTED FEATURES FOR DATASET B.

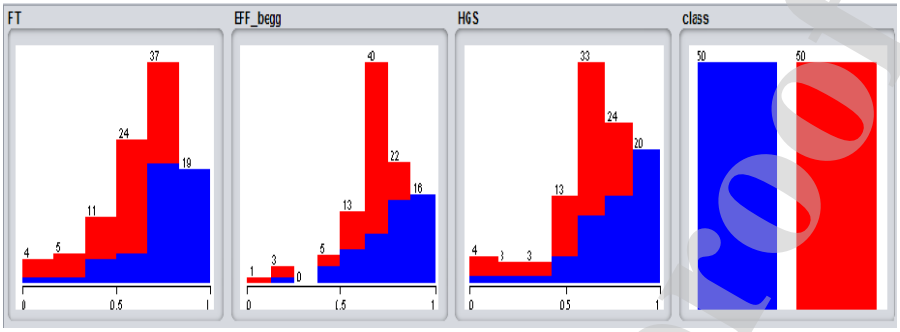



Figure 10: FEATURES DISTRIBUTION OF SELECTED FEATURES FOR DATASET C.

### *5.3. 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. In Table.6 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.7 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.8 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.

Table 6: 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 |

### *5.4. Performance Evaluation*

To measure the strength of classifiers and features, a 10-fold cross validation method is used to train and validate the classifiers using all the three datasets.

Table 7: 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 8: 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 |

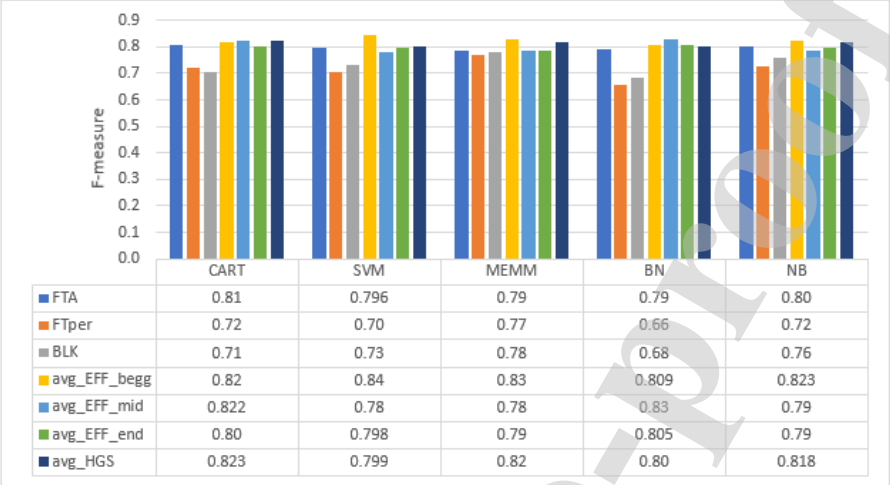
To find the impact of each feature on rising star prediction, each classifier is trained while exploiting each feature of the selected feature set of each dataset. This process is done in 7,9 and 3 cycles for Dataset A, Dataset B and Dataset C respectively. Each dataset consist of 100 labeled samples is divided into 10 equal parts, such as 10,20, 30. . . 100. All classifiers are trained for each partition. Precision, F-measure and Recall are computed for each dataset. We only present average F-measure for analysis of the results. The evaluations presented in this paper are the average of 10 observations of F-measure. All experiments are performed using open source software WEKA.

### *5.5. Results and Discussion*

#### *5.5.1. Individual Feature Analysis*

This analysis shows the strength of an individual feature on rising star prediction. We only used individual features from each of the selected feature set for their respective dataset.

*Individual Feature Analysis of Dataset A.* Fig.11 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



