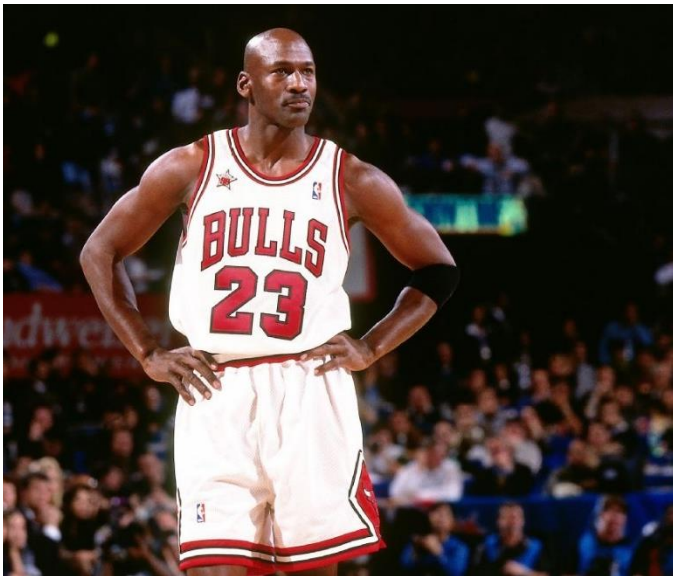
Analysis on NBA Player Position

Rain Wang
June.27.2020
ANLY 503 Assignment 3

Introduction

By mentioning basketball, for most people the first and the best league they can think of is NBA, which stands for the National Basketball Association. NBA is a men's professional basketball league in North America, composed of 30 teams (29 in the United States and 1 in Canada). It is one of the four major professional sports leagues in the United States and Canada, and is widely considered to be the premier men's professional basketball league in the world. Millions of people watch NBA games every season, in stadium or by TV.

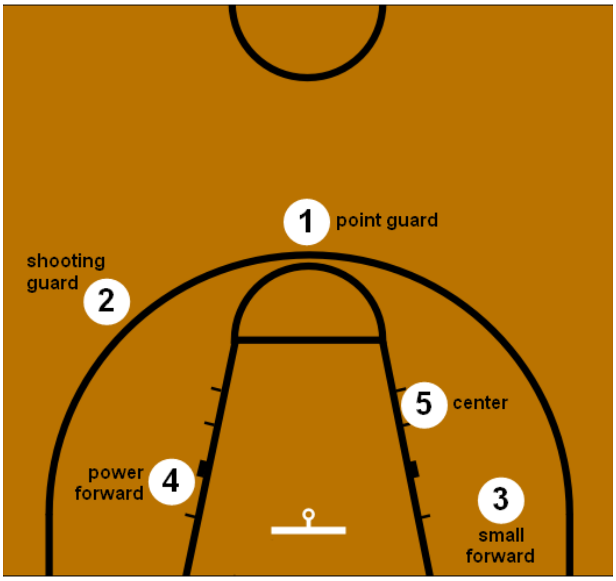
For people who are not familiar with the history of NBA, here are some brief information. The Basketball Association of America, or BBA, was founded in 1946 by owners of the major ice hockey arenas in the Northeastern and Midwestern United States and Canada. On August 3, 1949, the remaining NBL team merged into the BAA and the league name was changed to the present National Basketball Association. Beginning from 1980s, the league expanded quickly and got attention from basketball lovers all over the world, especially some talented player like



Michael Jordan

Magic Johnson and Michael Jordan joined the league. NBA programming during the 2017-18 season reached more than 1 billion unique viewers, and more than 35% of visitors to NBA.com come from fans outside North America.

In this project, the goal is to analysis the basketball positions and the data behind it. There are five positions in basketball. These positions are employed by all organized and unorganized basketball teams. They are defined as the point guard (PG), the shooting guard (SG), the small forward (SF), the power forward (PF), and the center (C). This report will explain how different positions affect the players' in-game stats like field goals they make or points they scores. And try to use machine learning methods to see if it is possible to predict the position of the player by its in-game stats.



Analysis

About the Data:

The NBA player stats dataset listed all the NBA player stats in season 2019. The original datasets have 560 rows and 58 columns, which stands for 560 players and 58 attributes for each player. Since there are too many columns, some important variable description is listed here:

Name: Player's name

Team: The abbreviation of the Team

Position: Player's position in the starting lineup (if started), otherwise the position he substituted for

Started: Number of games started

Games: The number of games played

Minutes: Total number of minutes played

FieldGoalsPercentage: Total field goal percentage

TwoPointersPercentage: Total two pointers percentage

ThreePointersPercentage: Total three pointers percentage

FreeThrowsPercentage: Total free throws percentage

OffensiveRebounds: Total offensive rebounds

DefensiveRebounds: Total defensive rebounds

Rebounds: Total rebounds

Assists: Total assists

Steals: Total steals

BlockedShots: Total blocked shots

Turnovers: Total turnovers

Points: Total points scored

PlusMinus: Total plus minus

DoubleDoubles: Total double-doubles scored

TripleDoubles: Total triple-doubles scored

Other variables are:

StatID, TeamID, PlayerID, SeasonType, Season, GlobalTeamID, Updated, FantasyPoints, Seconds, FieldGoalsMade, FieldGoalsAttempted, EffectiveFieldGoalsPercentage, TwoPointersMade, TwoPointersAttempted, ThreePointersMade, ThreePointersAttempted, FreeThrowsMade, FreeThrowsAttempted, OffensiveReboundsPercentage, DefensiveReboundsPercentage, TotalReboundsPercentage, PersonalFouls, TrueShootingAttempts, TrueShootingPercentage, PlayerEfficiencyRating, AssistsPercentage, StealsPercentage, BlocksPercentage, TurnOversPercentage, UsageRatePercentage, FantasyPointsFanDuel, FantasyPointsDraftKings, FantasyPointsYahoo, FantasyPointsFantasyDraft, IsClosed, LineupConfirmed, LineupStatus, PlusMinus

The label of the data in this report will be **Position**, as the position of each player. The report will focus on finding the relationship between player's position and his other gameplay stats.

The data this report used was found by API. Documentation about API can be found in the following link:

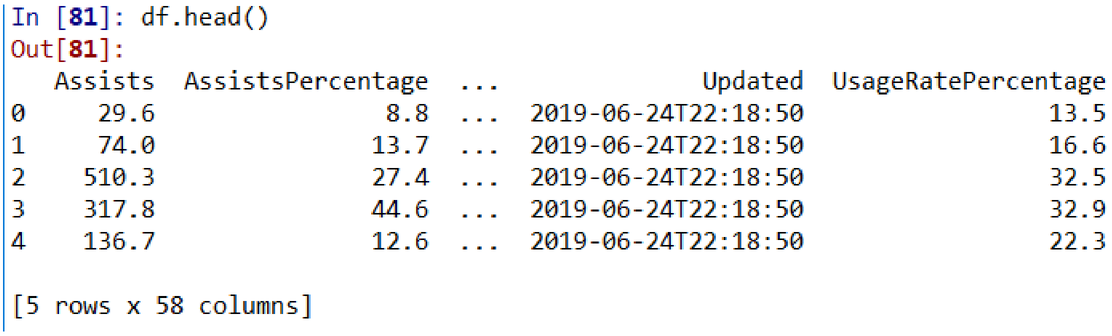
https://sportsdata.io/developers/api-documentation/nba#

Variable description can be found in following websites :

https://sportsdata.io/developers/data-dictionary/nba

A sample of raw data is here:

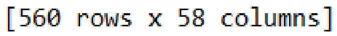
Head of Data (First Five Rows):



```
In [81]: df.head()
Out[81]:
   Assists  AssistsPercentage  ...              Updated  UsageRatePercentage
0     29.6                8.8  ...  2019-06-24T22:18:50                 13.5
1     74.0               13.7  ...  2019-06-24T22:18:50                 16.6
2    510.3               27.4  ...  2019-06-24T22:18:50                 32.5
3    317.8               44.6  ...  2019-06-24T22:18:50                 32.9
4    136.7               12.6  ...  2019-06-24T22:18:50                 22.3

[5 rows x 58 columns]
```

Size of the Data set:



```
[560 rows x 58 columns]
```

Data Cleaning:

The first cleaning of the data is to remove identical columns that does not help overall analysis. There are a lot of identical data for players, like **Name** or **PlayerID**. Team related variable are dropped as well, since this project aimed at the player's gameplay stats only. Also, some variable that are irrelevant to player's gameplay like all variable related to Fantasy points or ratings need to be dropped. So, the column drops in the first step is listed here:

Name, Team, StatID, TeamID, PlayerID, SeasonType, Season, GlobalTeamID, Updated, FantasyPoints, FantasyPointsFanDuel, FantasyPointsDraftKings, FantasyPointsYahoo, FantasyPointsFantasyDraft, IsClosed, LineupConfirmed, LineupStatus, PlayerEfficiencyRating, Started

Now, the datasets have 39 columns left, but it is still too many variables analysis on. In order to increase simplicity, all the columns about percentage are removed. Some variable has duplicate or similar meanings are redundant as well. The list of second drop is following:

FieldGoalsPercentage, TwoPointersPercentage, ThreePointersPercentage, FreeThrowsPercentage, EffectiveFieldGoalsPercentage, OffensiveReboundsPercentage, DefensiveReboundsPercentage, TotalReboundsPercentage, TrueShootingPercentage, AssistsPercentage, StealsPercentage, BlocksPercentage, TurnOversPercentage, UsageRatePercentage, OffensiveRebounds, DefensiveRebounds, PlusMinus.

After the first step of cleaning, there are 22 columns left in the dataset. Further detailed cleaning can be found in the further sections

Games

Game variable represents the number of game players played in the season. The distribution of games played by players can be seen in the histogram below. There are two peaks in the distribution, showing that many players are the core players, plays almost every game, and a lot of players are substitution players that has almost no chance to play in the entire season. Less performance players need to be dropped as their game stats might be biased. In this report, players with less than 10 games played is dropped. Total players remain after cleaning Games variable is 469.

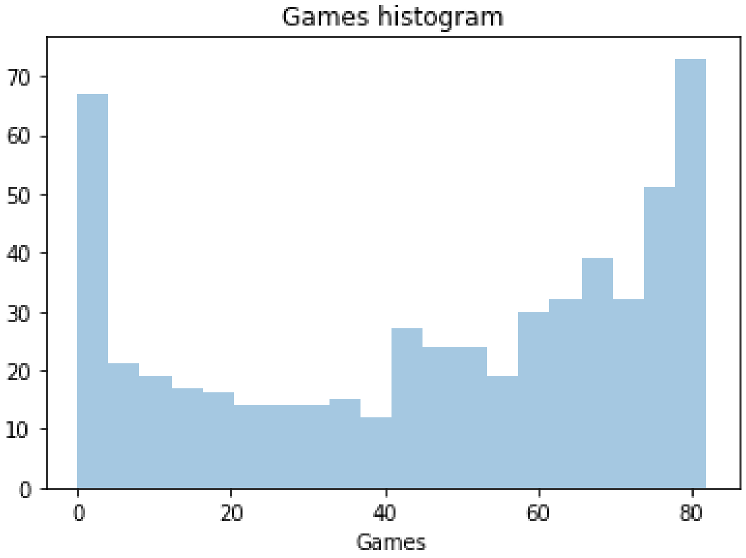


Figure 1.1 Games distribution before cleaning. Two peaks appear in the distribution, at both left end and right end. Player plays less amount of games are going to make the analysis bias.

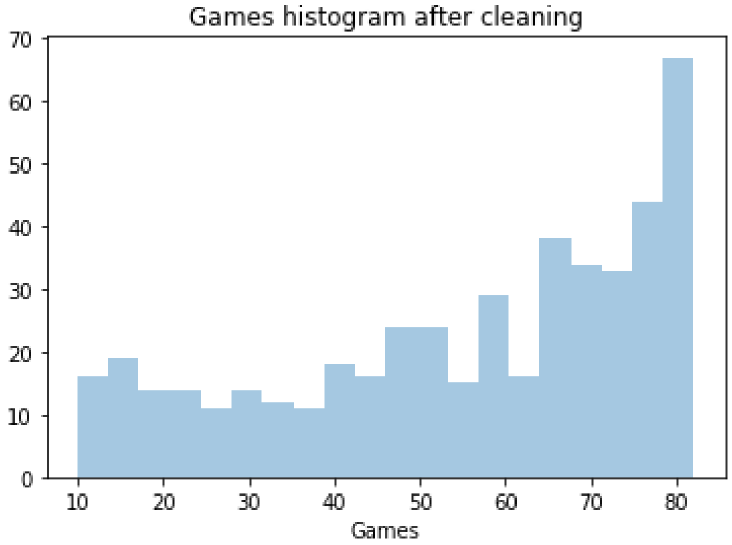


Figure 1.2 Games histogram after cleaning. The low peak is removed from the graph as those players are intended to create bias through analysis. After cleaning, there are only one peak at right end, represents the core players for each team that plays almost every game.

Minutes and Seconds

Minutes and Seconds variable both correspond to total time players played on field. By looking at the graphs, it is obvious that second is ranged from 0 to 60, which shown that Seconds variable here does not means total seconds. Instead, a player who played 60 minutes and 10 seconds this season is going to have Minutes variable as 60 and Seconds variable as 10. To make the further study clear, **Seconds** and **Minutes** are combined to a new variable named as **TotalSeconds**.

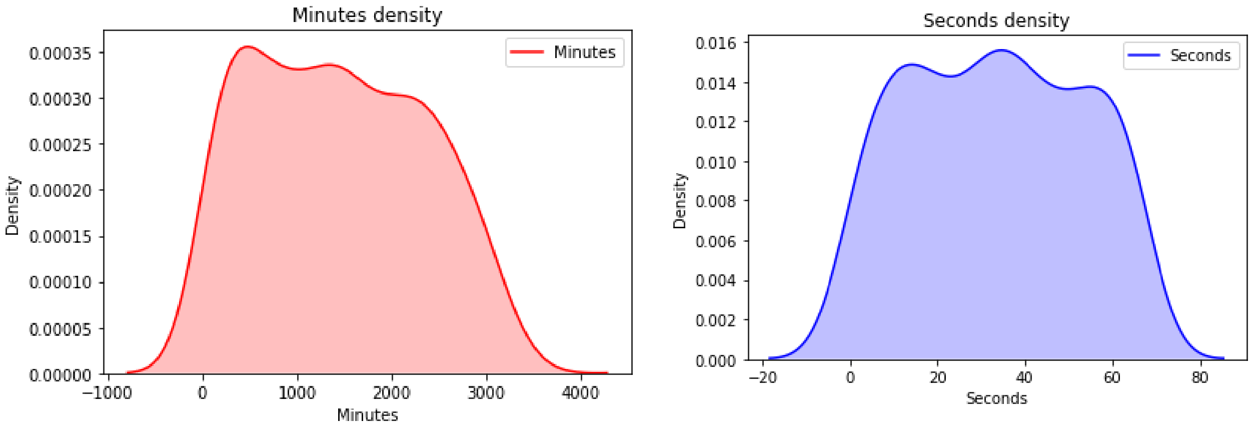


Figure 1.3 Minutes and Seconds density graphs are shown. For the Seconds variable, the range is from 0 to 60, which means that the Seconds variable here does not means total seconds. Needs to combine both variable to get total seconds.