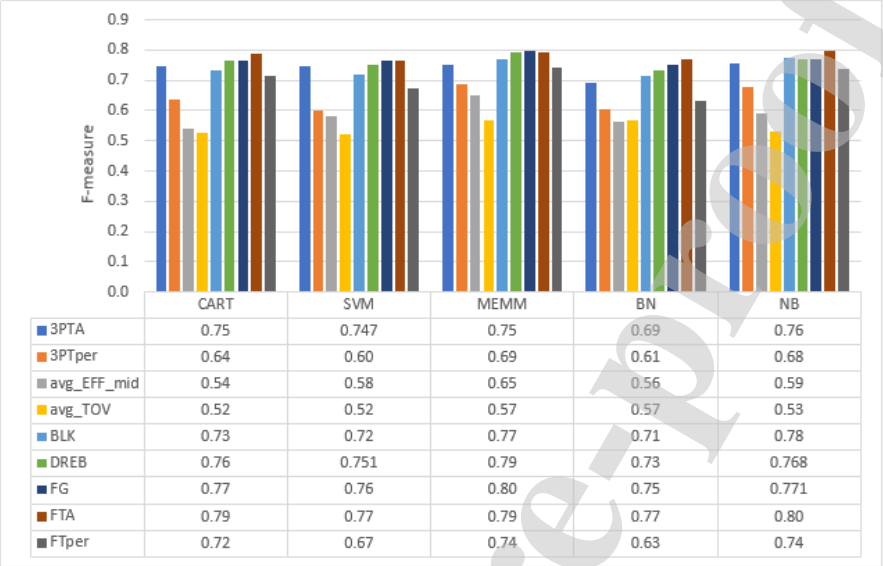

| | CART | SVM | MEMM | BN | NB |
|---|---|---|---|---|---|
| FTA | 0.81 | 0.796 | 0.79 | 0.79 | 0.80 |
| FTper | 0.72 | 0.70 | 0.77 | 0.66 | 0.72 |
| BLK | 0.71 | 0.73 | 0.78 | 0.68 | 0.76 |
| avg_EFF_begg | 0.82 | 0.84 | 0.83 | 0.809 | 0.823 |
| avg_EFF_mid | 0.822 | 0.78 | 0.78 | 0.83 | 0.79 |
| avg_EFF_end | 0.80 | 0.798 | 0.79 | 0.805 | 0.79 |
| avg_HGS | 0.823 | 0.799 | 0.82 | 0.80 | 0.818 |

Figure 11: INDIVIDUAL FEATURE ANALYSIS OF DATASET A.

feature and gives the lowest score of 66% using FTper feature. Naïve Bayes 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.

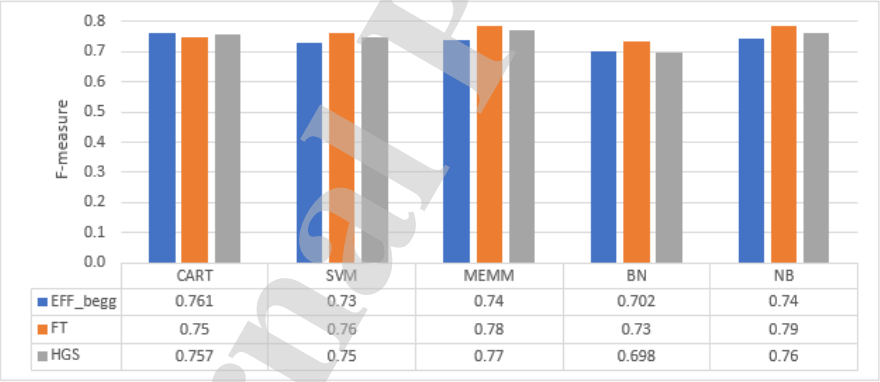
*Individual Feature Analysis of Dataset B.* Fig.12 shows individual feature analysis of selected feature set of Dataset B. FTA feature achieve highest average F-measure score of 79% and avg_TOV feature has the lowest score of 52% using CART classifier. SVM classifier achieves the highest average F-measure 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.

*Individual Feature Analysis of Dataset C.* Fig.13 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




| | CART | SVM | MEMM | BN | NB |
|---|---|---|---|---|---|
| 3PTA | 0.75 | 0.747 | 0.75 | 0.69 | 0.76 |
| 3PTper | 0.64 | 0.60 | 0.69 | 0.61 | 0.68 |
| avg_EFF_mid | 0.54 | 0.58 | 0.65 | 0.56 | 0.59 |
| avg_TOV | 0.52 | 0.52 | 0.57 | 0.57 | 0.53 |
| BLK | 0.73 | 0.72 | 0.77 | 0.71 | 0.78 |
| DREB | 0.76 | 0.751 | 0.79 | 0.73 | 0.768 |
| FG | 0.77 | 0.76 | 0.80 | 0.75 | 0.771 |
| FTA | 0.79 | 0.77 | 0.79 | 0.77 | 0.80 |
| FTper | 0.72 | 0.67 | 0.74 | 0.63 | 0.74 |

Figure 12: INDIVIDUAL FEATURE ANALYSIS OF DATASET B.




| | CART | SVM | MEMM | BN | NB |
|---|---|---|---|---|---|
| EFF_begg | 0.761 | 0.73 | 0.74 | 0.702 | 0.74 |
| FT | 0.75 | 0.76 | 0.78 | 0.73 | 0.79 |
| HGS | 0.757 | 0.75 | 0.77 | 0.698 | 0.76 |

Figure 13: INDIVIDUAL FEATURE ANALYSIS OF DATASET C.