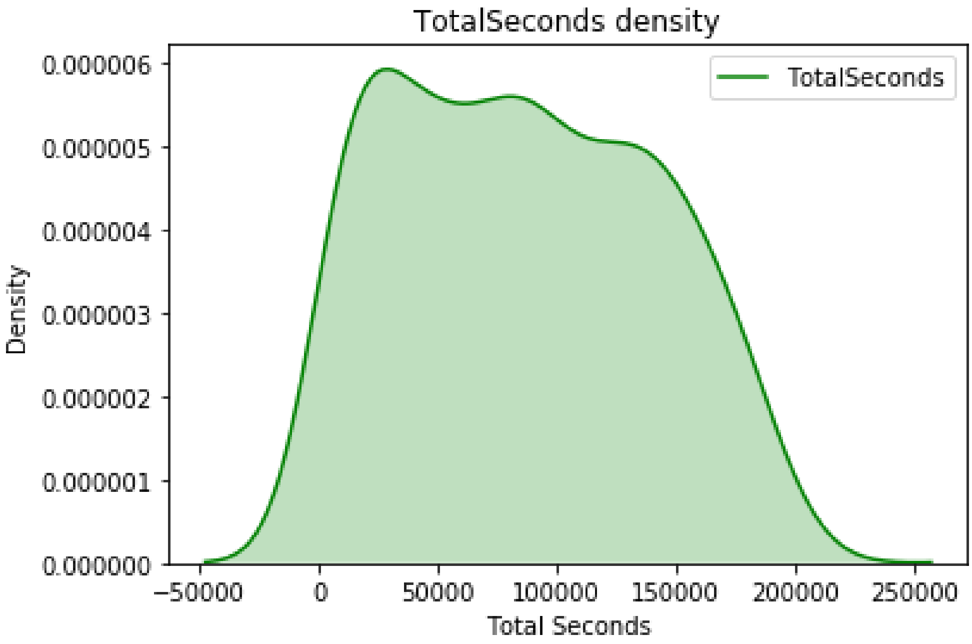


Figure 1.4 TotalSeconds variable is combined by Minutes and Seconds. The overall distribution is similar as Minutes variable. Most players tend to play around 45,000 seconds per season. Some core player plays around 200,000 in one season.

Other Gameplay Stats

Let's then look at other gameplay stats for players. The example here is Rebound, a gameplay stat that C and PF, or center and power forward position players are mainly responsible for, versus Assists, a gameplay stat that SG and PG are usually responsible for. The scatterplot of Figure 1.5 should the distribution of Rebounds and Assists stats by position. Skewness of the time players played on field makes a lot of low Assists and Rebounds data points and unclear boundary on scatterplot. In order to have a more accurate description for player, **all player's gameplay stats are cleaned to a per second base stats.**

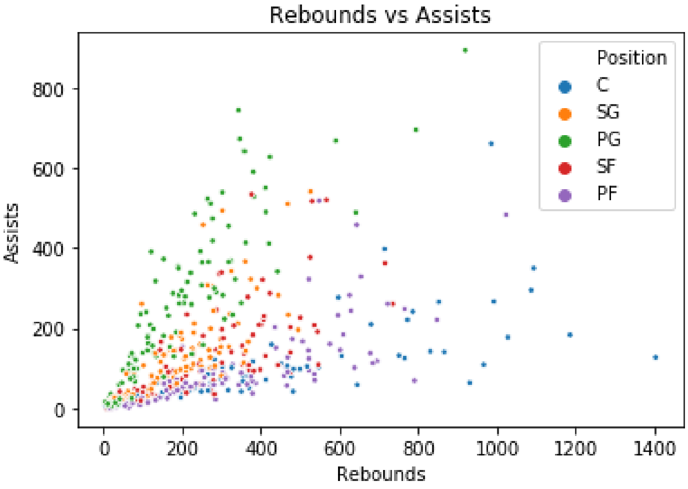


Figure 1.5 Rebounds and Assists scatterplot for player total stats of whole season. It is possible for a core PG player to have much more rebounds than a substitution C player just because he player much more game. There are too many datapoints has small amount assists and rebounds.

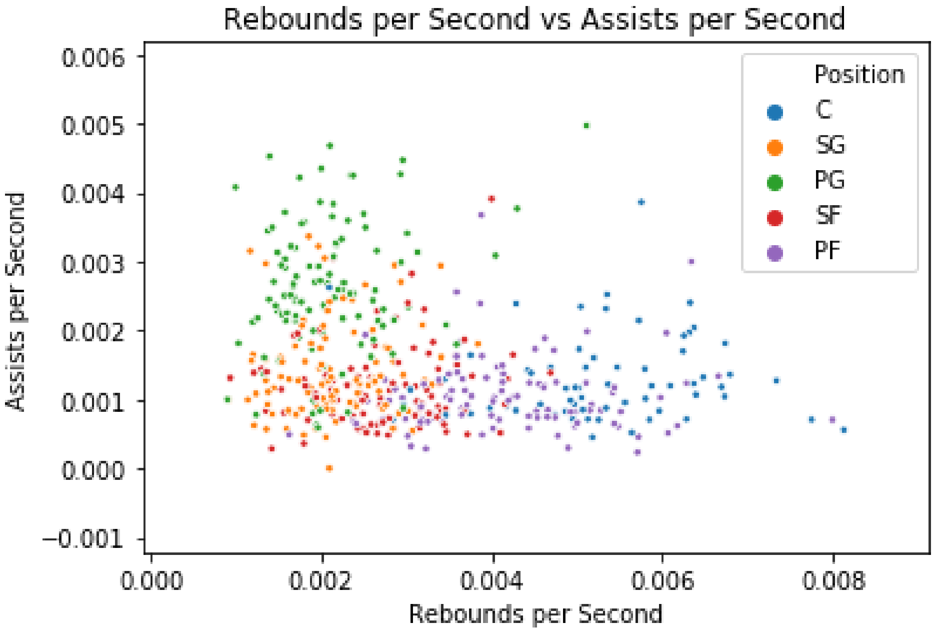


Figure 1.6 Rebounds Per Second and Assists Per Second scatterplot for players. The relation between each position are clearer by making both variable in a per second unit. PG used to have lower number of rebounds but more assists. SG and SF has similar attribute considering assists and rebounds. PF and C are mainly responsible for team's rebound work.