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 and HGS features.

*5.5.2. Category Wise Analysis*

This section presents category wise analysis of various features. Feature types w.r.t to type and size are analyzed in this section for all of the three datasets. Fig.14 shows visual comparison of different categories of features.

*Analysis of Features Classified By Type.* Table.9 shows analysis of three types of features for dataset A. Derived features achieve highest average F-measure score on all classifiers.

Table 9: F-Measure Analyis of Features Classified By Type

| *Model* | *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 10: F-Measure Analyis of Features Classified By Size

| *Model* | *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.
+: Model with highest F-measure on a specific feature Type.

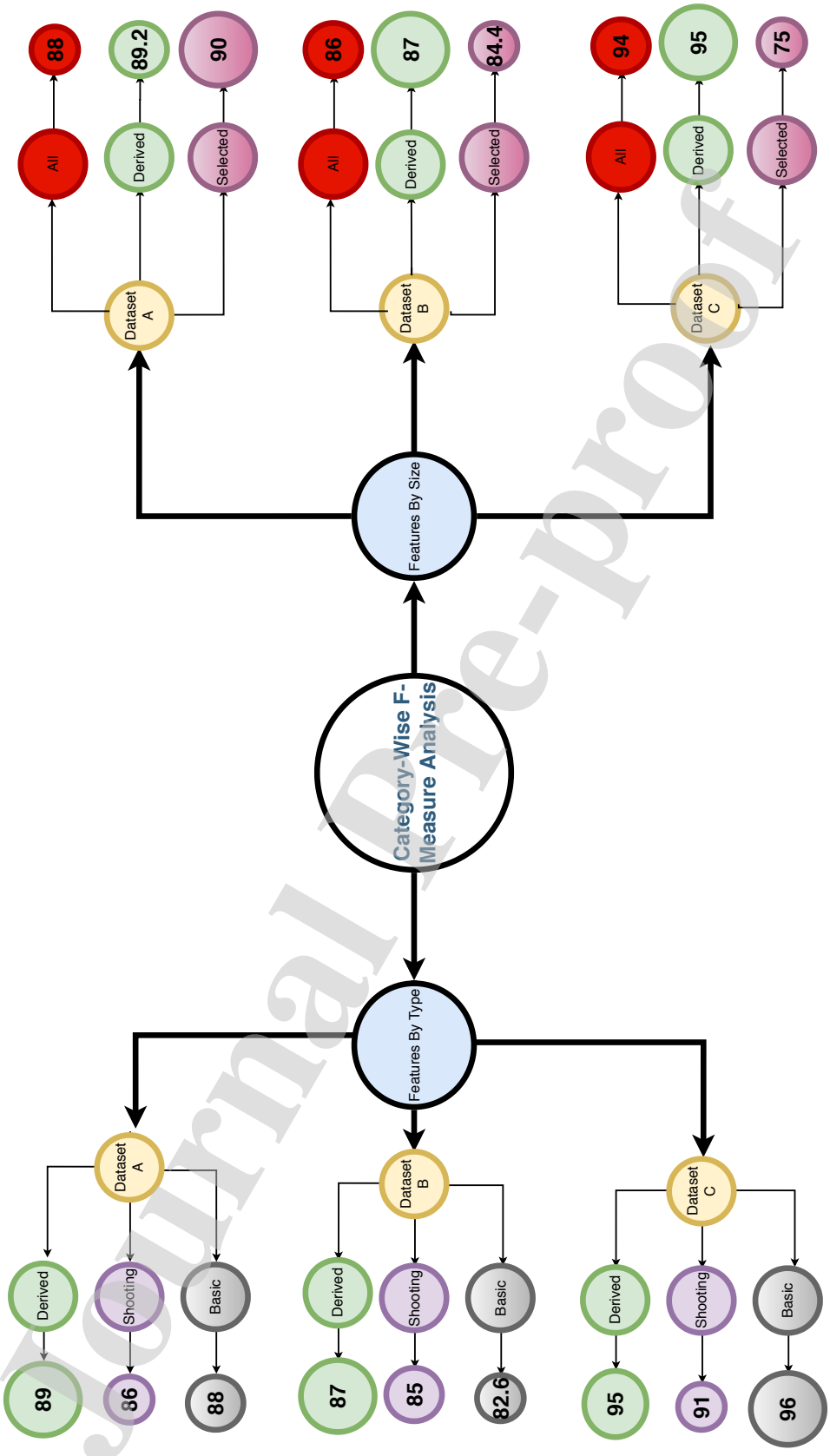



Figure 14: Comparison of F-measure score of different feature categories.