Exploratory Data Analysis:

For exploratory data analysis of the report, it contains visualization about distributions and relationships between each variable and our label, Position variable, as well as visualizations that explain relations between other variable. The description of all the visualizations can be find below the graphs.

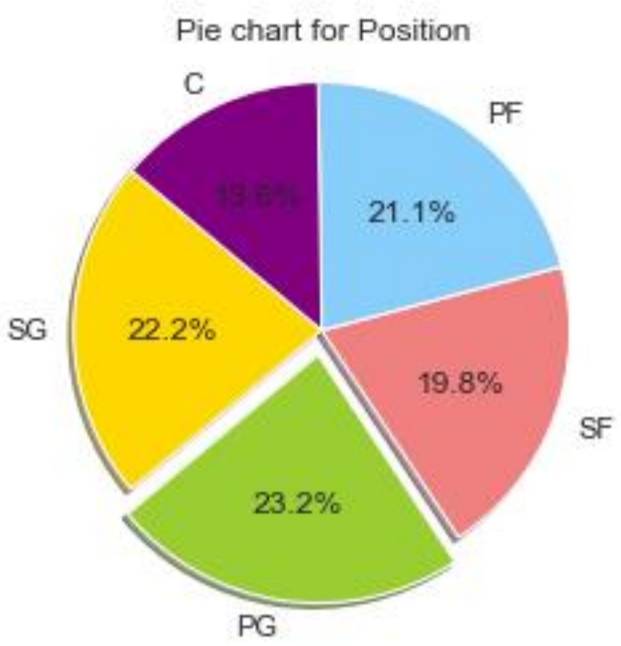


Figure 2.1 Position distribution of NBA players after all data cleaning. SG PG PF and SF have similar number of players in the dataset. But there are less Center position players in the league. Only 13.6% percent of players are C players

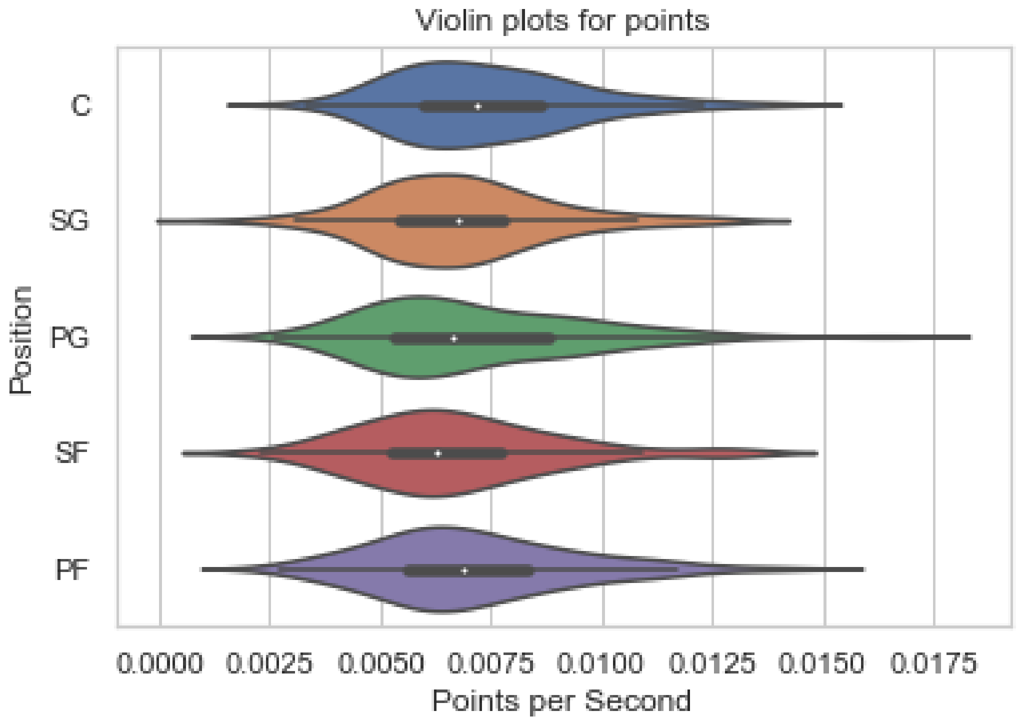


Figure 2.2 Violin plots for points the player gain per second versus their position. Surprisingly, the ability of players for each position to score for points is very close to each other. The violin graphs show similar distribution as well as close mean. PG have higher upper limit of gaining points, but overall, the difference between each position is small on Points

Correlogram for goal related variables

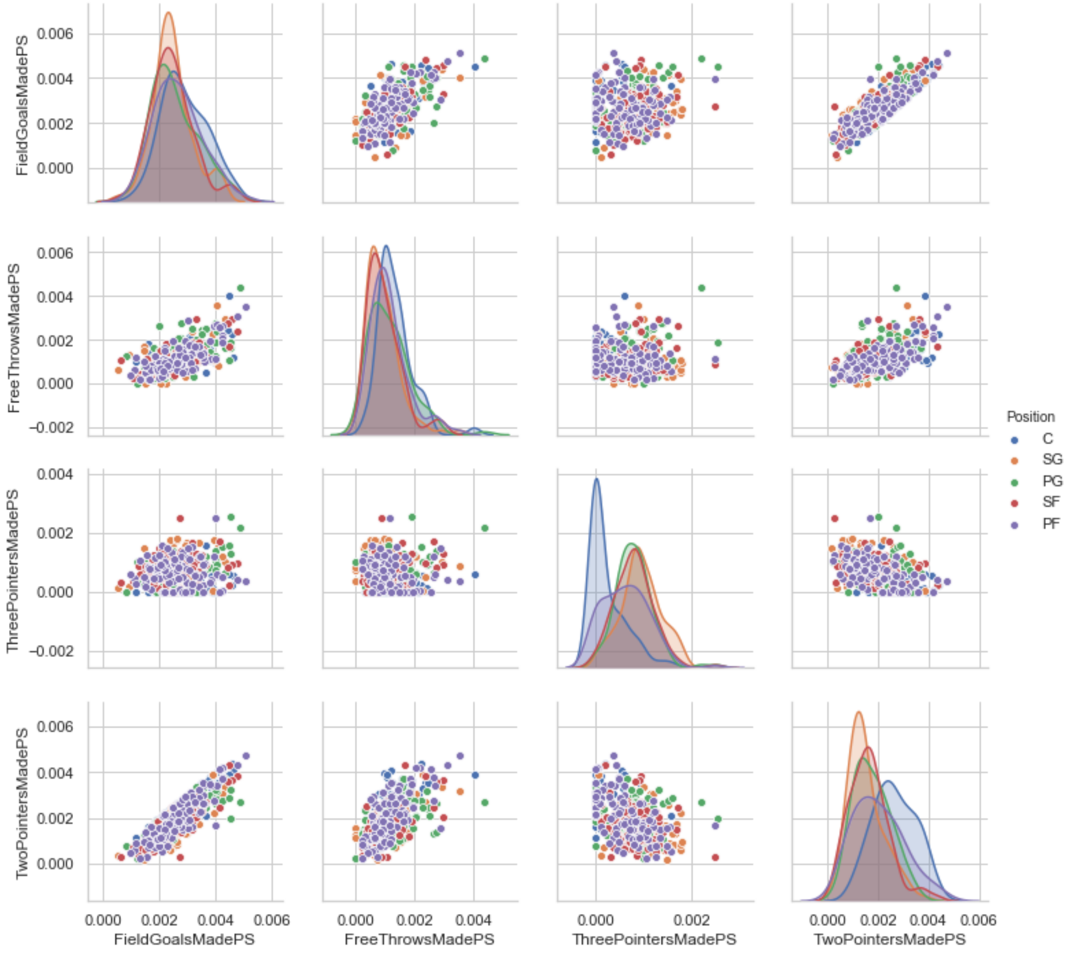


Figure 2.3 Correlogram for goals and shoots related variables. It is obvious to see a linear correlation between Field Goals and Two Pointers, as well as Field Goals and Free Throws. Also, by looking at the distribution if the Three Pointers, Center position players have significantly low three pointers made. For other goals abilities, all positions are similar