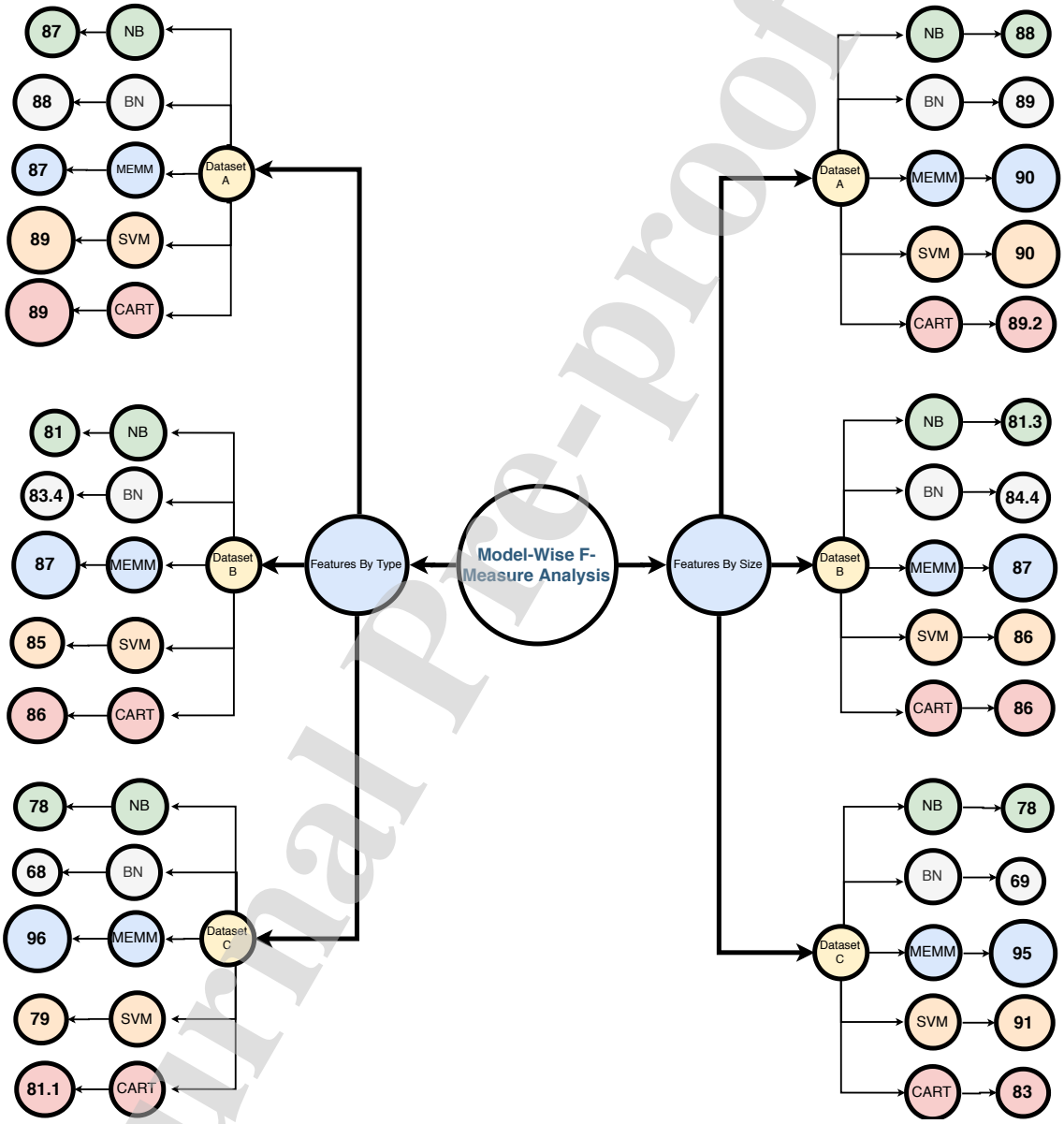



Figure 15: Comparison of F-measure score of different models on three datasets.

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.9 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.9 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.

*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.10 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. Us-

ing 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.10 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.10 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 outer perform than other classifiers on each of the feature set.

*5.5.3. Model 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.15 shows visual comparison of different machine learning models.

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

**Dataset A** Table.9 show F-measure score analysis of different classifiers while using Basic Feature set. Every classifier achieved a maximum of 100% accuracies 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.9 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.9 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.

**Dataset B** This section present analysis for different model by using dataset B with different type of feature sets. Table.9 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.9 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.9 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%.

**Dataset C** This section present analysis for different model by using dataset C with different type of feature sets. Table.9 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.9 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.9 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%.

*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.

**Dataset A** Table.10 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.10 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.10 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.

**Dataset B** Table.10 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.10 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.10 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%.

**Dataset C** Table.10 shows 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.10 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.10 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%.

### *5.6. SEASON-WISE RANKING COMPARISON OF TOP 20 LABELED RISING STARS*

From the labeled rising stars, Table.11 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.11. Season-wise ranking presented in Table.11 are taken from a sports website[6]. Fig.16 shows season wise change in the ranking of top-20 labeled players. Fig.16 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.

[6]http://www.espn.com/nba/seasonleaders

Table 11: 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 |

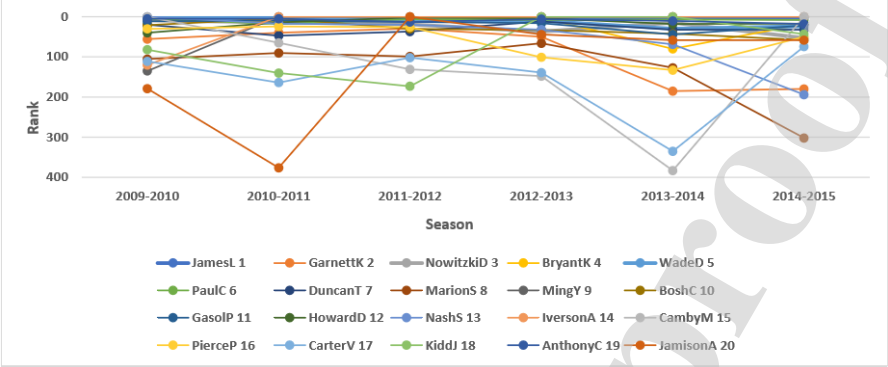



Figure 16: SEASON WISE RANKING OF TOP 20 LABELED RISING STARS.

### *5.7. 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 risings stars with MIP[7], 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.17 shows the rising stars have better average efficiency in each season as compared to Most Improved Players of that season.

[7]https://www.nba.com/history/awards/most-improved

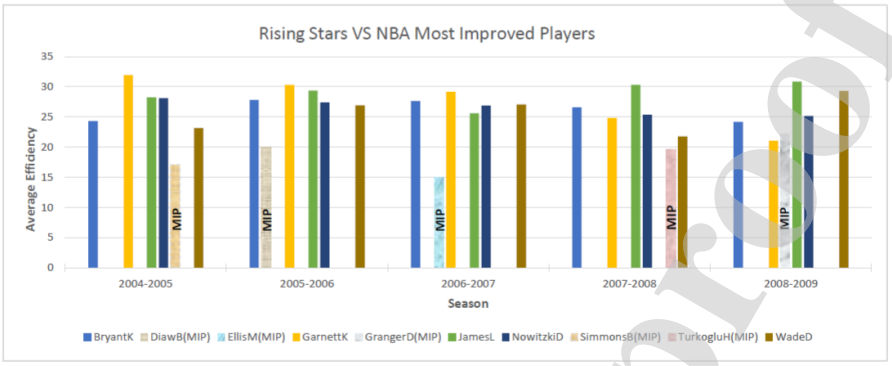



Figure 17: Comparison of Rising Stars Vs NBA Most Improved Players (Bar labeled with text "MIP" shows NBA most improved player of releavent season) .

## 6. Conclusion and Future Work

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 paper 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 paper 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) which belong to derived feature set, achieved highest F-measure of 84% by 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 feature set is 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 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.

In the future, we will extend our approach of rising star prediction to other sports like football and baseball. 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. One of the interesting work is to use social media data for prediction of rising star in sports.

## References


[1] X.-L. Li, C. S. Foo, K. L. Tew, S.-K. Ng, Searching for rising stars in bibliography networks, in: International conference on database systems for advanced applications, Springer, 2009, pp. 288–292.

[2] A. Daud, R. Abbasi, F. Muhammad, Finding rising stars in social networks, in: International conference on database systems for advanced applications, Springer, 2013, pp. 13–24.