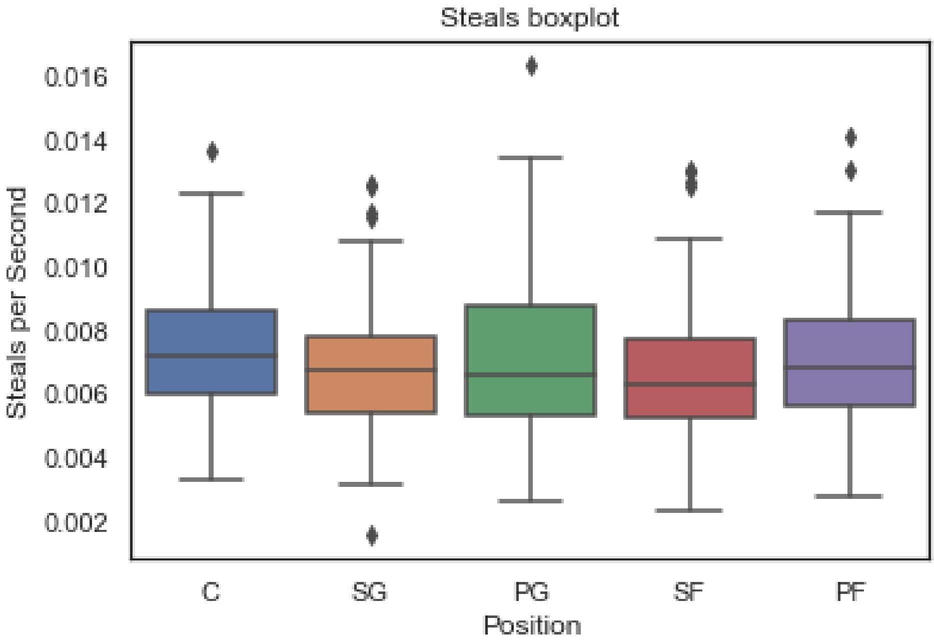


Figure 2.4 Boxplot for steals. Players for all position has similar steals distributions, as steals is the gameplay action that rarely take place during the game. In comparison, PG are doing a better job in steals, follows by SF and PF. One PG player has a really high Steals per second data than other players.

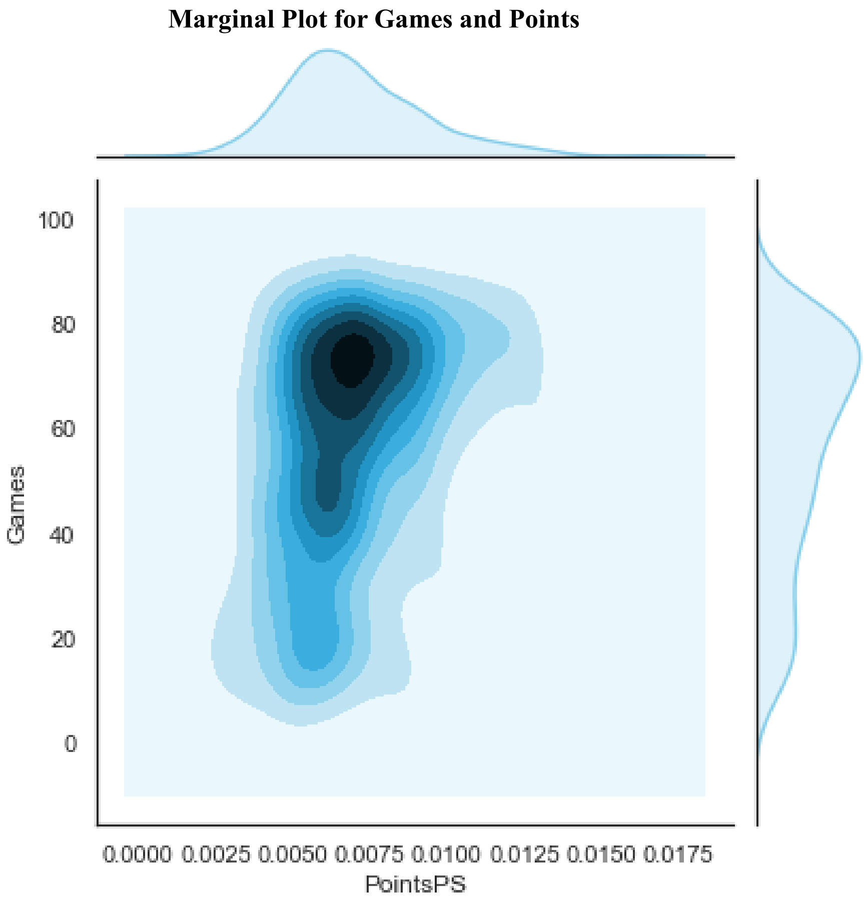


Figure 2.5 Marginal Plot for Games and Points per Seconds. The margin of the graphs shows the distribution of the variable corresponding to the axis. Middle of the graphs shows the densities. As the right side of the darkest color area shows, players who played more games tends to have stronger ability to gain points. Team want the players on field to have the ability to gain points for the team.

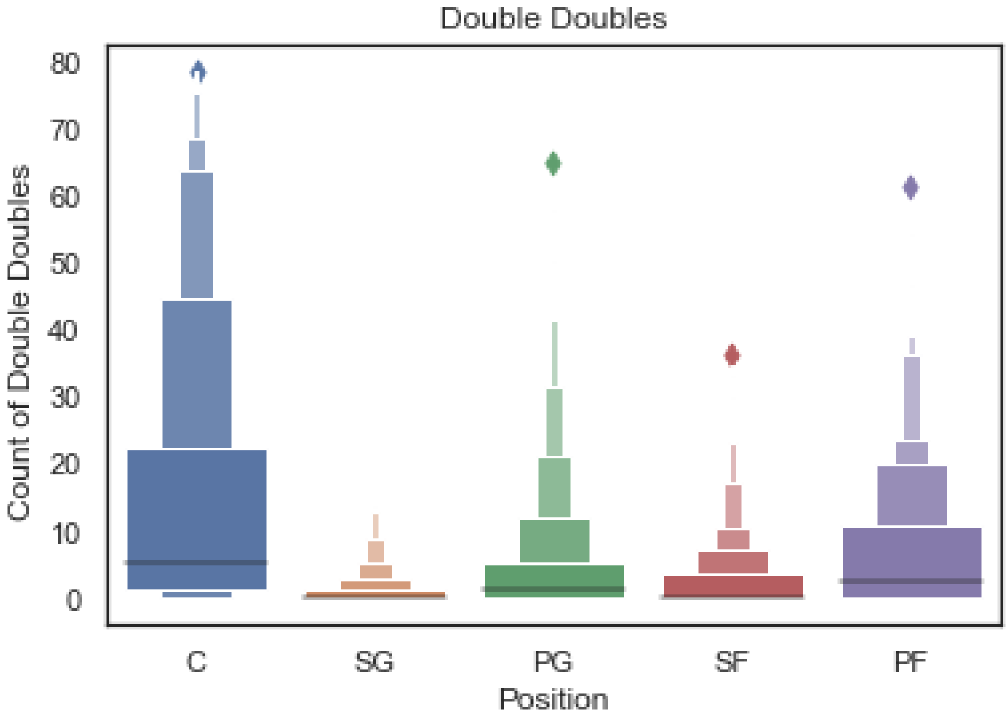


Figure 2.6 Double Doubles boxen plot. Double Doubles stands for having two in game stats that is ten or above in one game. Looking at the count of Double Doubles of each player for the entire season, the median for all position players are low, as Double Doubles is hard to achieve. But C players, or center position players, tends to have more Double Doubles then other positions, and PF comes the second. One major reason is that C and PF position players are good at taking rebounds, makes them easy to get a second double in rebounds stats other than points.

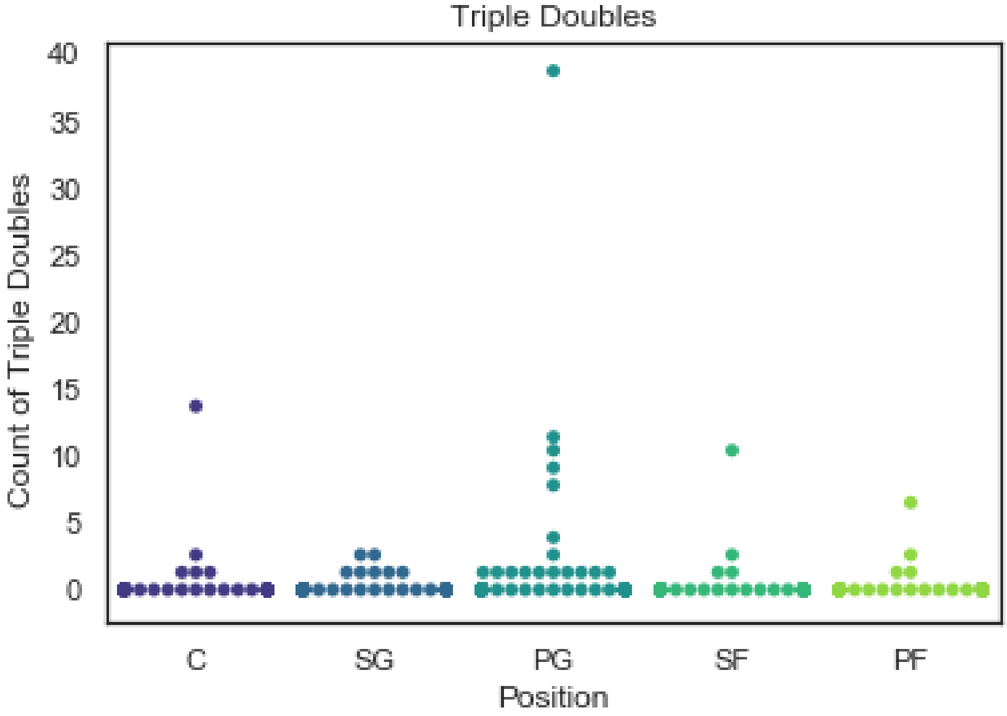


Figure 2.7 Triple Doubles Swarmplot. Different than Double Doubles shown in Figure 2.6, Triple Doubles are even harder to achieve, as most of players results in having 0 Triple Doubles in the entire season. For all those Triple Doubles owners, most of them are PG, which are very different than the distribution of Double Doubles. The explanation for this result is that C and PF are easy to get Double Doubles, but they usually have low assists, which makes them hard to reach Triple Doubles.

Machine Learning:

In this report, two machine learning methods are used in. Here are some basic descriptions of two models, some explanation about how they work, and what the expectation are for each model.

For both models, the dataset was split into test set and train set. Train set is used to build the model while test set is used to test the result of models. In this report, the train-test ratio is 7:3. By checking Figure 3.1, the distributions of Position variable of train and test set are similar, which means is good for machine learning.

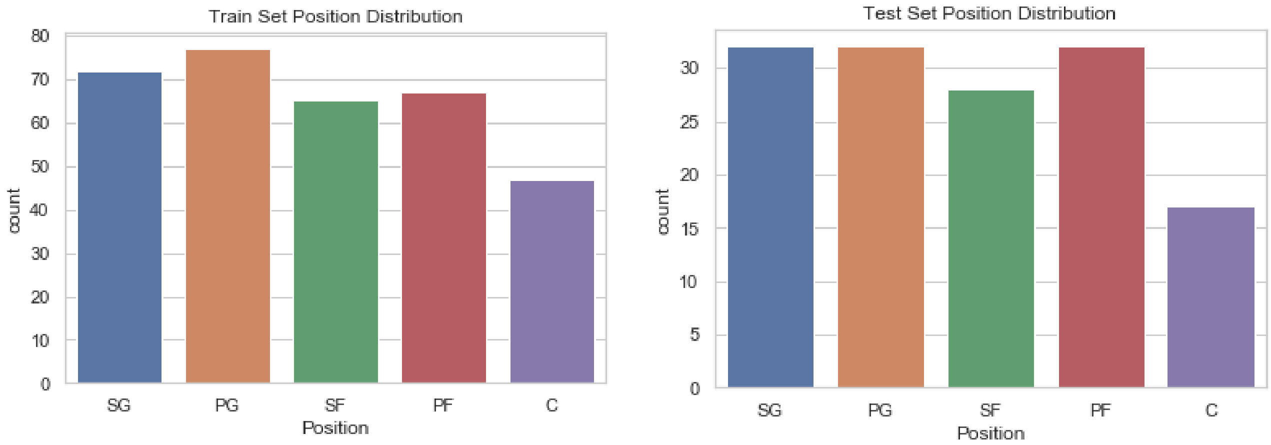


Figure 3.1 The Position variable distribution for both Train Set and Test Set. The distribution between two set are similar. PG, SG, SF, and PF has similar amount in both sets. C has less amount than other four positions.

Decision Tree

In this report, the first machine learning method used is decision tree. A decision tree goes from observations about an item (represented in the decision node and the decision node) to conclusions about the item's target value (represented in the leaf node). Since the goal is to classify the platform, this report is going to build a classification tree, as in this case leaves represent class labels and decision node represent conjunctions of features that lead to those class labels. The example of a decision tree is shown in Figure 3.2.