[3] G. Tsatsaronis, I. Varlamis, S. Torge, M. Reimann, K. Nørvåg, M. Schroeder, M. 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, Springer, 2011, pp. 15–26.

[4] A. Daud, M. Ahmad, M. Malik, D. Che, Using machine learning techniques for rising star prediction in co-author network, Scientometrics 102 (2) (2015) 1687–1711.

[5] A. Daud, N. R. Aljohani, R. A. Abbasi, Z. Rafique, T. Amjad, H. Dawood, K. 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, 2017, pp. 33–41.

[6] T. Amjad, Y. Ding, J. Xu, C. Zhang, A. Daud, J. Tang, M. Song, Standing on the shoulders of giants, Journal of Informetrics 11 (1) (2017) 307–323.

[7] G. Panagopoulos, G. Tsatsaronis, I. Varlamis, Detecting rising stars in dynamic collaborative networks, Journal of Informetrics 11 (1) (2017) 198–222.

[8] F. Ding, Y. Liu, X. Chen, F. Chen, Rising star evaluation in heterogeneous social network, IEEE Access 6 (2018) 29436–29443.

[9] Y. Nie, Y. Zhu, Q. Lin, S. Zhang, P. Shi, Z. Niu, Academic rising star prediction via scholar's evaluation model and machine learning techniques, Scientometrics 120 (2) (2019) 461–476.

[10] A. Daud, M. Song, M. K. Hayat, T. Amjad, R. A. Abbasi, H. Dawood, A. Ghani, et al., Finding rising stars in bibliometric networks, Scientometrics (2020) 1–29.

[11] A. Daud, N. ul Islam, M. K. Hayat, R. A. Abbasi, H. Dawood, Prediction of rising business managers in telecommunication networks.

[12] A. Daud, F. Muhammad, H. Dawood, H. Dawood, Ranking cricket teams, Information Processing & Management 51 (2) (2015) 62–73.

[13] A. Daud, T. Amjad, T. Khaliq, H. Dawood, All that glitters is not gold: Falsely predicted rising stars, Researchpedia Journal of Computing (2020) Accepted.

[14] H. Ahmad, A. Daud, L. Wang, H. Hong, H. Dawood, Y. Yang, Prediction of rising stars in the game of cricket, IEEE Access 5 (2017) 4104–4124.

[15] P. Fearnhead, B. M. Taylor, On estimating the ability of nba players, Journal of quantitative analysis in sports 7 (3).

[16] S. K. Deshpande, S. T. Jensen, Estimating an nba player's impact on his team's chances of winning, Journal of Quantitative Analysis in Sports 12 (2) (2016) 51–72.

[17] F. ASGHAR, M. ASIF, M. A. NADEEM, M. A. NAWAZ, M. IDREES, A novel approach to ranking national basketball association players, Journal of Global Economics, Management and Business Research (2018) 176–183.

[18] J. Koster, B. Aven, The effects of individual status and group performance on network ties among teammates in the national basketball association, PloS one 13 (4) (2018) e0196013.

[19] S. Zhang, A. Lorenzo, C. Zhou, Y. Cui, B. Gonçalves, M. 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) (2019) 28–48.

[20] K. Trawinski, A fuzzy classification system for prediction of the results of the basketball games, in: International conference on fuzzy systems, IEEE, 2010, pp. 1–7.

[21] F. J. Ruiz, F. Perez-Cruz, A generative model for predicting outcomes in college basketball, Journal of Quantitative Analysis in Sports 11 (1) (2015) 39–52.

[22] P.-F. Pai, L.-H. ChangLiao, K.-P. Lin, Analyzing basketball games by a support vector machines with decision tree model, Neural Computing and Applications 28 (12) (2017) 4159–4167.

[23] M. A Gómez, S. J Ibáñez, I. Parejo, P. 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) (2017) 47–56.

[24] Y. Li, L. Wang, F. Li, A data-driven prediction approach for sports team performance and its application to national basketball association, Omega (2019) 102123.

[25] F. Thabtah, L. Zhang, N. Abdelhamid, Nba game result prediction using feature analysis and machine learning, Annals of Data Science 6 (1) (2019) 103–116.

[26] J. Shi, K. Song, A discrete-time and finite-state markov chain based in-play prediction model for nba basketball matches, Communications in Statistics-Simulation and Computation (2019) 1–9.

[27] C. Cortes, V. Vapnik, Support-vector networks machine learning vol. 20 (1995).