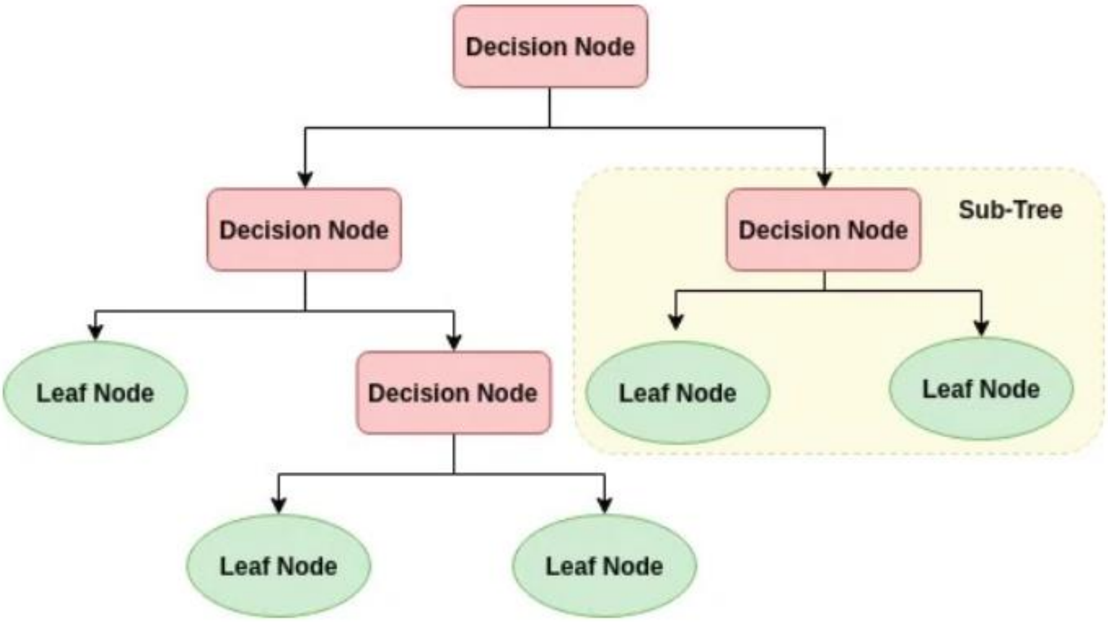
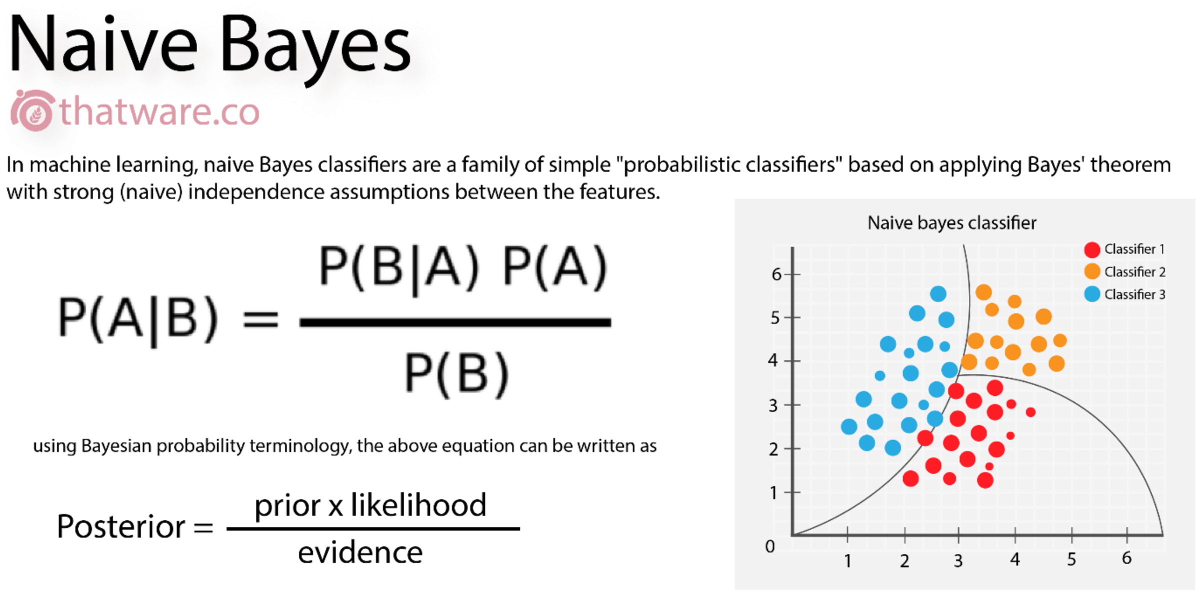


Figure 3.2 A general example of a decision tree. From the top decision node, according to the decision made, go through the branches. At the end it reaches to a leaf node, which is the label that the data belongs to.

The expectation for decision tree model is to build up and draw a decision tree from the train set, using the tree model to predict the label, or the position of the player, and build up to a confusion matrix to make comparison with other methods. The tree built and the confusion matrix can be found in the result section of the paper, under decision tree subsection.

Naïve Bayes

The second machine learning method used is Naïve Bayes. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. Naïve Bayes model is based on the Bayes' theorem of conditional probability: Posterior is equal to prior times likelihood divided by the evidence. A more detailed explanation about Naïve Bayes model and Bayes theorem are explained in the equation in Figure 3.3.



Naive Bayes

thatware.co

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$

Figure 3.3 An explanation of the function for Naïve Bayes. It is based on the Bayes' theorem of conditional probability: Posterior is equal to prior times likelihood divided by the evidence. Base on this formula and training set to build the Naïve Bayes model.

The expectation for Naïve Bayes is to build the model from the train set and predict the result from test set, using the Naïve Bayes model to predict the label, or the position of the players, and build up to a confusion matrix to make comparison with other methods. The confusion matrix can be found in the result section of the paper, under Naïve Bayes subsection.

Result

Decision Tree:

By putting all the variable into the decision tree model, the tree build up is shown in the following Figure 4.1. Decision tree provides most direct and understandable result. The logic behind this decision tree make sense in reality. For example, from the decision, if the players have more rebounds, more than 0.003 rebounds per seconds, they are predicted to be C or PF. ReboundsPS is the most important factor in this decision tree, and other important variables are AssistsPS, TreePointersAttemptedPS and BlockedShotsPS.

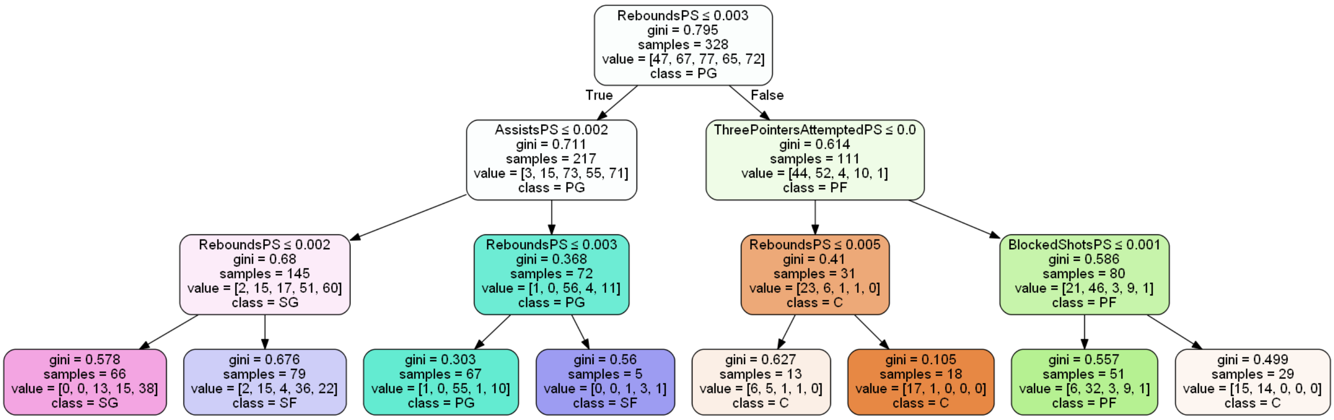


Figure 4.1 Decision Tree model result. For each node, left branch means yes and right branch means no. For example, for a player to fail into PF label, the player will have greater than 0.003 Rebounds per second, greater than 0.0 Three Pointers Attempted per second and less than 0.001 Blocked Shots per second.