[28] I. Rish, et al., An empirical study of the naive bayes classifier, in: IJCAI 2001 workshop on empirical methods in artificial intelligence, Vol. 3, 2001, pp. 41–46.

[29] S. V. Wawre, S. N. Deshmukh, Sentiment classification using machine learning techniques, International Journal of Science and Research (IJSR) 5 (4) (2016) 819–821.

[30] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, S. O. Rezende, Knowledge-enhanced document embeddings for text classification, Knowledge-Based Systems 163 (2019) 955–971.

[31] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, I. Androutsopoulos, Large-scale multi-label text classification on eu legislation, arXiv preprint arXiv:1906.02192.

[32] S. G. Burdisso, M. Errecalde, M. Montes-y Gómez, A text classification framework for simple and effective early depression detection over social media streams, Expert Systems with Applications 133 (2019) 182–197.

[33] O. Abayomi-Alli, S. Misra, A. Abayomi-Alli, M. Odusami, A review of soft techniques for sms spam classification: Methods, approaches and applications, Engineering Applications of Artificial Intelligence 86 (2019) 197–212.

[34] T. Xia, X. Chen, A discrete hidden markov model for sms spam detection, Applied Sciences 10 (14) (2020) 5011.

[35] T. Xia, A constant time complexity spam detection algorithm for boosting throughput on rule-based filtering systems, IEEE Access 8 (2020) 82653–82661.

[36] Y. Chen, Y. Zhou, Machine learning based decision making for time varying systems: Parameter estimation and performance optimization, Knowledge-Based Systems 190 (2020) 105479.

[37] J. A. Castellanos-Garzón, E. Costa, J. M. Corchado, et al., An evolutionary framework for machine learning applied to medical data, Knowledge-Based Systems 185 (2019) 104982.

[38] W. Yujia, L. Jing, S. Chengfang, J. CHANG, et al., Words in pairs neural networks for text classification, Chinese Journal of Electronics 29 (3) (2020) 491–500.

[39] E. A. Bayrak, P. Kırcı, T. Ensari, Comparison of machine learning methods for breast cancer diagnosis, in: 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT), IEEE, 2019, pp. 1–3.

[40] Y. Wu, J. Li, J. Wu, J. Chang, Siamese capsule networks with global and local features for text classification, Neurocomputing.

[41] W.-Y. Loh, Classification and regression trees, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1 (1) (2011) 14–23.

[42] A. McCallum, D. Freitag, F. C. Pereira, Maximum entropy markov models for information extraction and segmentation., in: Icml, Vol. 17, 2000, pp. 591–598.

[43] K. Nigam, J. Lafferty, A. McCallum, Using maximum entropy for text classification, in: IJCAI-99 workshop on machine learning for information filtering, Vol. 1, Stockholom, Sweden, 1999, pp. 61–67.

[44] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Machine learning 29 (2-3) (1997) 131–163.

[45] M. A. Hall, Correlation-based feature selection for machine learning.

[46] J. E. Hirsch, An index to quantify an individual's scientific research output, Proceedings of the National academy of Sciences 102 (46) (2005) 16569–16572.

## Appendix A.

### *Appendix A.1. Example of Co-Player Influence*

Example given in Fig.A.18 shows the calculation of co-player's influence

| 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 |

Figure A.18: 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).

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

### *Appendix A.2. 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 Fig.A.19 Now in order to calculate the H-index of John, we simply

| Game | EFF (Efficiency) | Avg_EFF (Average Efficiency) |
|---|---|---|
| 1 | 5 | |
| 2 | 8 | |
| 3 | 2 | |
| 4 | 7 | (5+8+2+7+23+11+4+10) / 8 = **8.75** |
| 5 | 23 | |
| 6 | 11 | |
| 7 | 4 | |
| 8 | 10 | |

Figure A.19: Example showing the calculation of Average Efficiency.

write the efficiency in decreasing order as show in Fig.A.20. 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 Fig.A.21

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

H-index(1)=5

| Game | EFF (Efficiency) |
|---|---|
| 1 | 23 |
| 2 | 11 |
| 3 | 10 |
| 4 | 8 |
| **5** | 7 |
| 6 | 5 |
| 7 | 4 |
| 8 | 2 |

Cut-Off

Figure A.20: Example showing the calculation of H-index.

| Player | i | Co-Player | H-index |
|---|---|---|---|
| John | 1 | Mark | 5 |
| John | 2 | Brian | 3 |
| John | 3 | James | 8 |
| John | 4 | Jacob | 6 |

Figure A.21: Example showing H-indexs of Co-Players of John.

H-index(2)=3
H-index(3)=8
H-index(4)= 6
Now subsituting values in formula.16
Co_H-index = 5+3+8+6=22

*Appendix A.3. Data Acquisition*

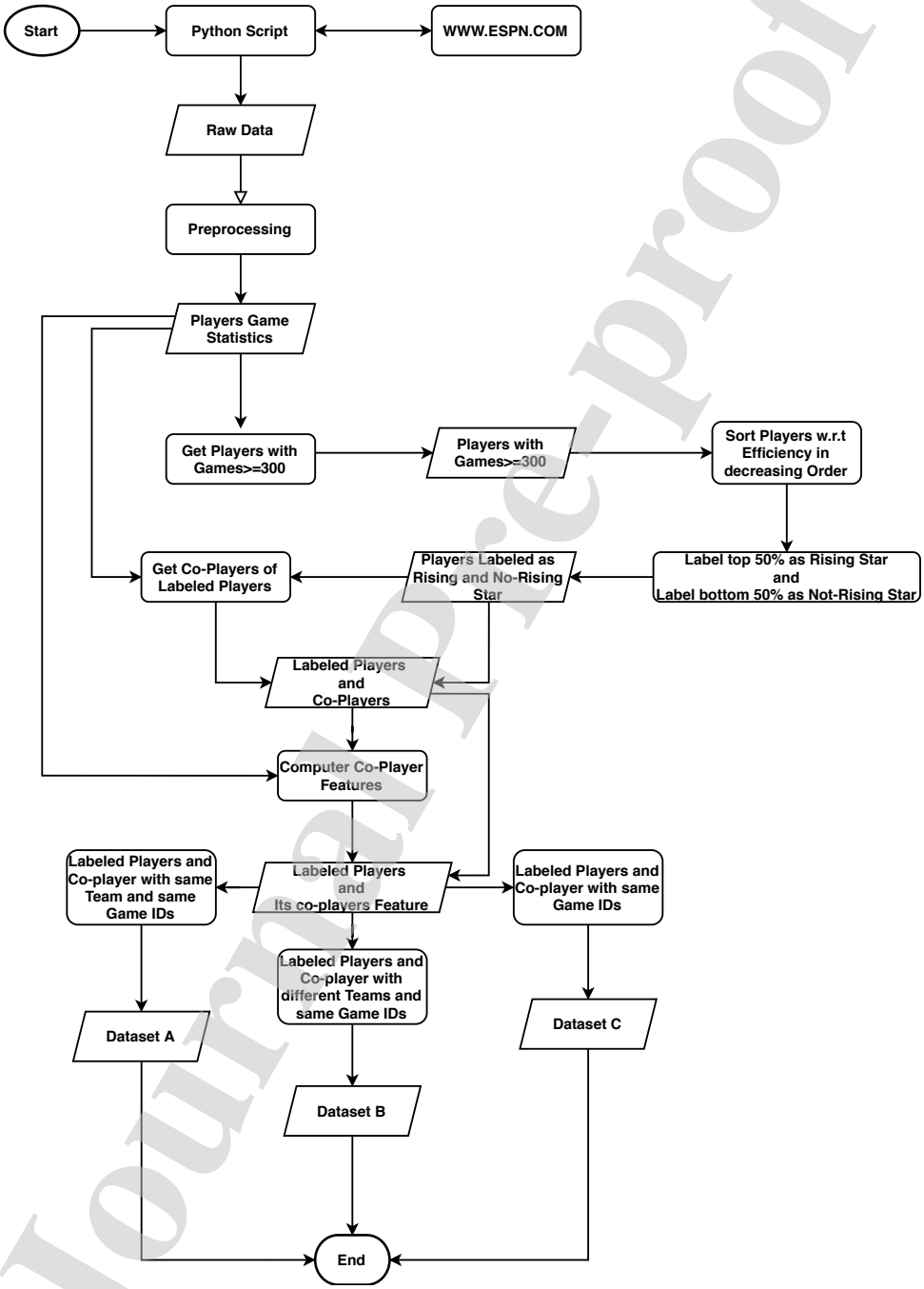



Figure A.22: Flow Chart showing dataset acquisition process.

**CREDIT AUTHOR STATEMENT**

ZAFAR MAHMOOD: Conceptualization, methodology, datasets, experiments, draft writing.

ALI DAUD: Supervision

Rabeeh Ayaz Abbasi: Co-Supervision

There is no conflict of interest.