By running the model on test set, a confusion matrix of the predict result of the decision tree can be built. The confusion matrix of the decision tree is shown in Figure 4.2. The calculated accuracy of the prediction for decision tree model is: **0.610**

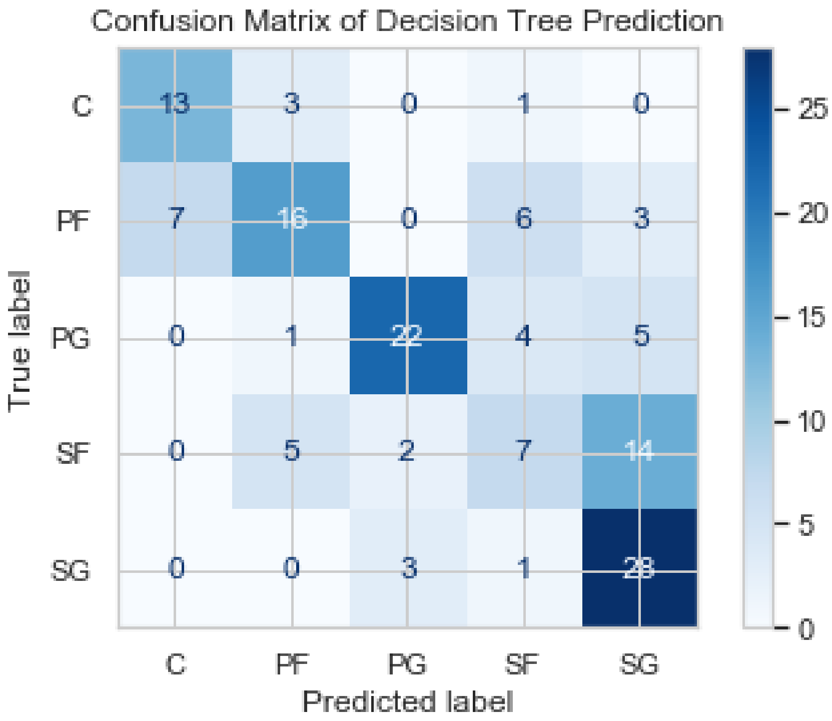


Figure 4.2 Decision Tree model confusion matrix. The decision tree model makes a good prediction on SG and PG players, as 28 and 22 correct prediction in respective. This decision tree has poor prediction on whether the player is SF or not, as considering 14 SF players as SG player.

Naïve Bayes:

Naïve Bayes model also successfully helps to predict the result of test set by train set. Since the data is multi-dimension, there is not an accurate plot to show the result of Naïve Bayes.

By running the model on test set, a confusion matrix of the predict result of the Naïve Bayes can be built. The confusion matrix of the Naïve Bayes is shown in Figure 4.3. The calculated accuracy of the prediction for decision tree model is: **0.418**

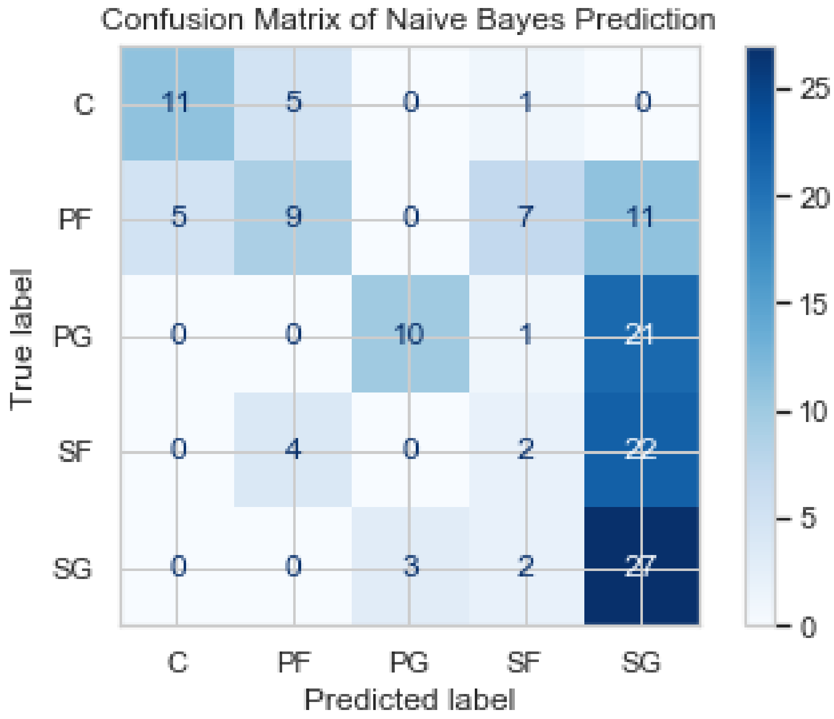


Figure 4.3 Naïve Bayes confusion matrix. The Naïve Bayes model also makes a good prediction on SG player, with 27 correct prediction. But for other positions, Naïve Bayes model does not make good prediction. Only 2 two correct SF prediction were made by Naïve Bayes model.

Comparison Between Models:

The results of machine learning model and the discussion are shown in the table below:

Decision Tree	Naïve Bayes
Accuracy: 0.610	Accuracy: 0.418
The More Effective Model	
Has a clear model expression. Easy to understand the model. Has a high accuracy rate on SG and PG prediction. Can make easily manmade prediction base on decision tree plot.	Has no clear model expression. Hard to understand the model. Model has low accuracy rate on NBA player dataset, possible reason is that the data set violates the assumption of independent features, as the predictors are dependent to each other.

Conclusion

This report analysis on the NBA players gameplay data for 2019 season. From the data, players in NBA can be consider as core players that plays almost every game in the season, and substitution players as some of them might not have chance to play on field for the entire season. Because of the huge diversity in the games or total time periods played between players, using per time unit data is more persuasive than the total stats. Different position players did have huge difference in their game play stats. C and PF players are better at rebounds, while PG are better at assists. But surprisingly, all position players have the similar ability to score points for their team, even though PG, SG and SF are better at three pointers and C and PF have more two pointers.

Lebron James with Triple Doubles

For the machine learning part, decision tree is the better performance model in this report. It is very suitable for this NBA data set that has some very clear relationship between our label, Position, and some variables, like Rebounds and Assists. Decision tree also provides a very clear tree plot to help people understand the model. Because some gameplay data are related to each other, Naïve Bayes does not have good accuracy in prediction the position of the player.

A small suggestion for future NBA player is that position is very important in the game, as it is highly related to the in-game action and gameplay stats. Also, NBA are lack of Center position players comparing to other position. So, if a player is confidence in his height and rebounds ability, he should try to play Center position, as more team might need a good center player, with also a higher chance to get Double Doubles than other positions.

NBA Center Position Players